Università degli studi Roma Tre

Department of Industrial, Electronic, and Mechanical Engineering

Doctor of Philosophy Thesis in Applied Electronics

Scene Understanding with Sound using Artificial Intelligence Techniques

Supervisors
Prof. Marco Carli

Prof. Alessandro Neri

Coordinator
Prof. Maurizio Schmid

Ph.D. Candidate
Michael Neri

A.Y. 2024/2025

A dissertation thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Applied Electronics at the Department of Industrial, Electronic, and Mechanical Engineering of Roma Tre University.

Abstract

This PhD thesis explores the field of scene understanding using sound through artificial intelligence techniques. It addresses the challenge of extracting relevant information from sound in environments where other sensory inputs, such as vision, are limited or occluded. The work contributes novel methods and models for Acoustic Scene Classification (ASC), Sound Event Detection (SED), Unsupervised Anomalous Sound Detection (UASD), and speaker Distance Estimation, with a focus on reducing the complexity of these systems while maintaining high performance.

The core of this research lies in the design of low-complexity deep learning models, such as lightweight convolutional networks and methods leveraging Chebyshev moments, which are applied to various sound recognition tasks. These models are tested in noisy environments and shown to be robust, offering state-of-the-art results while being computationally efficient.

In addition to the theoretical contributions, the thesis explores practical applications of sound-based scene understanding in domains such as smart devices, security systems, and autonomous vehicles, enhancing human-computer interaction and safety. Future research potential includes the integration of multi-modal sensory data and the development of more interpretable AI systems.

Abstract

Questa tesi di dottorato esplora il campo della comprensione delle scene attraverso il suono, utilizzando tecniche di intelligenza artificiale. Affronta la sfida di estrarre informazioni rilevanti dal suono in ambienti dove altri input sensoriali, come la vista, sono limitati o ostruiti. Il lavoro fornisce nuovi metodi e modelli per *Acoustic Scene Classification* (ASC), il *Sound Event Detection* (SED), l' *Unsupervised Anomalous Sound Detection* (UASD) e la *speaker distance estimation*, con un'attenzione particolare alla riduzione della complessità di questi sistemi mantenendo alte prestazioni.

Il nucleo di questa ricerca risiede nella progettazione di modelli di deep learning a bassa complessità, come reti convoluzionali leggere e metodi che sfruttano i momenti di Chebyshev, applicati a vari compiti di riconoscimento sonoro. Questi modelli sono testati in ambienti rumorosi e si dimostrano robusti, offrendo risultati all'avanguardia e garantendo al contempo un'efficienza computazionale.

Oltre ai contributi teorici, la tesi esplora applicazioni pratiche della comprensione delle scene basata sul suono in ambiti come dispositivi intelligenti, sistemi di sicurezza e veicoli autonomi, migliorando l'interazione uomo-computer e la sicurezza. Le potenzialità future di ricerca includono l'integrazione di dati sensoriali multimodali e lo sviluppo di sistemi di intelligenza artificiale più interpretabili.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

1.1 Motivation

One of the most engaging aspects of our perception of the environment is sound. It encompasses vast and diverse pieces of information about sources, events, and activities in the environment. It can even evoke emotions, memories, and associations that create our experience and behavior. Humans have evolved to process sound efficiently and effectively, using it for communication, navigation, and interaction with the environment. Typically, sound is ignored by most machines, which will turn instead to their visual or textual inputs to do their job. However, sound can offer not only complementary but also superior information to other modalities, especially in complex and dynamic settings, or when other modalities are occluded.

In this context, **scene understanding** can be denoted as the act of inferring pieces of information about a scene from its acoustic properties alone. This encompasses identifying the source, its location, and associated actions; understanding the geometric configuration and spatial arrangement of the scene; as well as discerning the contextual atmosphere and emotional tone. Mastering these elements constitutes a fundamental capability that artificial agents must develop to operate effectively within authentic and realistic environments. In such settings, sounds not only form an essential component of the scene but also provide significant information about the environment. This can be enhanced by the use of AI techniques—machine learning, deep learning, and computer vision—to the effect that large-scale data and powerful models can enable learning from sound to extract meaningful features and representations. We motivate our research by providing some examples of scenarios where sound can play an important role in scene understanding, and some applications where scene understanding with sound can have a positive impact. We also discuss the main challenges and opportunities of scene understanding with sound using artificial intelligence techniques and outline the main objectives and contributions of this thesis.

In most cases, sound can be an important component to scene understanding whenever visual or textual information is limited, incomplete, or unreliable. This makes it possible, for example, to locate and identify sources that are not in the line of sight because they are invisible or occluded by objects in the scene, such as a person talking behind a wall, a car horn in a traffic jam, or even a bird chirping in the woods. In addition, it can be used to identify and recognize slight or undefined sound events such as a door opening, glass breaking, or even a gunshot. Sound can also be used to classify and characterize complex or heterogeneous acoustic scenes such as a busy street, a quiet park, or a noisy factory. Sound can be used to play out vivid audio-visual scenarios, be it realistic or creative, like scenes of a movie, video game, or virtual reality environment. It could also express the context and tone of a conversation, music performance, or sports game.

Sound-based scene understanding can be beneficial in many applications, such as robots, smart devices, and intelligent systems in which artificial agents interact with real, naturalistic environments. For example, scene understanding with sound will underlie advances in human-computer interaction through enabling possibilities for artificial agents to communicate, respond, and adapt to humans' speech and general sounds, such as voice assistants and speech recognition and sound synthesis systems. This may enhance scene understanding with sound to make artificial agents be able to assist, protect, and alert humans in performing their partial labor of navigation, surveillance, emergency systems, and so on, to increase both accessibility and safety. It will further support education and entertainment by providing scene understanding with sound to help artificial agents produce at will a good deal of audio-visual content and experience, from educational games to music generation systems to virtual reality systems. Sound-aware scene interpretation can also help in scientific discovery and further innovation by allowing artificial agents to analyze, model, and understand natural and artificial phenomena and processes—bioacoustics, environmental monitoring, and sound engineering.

Scene understanding with sound using artificial intelligence techniques poses many challenges and opportunities for research and development. Some of the main challenges are related to the diversity and complexity of sound sources and events. In addiction, the variability and ambiguity of sound perception and interpretation, together with the scarcity and quality of sound data and annotations, have a huge impact on the design of learning-based approaches. Moreover, the alignment and synchronization of sound and vision, and the consistency and coherence of sound and vision make multimodal architectures complicated.

One of the significant challenges that computational artificial intelligence techniques address with respect to scene understanding with sound is the complexity of computational models and methods. That is, deep learning models that have managed to reap impressive performance, namely, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Network (GAN), on many varied tasks and datasets are voracious consumers of data, memory, and processing power while being trained and run. This limits their applicability and scalability in real-world scenarios, where resources and time are usually limited. Additionally, the complexity of the models might lead to overfitting, generalization problems, and interpretability issues, which would lower the dependability and usability of the model. Therefore, in this thesis, we put attention on the complexity of our models and balance them with performance and quality. We have applied techniques like model compression, pruning, quantization, regularization, and distillation to develop smaller, faster, and more energy-efficient models that retain or sometimes improve in accuracy and diversity. Measures of model complexity, efficiency, or trade-offs are applied in this context. We show, at the same time, that our models and methods provide state-of-the-art results on a variety of tasks and datasets, with a low level of complexity and efficiency. We believe this to be an important and very promising direction for the scene-understanding artificial intelligence technique to develop models and methods that can operate in an accessible, scalable, and reliable mode in natural and realistic environments.

1.2 Understanding the environment using sound

The understanding of the environment using sound alone is very enlightening in terms of how humans and machines understand their environment. Actually, sound in itself is a very rich and wide-ranging medium that conveys much about the environment. It carries information about the presence of objects, types of activities, features of spaces, and even the emotional tone of an environment. The human mind is hard-wired to interpret these audio signals; therefore, replicating this in machines would usher in a new paradigm in the field of artificial intelligence.

Unlike visual information, which may be blocked by physical barriers or darkness, sound is able to go through walls, around corners, and in complete darkness. Hence, it forms a very rich source of information in learning about an environment for which visual data might be unavailable or insufficient. For example, take a city street: the din of the car horns and the people walking by is kind of a live soundtrack that will guide you through the level of traffic and the pedestrian density and other hazards—no vision necessary. Similarly, sounds in a forest—like rustling leaves, chirping birds, or running water—carry information on what type of vegetation, the presence of wildlife, and how far one is from a water source.

One of the fundamentals in the environment sound analysis field is acoustic scene classification, and it tries to reproduce human understanding within machines. AI models can be trained on patterns that exist in audio data; hence, such systems are to be developed which can identify and categorize different environments—be it a quiet library, a bustling market, or a peaceful park—only based on sound. This is not some sort of theoretical ability but one that is finding its applications in many domains, from smart cities to autonomous vehicles and personal assistants, which have the potential to learn by themselves in order to adapt according to the context.

Artificial intelligence, especially by deep learning techniques, has revolutionized the way scene understanding is based on sound. Traditional approaches in the processing of audio typically involve handcrafted features and domain-specific knowledge for the estimation of meaningful information from the sound signals. These methods are sometimes useful, but they tend to face the large variability and complexity of real-world sound scenes.

On the other hand, deep learning offers a much more powerful and flexible approach. AI systems can be seen as automating the learning of complex patterns and relationships in the audio data. Convolutional neural networks, recurrent neural networks, and more recently, transformer models have all been applied to scene understanding. These models can instill both the temporal and spectral characteristics of sound; therefore, such models can identify very fine-grained cues that might otherwise be lost if the traditional signal processing techniques were effective.

In addition to recognition of environments, AI techniques can also make further inferences about the context of the environment. This is done through estimation of sound source distances, detection of anomalies in sound patterns, and even prediction of future auditory events. Such multi-dimensional understanding of sound empowers machines with decisions driven by their auditory perception, moving us to a future where AI machines could easily interact with the world in a more human-like approach.

These developments in AI further increase the opportunities for sound-based scene understanding, but they also bring huge challenges. One major challenge is the inherent difficulty in deep learning models; while they are potent models, they are very data- and computationally demanding. In fact, this need for huge labeled datasets increases in the audio domain, which usually becomes time-consuming and costly.

Moreover, deep learning models are still seen as *black boxes* because it is not very clear how they make their decisions. It could be a drawback for critical applications where reasoning about a model's prediction is as important as the prediction itself. Research on the development of explainable AI techniques and on even more effective model architectures is thereby of high importance and accordingly plays an active role in the research landscape.

Also, environmental sound usually has a complex and unpredictable nature. Background noise, overlapping sounds, and changing conditions of recordings—they all cause the degradation in the performance of AI models. The development of systems that are robust to these kind of variations stands out as the key challenge researchers are actively trying to overcome.

The complexity in both model architecture and required data of deep learning models is a double-edged sword. On one hand, this can make very sophisticated systems possible; on the other hand, it opens the way to large challenges regarding computational efficiency, data requirements, and model interpretability.

In the following Sections, we will go in more detail regarding this complexity problem of deep learning in developing AI systems for sound-based scene understanding. We will see trade-offs between model accuracy and efficiency, study the contribution of model optimization techniques, and how some of these challenges can be mitigated by new advances in AI. The process of these issues will be understood and dealt with on the way to make sound-based AI systems powerful but practical for real-world applications.

1.3 Low-complexity approaches for fast training and inference

The evolution of deep learning models escalated to new dimensions of complexity with very brilliant achievements in all domains, especially sound-based scene understanding, while AI goes on improving. Their rising model complexity, however, comes along with severe side effects: high computational load, along with long training time and powerful hardware requirements. All these factors make deployment not so easy for AI methods within real-world applications, mostly with respect to resource-constrained application scenarios where efficiency and speed are crucial. Need for designing intrinsically low-complexity models

Most real-world applications, such as real-time sound recognition in mobile devices, autonomous systems, or edge computing, require efficient and effective AI models in the first place. Instead of doing this post hoc with reduction techniques, it is much more efficient and more sustainable to design models from scratch that are intrinsically low in complexity. This design philosophy guarantees that models are intrinsically amenable to fast training and inference without giving up much in the way of accuracy or massive computational resources.

The main rationale for developing low-complexity models by design is to enable AI systems that can be deployable on high-performance servers all the way down to resource-constrained edge devices. Building simplicity and efficiency into the initial model design guarantees that these systems are not only powerful but also practical for real-world use.

The efficient design would then have to take into consideration efficiency at each stage of model development. This field has its focus on the selection and optimization of neural architectures that can intrinsically balance performance with computational efficiency, getting rid of the need for extensive post-training modifications.

The design can be described by the following steps:

- **Efficient Architecture Selection.** At the least, the selection of a neural architecture itself has a lot to say in deciding the complexity of the model from scratch. Some architectures, such as MobileNet, SqueezeNet, and EfficientNet, have been designed to be efficient in performance and to keep computational demands at their minimum. These models have been designed with such techniques as depth-wise separable convolutions and compound scaling, making it possible to achieve higher accuracy with fewer parameters and less resource consumption.

- **Layer and Operation Optimization.** Designing low-complexity models entails not only keeping the parameters at a minimum but also taking great care in regard to the type of layers and operations that are used with networks. For instance, replacing conventional convolutional layers with the lightweight counterparts or reducing the number of layers and parameters without any performance degradation to a meaningful extent would bring down the computational footprint of the model drastically. The aim is at creating a network that has been tailored to that particular and specific task at hand; removing complexities not required. Another important factor in low-complexity model designing is the tailoring of the architecture as required by the task. For example, certain audio features will be more relevant in sound-based scene understanding compared to others. Thus, by focusing the design of the model on such relevant features and avoiding those irrelevant, we could make the model low in complexity without losing its accuracy.

- **Algorithmic Efficiency.** Going beyond the architecture itself, one can optimize the efficiency of algorithms used for training and inference. It involves choosing lightweight activation functions, dropping precision wherever possible in operations, and using efficient data-handling techniques. All of these make sure that not only will the model's complexity be kept low because of its structure, but in actual training and deployment, as well.

Designing low-complexity models by design requires making a delicate balance between simplicity and performance. A sophisticated enough model to do the job efficiently needs to be devised, yet simple enough that it is train and deploy in haste, even on limited hardware. This is attained through targeted decisions at the model design stage itself, with a focus only on core functionality and efficiency while side-lining any accidental complexities.

By focusing on intrinsic low-complexity design from the outset, we enable Artificial Intelligence (AI) systems with the intrinsic ability for speed in training and inference. This will then make the process of development more direct and the models obtained at the end more robust and versatile across different deployment scenarios, with less reliance on post-hoc optimization strategies.

Now, as we close off the discussion on low-complexity approaches for fast training and inference, it becomes very important to have designs of efficient models right from the beginning in order to see a wide diffusion of AI into practical applications. That is to say, efficiency intrinsic to model design guarantees that AI systems will be powerful, accessible, and scalable for real-world challenges.

1.4 Scope and objectives

The main objectives of this thesis are the following:

- **$\mathbf{O_1}$**. Design and implement AI models for the acoustic scene classification task that will correctly classify numerous acoustic environments from sound only, optimising both for performance and computational efficiency.

- **$\mathbf{O_2}$**. Investigate low-complexity neural architectures that enable fast training and inference with high accuracy in tasks related to the understanding of scenes from sounds.

- $\mathbf{O_3}$. Improve AI systems in the perception and interpretation of complex soundscapes involving scenarios with overlapping sounds and changing environmental conditions.

- $\mathbf{O_4}$. Implementation and evaluation of AI techniques for the estimation of the distance of sound sources for more complete scene understanding.

- $\mathbf{O_5}$. In the context of device resource constraints, develop and optimize AI models that enable real-time sound recognition on, for example, mobile phones and edge computing platforms.

- $\mathbf{O_6}$. Deep learning's model complexity versus performance trade-offs has to be reviewed and addressed to come up with models efficient enough yet effective for practical applications concerning sound-based scene understanding.

1.5 Thesis outline

This dissertation is composed of six Chapters (as described in Figure 1.1) and four Appendices. The major contributions of this thesis on the research objectives are described from Chapter 3 to Chapter 6, covering most of the author's publications which are listed in Appendix B. In the following, we provide a brief description of each Chapter, illustrating contributions, achievements, and advances with respect to state-of-the-art approaches.

In Chapter 2 a comprehensive background necessary for understanding methodologies and techniques for sound event recognition in particular with regard to artificial intelligence are provided. It starts by providing some background knowledge of the sound, audio signal, and signal processing, in particular, room acoustics and audio representation, including some of the basic techniques in signal processing. These basic concepts are very important for a further understanding of how the audio data is transformed and analyzed before actual recognition techniques.

The Section also discusses the application of artificial intelligence in the recognition of audio events. It describes the theoretical aspects of the AI sound recognition process, including a few methodologies of supervision that the training of an AI model can use. In more detail, the application of neural networks is explained to determine acoustic events and points out the most used datasets for training and evaluation. Besides that, it illustrates the importance of the loss function during optimization. It also introduces several methods of performance assessment of models. This section provides

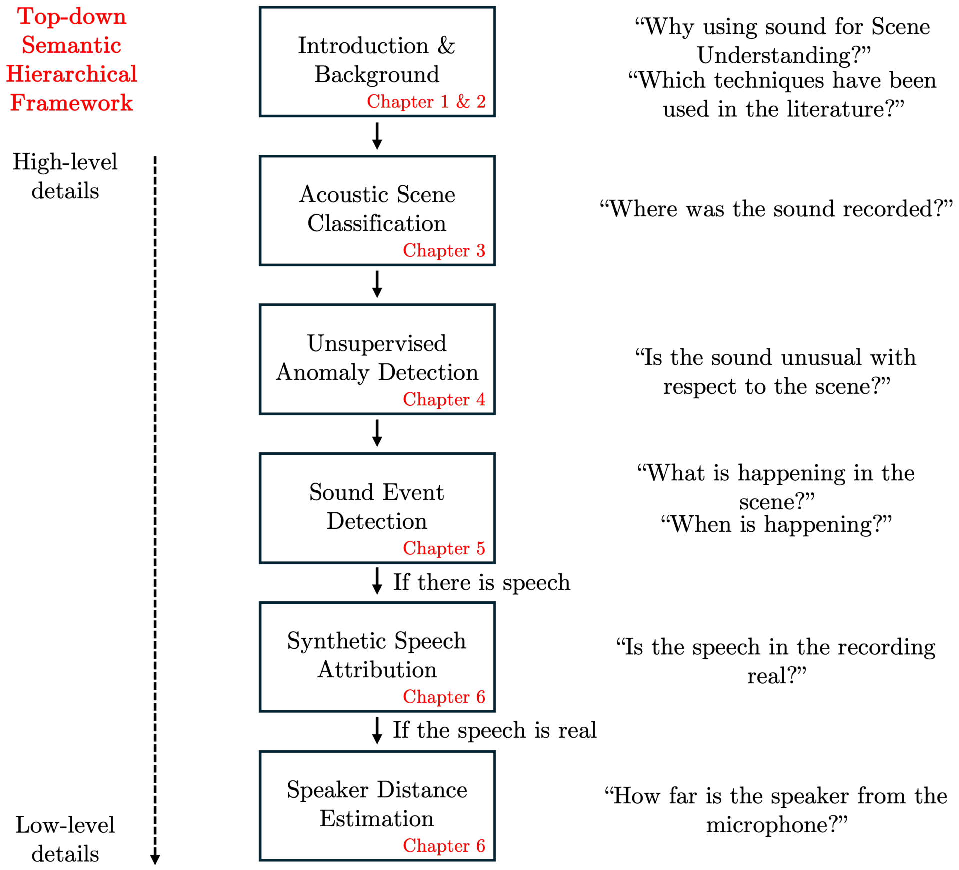

Top-down
Semantic
Hierarchical
Framework
High-level details
Low-level details
Introduction & Background
Chapter 1 & 2
Acoustic Scene Classification
Chapter 3
Unsupervised Anomaly Detection
Chapter 4
Sound Event Detection
Chapter 5
If there is speech
Synthetic Speech Attribution
Chapter 6
If the speech is real
Speaker Distance Estimation
Chapter 6
“Why using sound for Scene Understanding?”
“Which techniques have been used in the literature?”
“Where was the sound recorded?”
“Is the sound unusual with respect to the scene?”
“What is happening in the scene?”
“When is happening?”
“Is the speech in the recording real?”
“How far is the speaker from the microphone?”

Figure 1.1: Visual explanation of the research context of the Thesis.

an extensive background that will help the reader in appropriately understanding and handling the advanced topics within the later chapters of the thesis.

In Chapter 3 deals with ASC as one of the fundamental tasks in machine listening and audio signal processing. This chapter presents an insight into how an ASC system is designed to automatically detect and classify the environmental context such as distinguishing different sounds recorded in an airport, bus, or metro. The Chapter gives the important focus that it is necessary to develop robust models by using various audio data, captured in a wide variety of urban environments and with different devices, such that generalization across different acoustic conditions can be attained.

It includes the following major contributions: a novel use of Chebychev moments for audio classification on constrained hardware; a low-complexity neural network designed based on the Chebychev moments; and an advanced deep learning approach exploiting an attention module with a Wavegram representation in order to enhance the power of the feature discrimination, together with an algorithm for multi-iteration fine-tuning that is designed to improve model generalization. The contributions target domain adaptation and degradation in performance due to changes in recording conditions, and this work attempts to augment real-world effectiveness in ASC systems.

Chapter 4 focuses on how to tackle some of the challenges of UASD, i.e., in cases where labeled data is minimal or not available, such as industrial varieties or environmental noise monitoring. Specifically, the Chapter considers the challenges of modeling *normal* behavior in dynamic acoustic environments, where factors such as operational changes and environmental noise can drastically vary. It examines state-of-the-art approaches, including reconstruction-based and classification-based methods in UASD, with further consideration for aspects related to computational efficiency and model interpretability.

The attention module developed in the Chapter enhances anomaly detection through significant time-frequency pattern focusing, the use of separable convolutions for reducing model complexity, and a statistical analysis of attention maps for an understanding of the cues for detecting anomalies. These are all innovations meant for enhancing the performance and efficiency, and providing insights into the UASD systems for real-world applications.

In Chapter 5 a SED system for the identification and localization of anomalies in audio clips, detecting event type and their exact onset and offset times, is developed. Unlike the traditional classification of audio, SED requires the identification of an event precisely both in terms of type and exact timing-both of which current models, including those trained on large-scale general purpose audio datasets, cannot achieve, especially when the operating conditions are noisy.

The solutions for these problems are shown in Chapter 5, together with a new SED model consisting of a CNN with an incorporated Atrous Spatial Pyramid Pooling (ASPP) module, called AuSPP, which is designed and optimized for recognizing safety-threatening audio events-like gunshots or shouting-within public transportation settings. Since there is a lack of specialized datasets in this field, the Chapter introduces a new dataset, Sound Event Detection Dataset On Bus (SEDDOB) for sound event detection in noisy bus environments.

The performance of this system, measured in terms of recall and F1-Score metrics, is more accurate compared to the state-of-the-art. The proposed model can also be adapted according to the time resolution and anomaly classes. Our major contributions include: better spectrogram design by using ASPP, a lightweight yet customizable SED system, and developing a specialized SED dataset for bus environments.

In Chapter 6 an extension of the work on SED is proposed by addressing the task of *Continuous Speaker Distance Estimation.* Following the detection of speech events in an audio scene, the estimation of the distance of the speaker from the microphone becomes an important task to gain deep insight into the spatial dynamics of the environment. First, the problem of identifying whenever a speech is real or fake is considered. Recently, generative deep learning architectures raised a growing concern about the deep-fake problem, regarding fake audio. Deepfakes are synthesized by means of AI algorithms - such as GANs, CNNs, and Deep Neural Networks (DNNs) - to generate artificial media contents that are difficult to distinguish from real ones. In this way, this technology could be used to implement attacks against persons and institutions. Therefore, interest has spread from generating to recognizing deepfakes, creating huge interest in this research community. Moreover, knowledge of the synthesis method of a deepfake audio can reveal some information about the forger himself. Despite its importance, this problem is still in an embryonic stage.

The next step involves the definition of a baseline for single-channel speaker distance estimation, that is a Convolutional Recurrent Neural Network (CRNN). Particular attention is paid to the problems arising due to modified acoustic conditions and how these methods can be adapted to a variety of real-world situations.

Finally, in Chapter 7 an overall summary of the thesis and its main conclusions are drawn, together with the main contributions, and a discussion is proposed about possible future perspectives of audio processing and recognition. In addition, this Thesis encompasses four Appendices. Appendix A contains additional figures for a better interpretation of the evaluation results. Appendix B lists all the publications by the author during the Ph.D. period. Appendix C details the open data and code resources. Appendix D includes all the acronyms used in this Thesis.

Chapter 2

Background

2.1 Introduction

This Chapter aims to provide a detailed basis for understanding the methodologies and techniques used in sound event recognition, especially with the use of artificial intelligence. As this thesis deals with deep learning in audio processing, it is useful to establish a basic understanding of the concept of audio signals and their processing and the role played by artificial intelligence methods in this area.

The basic principles of sound, audio signals, and signal processing are first introduced in Section 2.2. Elements of room acoustics, audio representation, and the basic techniques for processing audio signals will be introduced. These are the fundamental concepts for understanding audio data processing and analysis before progressing to powerful and advanced recognition techniques.

Section 2.3 investigates the recognition of audio events using artificial intelligence techniques. Various theoretical bases related to artificial intelligence for sound recognition are discussed, such as the types of supervision methods in an AI model training, using neural networks to identify acoustic events, and common datasets utilized for training and testing purposes.

2.2 Fundamentals of audio signal representation andprocessing

2.2.1 Helmoltz equation for acoustics

The acoustic properties of a room play a crucial role in shaping the sound that a microphone captures and analyzes. The acoustic properties of a room or environment affect the way sound waves propagate, reflect, absorb, and diffuse, thus affecting the quality and characteristics of the audio signal [1].

In general, the propagation of a sound in an environment that is bounded in all directions, i.e., in a room, can be described in closed form through wave theory. Starting from the Helmoltz equation, it is possible to solve the Partial Differential Equation (PDE) in steady-state conditions where the sound fields are time-invariant (i.e., time-harmonic) to evaluate room modes, which are specific patterns of pressure variations at different frequencies [1]. The general wave equation for sound pressure $p(\mathbf{r}, t)$ as a function of position $\mathbf{r}$ and time t is:

$$\frac{1}{c^2}\frac{\partial^2 p(\mathbf{r}, t)}{\partial t^2} = \nabla^2 p(\mathbf{r}, t), \tag{2.1}$$

where c is the speed of sound in the medium (approximately 343 ms^{-1}), and ∇^2 is the Laplacian operator, which accounts for the spatial variation of the sound pressure. For time-harmonic sound waves, where the sound pressure varies sinusoidally with time, we can express the sound pressure as

$$p(\mathbf{r}, t) \overset{\text{def}}{=} \Re\{\tilde{p}(\mathbf{r})e^{-i\omega t}\}, \tag{2.2}$$

where $\tilde{p}(\mathbf{r})$ is the complex amplitude of the sound pressure as a function of position, ω is the angular frequency of the sound wave, $\Re\{\cdot\}$ is the real part operator, and i is the imaginary unit.

Applying the temporal Fourier transform to Equation (2.1), we obtain the Helmholtz equation.

$$\nabla^2\tilde{p}(\mathbf{r}, f) + k^2\tilde{p}(\mathbf{r}, f) = 0. \tag{2.3}$$

Here, k is the wavenumber, defined as $k = \frac{\omega}{c}$. Equation (2.3) describes the variation of sound pressure in an environment or space, such as a room or concert hall. This solution allows better estimation of the interaction of sound waves at boundaries (e.g., the walls of a room). These conditions specify how waves reflect, absorb, or transmit at surfaces, and they are applied to the Helmholtz equation to yield solutions for the applied sound field in the space. In addition, the Helmholtz equation also indicates the resonant frequencies in an enclosed space. These represent frequencies of the applied sound wave, where the average levels of sound pressure will be substantially increased through the constructive interference of waves in the confines of the room. The resonant frequencies correspond to the natural frequencies where the standing waves will be formed in the room.

By solving the Helmholtz equation, it is possible to predict how the sound will behave in the environment, where augmentation or cancelation will take place, or predict the interaction of sound in the environment with other boundaries or reflective surfaces.

However, the use of the Helmholtz equation is untractable for several reasons. First, we assumed that sound is time-harmonic, that is, it oscillates sinusoidally at a fixed frequency. Sounds such as clicks, pulses, or any kind of noncontinuous noise do not have a fixed frequency in time and are not adequately represented by the Helmholtz equation. These might require shock wave analysis using time domain methods, progressing towards the full wave equation. Moreover, in real acoustical situations where there is complex noise that has a wide range of frequencies (i.e., environmental noise or music), a model needs to address multiple, varying frequencies simultaneously, which Helmholtz does not provide.

Moreover, the Helmholtz equation requires that the medium be linear and homogeneous; in particular, the presence of pressure- and amplitude-independence of medium properties, such as compressibility (relative volume change), sound speed, density, and related parameters. In many physical situations, this may not be true.

2.2.2 Acoustic reflections

The equations described in the previous Section are based on unbounded or infinite mediums such as free space; the latter is an unrealistic scenario in our daily applications. Real mediums are usually bounded, at least in part. An example is air, which is the propagation medium in a room bounded by walls, a ceiling, and a floor. When sound travels in an outdoor environment, the ground is a boundary in one of the propagation directions. Hence, sound waves do not cease when reaching the boundary edge or simply encountering an object. Sound waves interact with these mediums in ways that depend on the majority of the acoustical and geometrical properties of collided objects.

Depending on the acoustic and geometric properties of the obstacles, the sound waves will interact with them in different ways, as shown in Figure 2.1. The wave can be reflected from the obstacle, diffracted by it, or transmitted through it. During transmission, the wave may also experience refraction while passing through an obstacle and lose some of its energy inside the material.

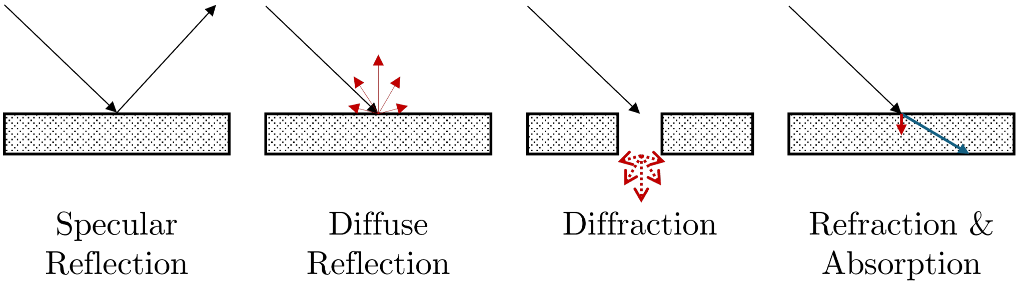


Figure 2.1: Types of possible behaviors when a sound wave hits an obstacle.

Reflections generally occur when a sound wave encounters a large surface, such as a room wall. When the wave reaches an edge or a slit in the wall, it undergoes diffraction, bending around the corners of the obstacle. The diffraction point essentially acts as a secondary source, which can interact with the original wave. The portion of energy that is transmitted into the object may be absorbed or refracted.

Two kinds of acoustic reflections may occur when sound falls on a solid surface: part of the energy in the sound is reflected *specularly*, wherein the angle of incidence is equal to the angle of reflection, while another part is reflected *diffusely*, or scattered wherein it disperses in all directions.

The relative proportions of these phenomena depend upon the acoustic and geometrical properties of the surfaces and the frequency content of the wave. In acoustics, it is customary to define certain operating points and different regimes based on the wavelength $\lambda = \frac{2\pi}{k}$, such as near-field versus far-field conditions. We can distinguish three responses of objects (or irregularities) of size d to a plane wave:

- For $\lambda \gg d$, the inhomogeneities are insignificant, and the sound wave reflects specularly.

- When $\lambda \approx d$, the irregularities interfere with the sound wave, which reflects the wave in many different directions.

- When $\lambda \ll d$ each roughness becomes a surface that specularly reflects the sound waves.

However, real-world surfaces are not perfectly flat and smooth. Other examples include rough-faceted walls and raw brick walls, while in the case of a concert hall, the entire audience area could be considered as another example. If these surface irregularities are of the same scale as the wavelength of sound, *diffuse* reflections happen.

The acoustic ray of a plane wave can also be imagined as a packet of rays that move in unison. If such a packet impacts a rough surface, each individual ray is reflected in some different direction. This gives rise to the phenomena named *scattering*. It creates a very large number of new rays and smears them out uniformly in the original half-space. The carried intensity in each outgoing ray depends on the angle and can be modeled to a good approximation by Lambert's cosine law, originally developed to describe optical diffuse reflection.

Added energy reflected can be calculated a-priori by the scattering coefficient of the material of the reflecting surface, or it is determined a-posteriori by the diffusion coefficient, i.e., the ratio of the energy of specular reflection to totally reflected energy.

This is because, with sound diffraction, it occurs at the edge of a limited surface. For instance, this behavior happens when a sound passes a corner or a door opening. When it reaches an edge of a reflector, the wave diffracts around the back of it. These diffraction waves that occur around an edge of a semi-infinite reflector will enable sound to travel into areas *behind* the reflector. This physical effect is naturally exploited by the human ear as part of its approach to sound source localization.

2.2.3 Elements of room acoustics

Room acoustics examines how acoustic waves propagate within an enclosed space bounded by surfaces such as walls and floors and how these waves interact with those surfaces. From a mathematical point of view, it is possible to analyze the propagation of the sound by solving Equation (2.3).

From now on, this thesis will follow a discrete-time signal processing notation, and any room can be acoustically described by its Room Impulse Response (RIR). Any two points in an enclosure can be denoted as the input and output of a Linear Time-Invariant (LTI) system. Figure 2.2 shows an example of a RIR described as

$$h[n] \overset{\text{def}}{=} h_d[n] + h_e[n] + h_l[n]. \tag{2.4}$$

Except for trivial cases, calculating RIRs in closed form is a complex task. As a result, numerical solvers or approximate models, such as wave-based, geometric, and hybrid simulators, are typically used.

As depicted in Equation (2.4), a RIR $h[n]$ can be decomposed into three main components:

- the direct path $h_d[n]$ of the sound wave from the transmitter to the receiver. It is equal to the free-field sound propagation.
- the early reverberation (or also early reflections or echo) $h_e[n]$ consists of a few distinct reflections, typically originating from the surfaces of the room. These reflections are usually sparse in the time domain and have a greater prominence of amplitude compared to the reflections that occur later.
- the late reverberation $h_l[n]$ encompasses numerous reflections that occur at the same time with energies that decrease exponentially. It gives the perception of the spaciousness and the features of the material that exist within the room [2]. This element of the RIR is part of the listener envelopment, which is related to the immersiveness of the sound field. The region is mainly characterized by the sound diffusion, which in turn is affected by the roughness of the surfaces.

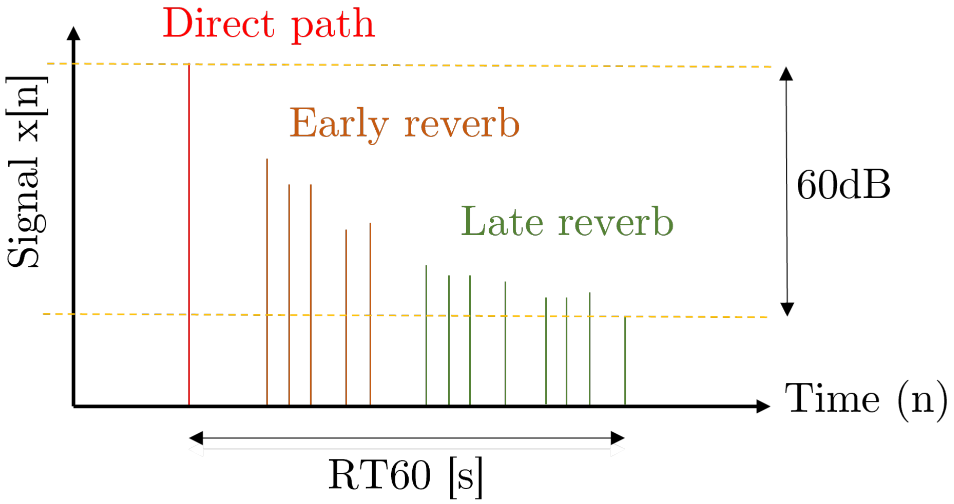


Figure 2.2: Illustration of the RIR, showing the direct path, early reflections (early reverb), and late reflections (late reverb).

Focusing on early reverberation cues, their effects have been studied in the state-of-the-art:

- The precedence effect occurs when two correlated sounds are perceived as a single auditory event [3]. This typically happens when the sounds reach the listener with a delay of 5 to 40 milliseconds.

However, the spatial location perceived from the first-arriving sound dominates, effectively suppressing the location of the lagging sound. This phenomenon enables humans to accurately pinpoint the direction of the primary sound source, even in the presence of strong reflections.

- The *comb* filter effect refers to a change in the timbre of perceived sound, known as coloration. This occurs when multiple reflections arrive in a periodic pattern, causing constructive or destructive interference. This phenomenon can be effectively modeled using a comb filter [4].

- *Apparent source width* is the audible impression of a spatially extended sound source [5]. By the presence of early reflections, the perceived energy increases, providing the impression that a source is larger than its true size.

- Distance and depth perception provide the listener with cues about the 3D location of the source. A fundamental cue for distance perception is the DRR, i.e., the ratio between the direct path ratio and the remaining portion of the RIR

$$\mathrm{DRR} \overset{\text{def}}{=} 10\log_{10}\frac{h_d^2[n]}{(h_e[n]+h_l[n])^2}. \tag{2.5}$$

The audio recording $y[n]$ obtained from a microphone placed in a room with RIR $h[n]$ can be retrieved by convolution

$$y[n] = x[n] * h[n] \overset{\text{def}}{=} \sum_{m=-\infty}^{\infty} x[m]h[n-m], \tag{2.6}$$

where $x[n]$ is the sound produced by a source.

Another acoustic characteristic of rooms is the reverberation time. Specifically, it measures the time that takes the sound to “fade away” after the source has ceased to emit. As shown in Figure 2.2, the reverberation time RT_{60} is the required time of the sound wave's energy to decay of 60dB. This value depends on the size and absorption level of the room (including obstacles), but not on the specific positions of the source and the receiver. Real measurements of RIRs are affected by background noise. As a consequence, it is not always possible to consider a dynamic range of 60dB, i.e., the energy gap between the direct path and the ground noise level. In this case, the RT_{60} value is approximated with other methods, such as Sabine's formula

$$\mathrm{RT}_{60} \approx \frac{0.146 V_{TOT}}{\sum_i \alpha_i A_i} \quad [\mathrm{s}], \tag{2.7}$$

where V_{TOT} [m^3] is the volume of the room, α_i [$\frac{\mathrm{m}}{\mathrm{s}}$] is the absorption coefficient of the surface with area A_i [m^2].

2.2.4 Representations of audio signals

In the previous Section, the principles governing sound propagation from the source to the microphone have been explored. A single-channel raw audio signal represents the variations in pressure over time on the microphone membrane and is mathematically expressed as the continuous function

$$\hat{x} : \mathbb{R} \to \mathbb{R} \tag{2.8}$$

$$t \to \hat{x}(t). \tag{2.9}$$

If an audio signal is multichannel, it is denoted as the vector $\hat{\mathbf{x}}(t) = [\hat{x}_i(t), i = 1, \ldots, c]$ with c channels. For example, stereo recordings are composed of two channels, i.e., left and right, and they can be defined as $\mathbf{x_s}(t) = [\hat{x}_l(t), \hat{x}_r(t)]$. In this Thesis, if not mentioned, all the audio signals are single-channel.

To process audio signals by computers, they require sampling and quantization of $\hat{x}(t)$, yielding a finite time-series audio signal

$$x \in \mathbb{R}^{1 \times N}, \tag{2.10}$$

where N is the number of samples with sampling frequency f_s in Hertz. The selection of the sampling rate f_s depends on the specific application, balancing computational power with processing and rendering quality. Historically, the two standard values were 44.1 kHz for music distribution on CDs and 8 kHz for early speech communication. Today, multiples of 8 kHz, such as 16, 48, 96, and 128 kHz, are commonly used in audio processing.

While the raw audio signal captures the amplitude of sound over time, its spectrum represents the sound as a function of frequency. In more detail, signals in this domain are represented as a combination of sinusoids based on their frequencies. This transformation is accomplished using the Fourier transform ($\mathscr{F}$) which projects a continuous-time-domain square-integrable signal $\hat{x}(t)$ onto a space defined by continuous-frequency complex exponentials

$$\hat{X}(f) = \mathscr{F}\{\hat{x}(t)\} \overset{\text{def}}{=} \int_{-\infty}^{+\infty} \hat{x}(t) e^{-i2\pi f t} \,\mathrm{d}t \in \mathbb{C}, \tag{2.11}$$

where $f \in \mathbb{R}$ denotes the frequency in Hertz. In this domain, linearity and the convolution theorem are defined as follows:

$$\sum_i \hat{x}_i(t) \overset{\mathscr{F}}{=} \sum_i \hat{X}_i(f) \quad \text{(Linearity)} \tag{2.12}$$

$$\hat{x}(t) * \hat{h}(t) \overset{\mathscr{F}}{=} \hat{X}(f)\hat{H}(f) \quad \text{(Convolution theorem)} \tag{2.13}$$

However, it is not possible to accomplish the continuous Fourier transform in a digital environment. Hence, in the case of discrete and finite-time signals, the Discrete Fourier Transform (DFT) is performed instead

$$X[k] = \mathscr{F}_{\text{DFT}}\{x[n]\} \overset{\text{def}}{=} \sum_{i=-\infty}^{+\infty} x[n]e^{-i2\pi k \frac{n}{F}} \tag{2.14}$$

where $k \in \{0, \dots, F\}$ denotes the frequency bin and F is the total number of bins. Albeit the difference of domain, the linearity and convolution theorem still hold using the DFT, but in discrete-time signals, the continuous convolution is substituted with its circular version.

However, both time and frequency representations alone are poor concerning the amount of information they encompass. Pieces of information are encoded in the evolution of frequencies and their amplitude over time, as we can inspect in Figure 2.3. Time-frequency representations are an adequate representation of sound to consider both temporal and spectral characteristics of sound simultaneously. An approach that is frequently used in signal processing to examine how a signal's frequency content changes over time is the STFT. Differently to the conventional Fourier transform, which provides an overview of a signal's frequency components, the STFT divides the signal into smaller time intervals and applies the Fourier transform to each one separately. This is especially helpful for evaluating non-stationary signals, such as audio recordings, whose frequencies fluctuate over time, as it yields a time-frequency representation that discloses the signal's spectral and temporal properties.

To compute STFT of $x[n]$, let $w : [0, \dots, N-1] \to \mathbb{R}$ be a window function (usually Hann) of N samples and $H \in \mathbb{N}$ be the hop size which determines the number of overlapped samples between time segments. Then, the discrete STFT $X_{\text{STFT}} \in \mathbb{C}^{M \times K}$ of the input signal $x[n]$ is given by

$$\text{STFT}\{a[n]\}[m, k] \overset{\text{def}}{=} \sum_{n=0}^{N-1} a[n]w[n-k]e^{-i2\pi mn/N_{\text{DFT}}}, \tag{2.15}$$

where $m \in [0, \dots, M-1]$ and $k \in [0, \dots, K]$ are the time and frequency

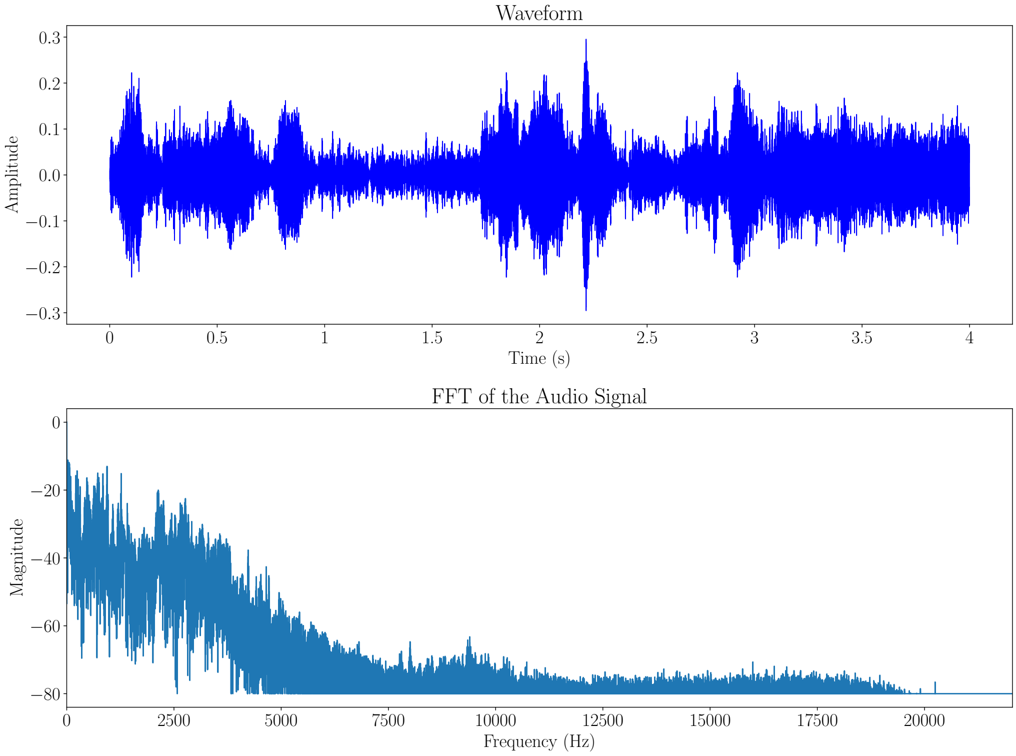


Figure 2.3: Time and spectrum of an audio signals in which street music is present. Audio sample from UrbanSound8k.

bins, respectively. The number $M = \lfloor \frac{L-N}{H} \rfloor$ represents the maximum time frame index in which the input signal $x[n]$ is nonzero. The maximum frequency index $K = \frac{N}{2}$ represents the Nyquist frequency $\frac{f_s}{2}$. To emphasize time-frequency patterns, log-spectrum is usually employed as $20 \log_{10} ||\text{STFT}\{a[n]\}||$, where $|| \cdot ||$ denotes the norm operator. In this thesis, all the time-frequency representations' sizes are denoted using T and F for the number of time and frequency bins, respectively. Similarly, each bin is denoted with t and f.

It is important to note that the STFT has linearly distributed frequencies. This representation is useful when working with speech data. However, this characteristic does not mimic the human auditory system, failing to capture the meaningful features of the audio signal for human perception.

Using power-spectrum smoothing is one way to solve the problem. One possible application for this would be a triangle-shaped Finite Impulse Response (FIR)-filter. The average of the power surrounding a given frequency is evaluated, giving nearby frequencies more weight than distant ones. Next, a triangle-shaped selection of the weighting parameters is made. In order to ensure that the number of samples collected and the amount of smoothing are equal, the smoothed samples are computed at intervals equal to half the triangle's width. Connecting multiple FIR filters yields a *filterbank*.

The Mel filterbank is one of the most used frequency projectors in the context of audio processing. Specifically, it converts linear frequencies into log-scale, mimicking the Human Auditory System (HAS), using the mapping

$$f_{\text{Mel}} = 2595 \log_{10}(1 + \frac{f_{\text{STFT}}}{700}). \tag{2.16}$$

In this Thesis, the operator $H_{\text{Mel}_b}\{\cdot\}$ denotes the Mel filterbank with b filters. A single triangular filter $H_k(f)$ at index k can be defined as:

$$H_k(f) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } f < f_{k-1} \text{ or } f > f_{k+1} \\ \frac{f - f_{k-1}}{f_k - f_{k-1}} & \text{if } f_{k-1} \leq f < f_k \\ \frac{f_{k+1} - f}{f_{k+1} - f_k} & \text{if } f_k \leq f < f_{k+1} \end{cases} \tag{2.17}$$

where f_k is the Mel frequency of the k-th filter. The output of the Mel filterbank is also denoted as the Mel spectrogram. This representation is extensively used for SED, in conjunction with MFCC, which are obtained by performing a Discrete Cosine Transform (DCT) on the Mel spectrogram. It has been shown that this procedure permits the reduction of the autocorrelation between lower and higher frequencies. This property helped to use MFCC in speaker identification tasks. The procedure of extracting MFCC is denoted as $H_{\text{MFCC}}\{\cdot\}$

Other types of filterbanks can be employed for the application and context. For example, Bark spectrogram [6] follows a different frequency mapping with respect to Mel:

$$f_{\text{Bark}} = 13 \arctan(0.00075 f_{\text{STFT}}) + 3.5 \arctan((\frac{f_{\text{STFT}}}{700})^2). \tag{2.18}$$

The objective of Bark spectrograms is to reproduce a perceptual scale of pitches judged by listeners to be equal in distance from one another. We denote this filterbank processing as $H_{\text{Bark}}\{\cdot\}$.

GTCCs [7] are other time-frequency representations of audio signals that have been designed for non-speech audio classification. The extraction of coefficients is similar to MFCC, i.e., the use of DCT to the log-spectrum, but the impulse of a gammatone filter is a gamma distribution multiplied by a sinusoid:

$$h_{\text{GTCC}_k}(n) = n^{k-1} e^{-2\pi B n} \cos(2\pi f_c n + \phi) \tag{2.19}$$

where k is the order of the filter (typically set to 4), B is the bandwidth, f_c is the center frequency, and ϕ is the phase. This filterbank is commonly used to model the auditory filters in the human cochlea. In this Thesis, we refer to $H_{\text{GTCC}}\{\cdot\}$ the entire GTCC feature extraction pipeline.

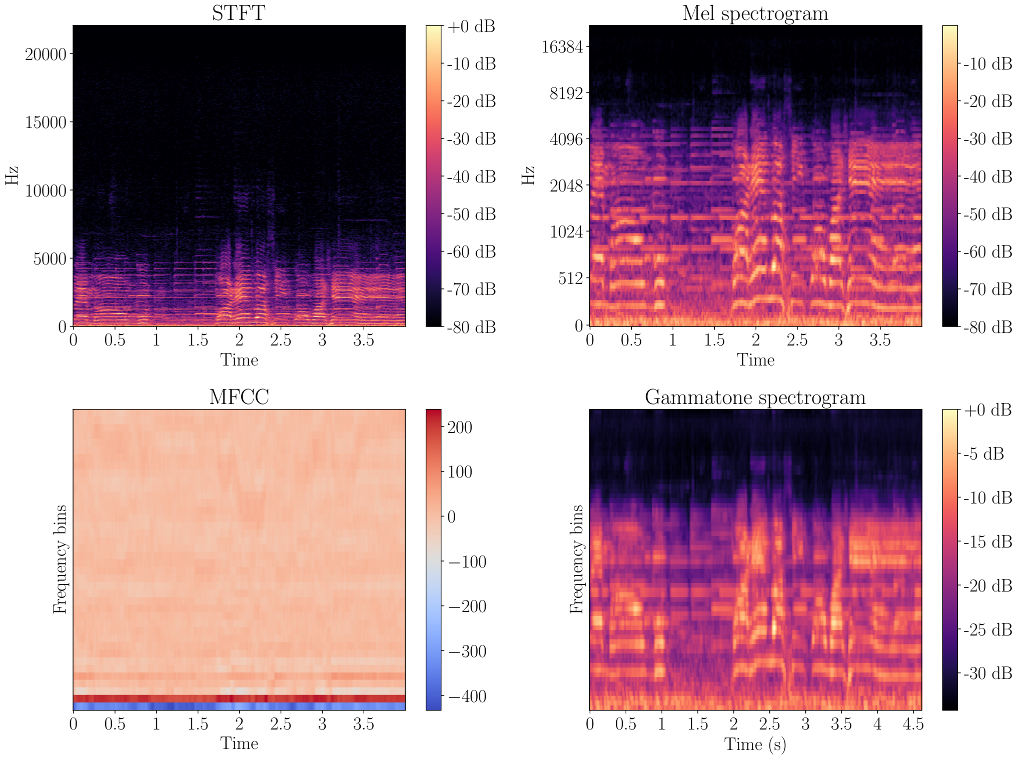


Figure 2.4: Time-frequency representations of the audio samples (the same as in Figure 2.3 encompassing street music.)

With a focus on audio processing for music analysis, Chromagram has been introduced as a time-frequency representation that highlights the intensity of each of the 12 musical pitch classes of the octave at each time frame. Specifically, the filterbank maps audio signals onto a 12-dimensional feature vector, where each dimension corresponds to one of the 12 distinct pitch classes, or chroma, of the musical octave: C, C#, D, D#, E, F, F#, G, G#, A, A#, and B. That is, all pitches that are an octave apart are mapped to the same chroma value. This representation is especially useful for music transcription, genre classification, and chord recognition. It works by dividing the frequency spectrum into a set of filters, one set corresponding to each of these pitch classes.

Clearly, each representation provides a trade-off between time and frequency resolution (the so-called *Uncertainty principle*), making one approach more suitable in some tasks than others. This principle can be inspected in Figure 2.4, mostly in the range of low frequencies of the STFT. For this reason, learnable filterbanks have been recently explored in the state-of-the-art. It is possible to learn from data a tailored time-frequency representation that is specific to the task we want to solve. We will use them in the following Sections.

2.3 Sound event recognition with artificial intelligence techniques

2.3.1 Elements of artificial intelligence theory

AI refers to a subfield in computer science wherein researchers direct their effort toward the development of systems that are competent in performing tasks normally executed by humans. These operations may include, among others, comprehension of natural language, recognition of patterns, decision-making, problem-solving, and learning from experience.

Among the core elements of AI, Empirical Risk Minimization (ERM) emerges as one of the most basic ideas within the theory of machine learning. ERM is a principle used in the context of supervised learning to guide the training of models. The main idea of ERM is that, from a hypothesis class, it tries to find the hypothesis or model that has the minimum empirical risk, which is defined as the average loss over a given set of training data.

Consider a dataset $\mathcal{D} = \{x_i, y_i)\}_{i=1}^{n}$, where x_i is a generic input belonging to the input space $\mathcal{X}$ and y_i is its corresponding label belonging to the label space $\mathcal{Y}$. The aim is to find a function $g : \mathcal{X} \to \mathcal{Y}$ from hypothesis space $\mathcal{H}$ in such a way that it can predict the label y of new input x_i, which were not seen during the training.

To evaluate the goodness of a prediction, a cost function is employed, which is also denoted as *loss*. In this Thesis, all the loss functions are denoted with $\mathcal{L}$

Then, the empirical risk is the sample average loss over the training dataset:

$$\hat{R}(g) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(g(x_i), y_i). \tag{2.20}$$

ERM focuses on searching the hypothesis g^* that minimizes this empirical risk:

$$g^* = \arg\min_{g \in \mathcal{H}} \hat{R}(g). \tag{2.21}$$

The formal justification for ERM lies in the statistical learning theory, which sets the roots of understanding how well a learning algorithm will generalize from the training data to new data. In this regard, a crucial quantity is the *true risk* or *expected risk* $R(g)$, measuring the expectation of the loss over the full data distribution $p_{(x,y)}$:

$$R(g) \overset{\text{def}}{=} \mathbb{E}_{(x,y)\sim p_{(x,y)}} \left[\mathcal{L}(g(\mathbf{x}), y)\right], \tag{2.22}$$

where $\mathbb{E}[\cdot]$ is the *expected value* operator. Since the true data distribution $p_{(x,y)}$ is not known, it is impracticable to directly minimise the expected risk (only estimate it empirically). Instead, it is possible to minimise it by optimizing the empirical risk $\hat{R}(g)$ as calculated over the finite training set $\mathcal{D}$.

The focal problem of the field of machine learning is to enable ERM to eventually find a model whose performance generalizes well beyond the data on which it has been trained. The generalization error is the difference between the expected risk, $R(g)$, and empirical risk, $\hat{R}(g)$, present in the hypothesis set:

$$\text{Generalization Error} = R(g) - \hat{R}(g) \tag{2.23}$$

In statistical learning theory, generalization bounds are usually results exposing how the generalization error is top-bounded by one, sometimes in more complicated terms, of the properties of the hypothesis space $\mathcal{H}$. One of the most famous results in this area is the Vapnik-Chervonenkis (VC) inequality [8], which relates the generalization error to the VC dimension (a measure of the capacity of the hypothesis space) and the number of training samples n.

The way it realizes this trade-off between *bias* and *variance* underlies a lot of the practical success of ERM:

- **Bias**: This is an error in the approximation of a reality-bound problem, which might be too complex for a simple learned model. A model with high bias might underfit the data, with both high training and testing errors.
- **Variance**: The error due to the model's sensitivity to the fluctuations in the training data. A high-variance model, that is to say, a too complicated neural network, might overfit the training data, capturing noise rather than the underlying distribution. The training error would be low, but at the same time, the testing error would be high.

ERM tries to balance the two sources of error by choosing a model that minimizes the empirical risk while staying clear of overfitting. This is done, in most cases, through regularization or punitive measures using modeling of too complex models.

Regularization techniques under the ERM framework usually work to alleviate the overfitting of the hypothesis space $\mathcal{H}$ to the data. The idea is to add a penalty term to the empirical risk in a way that does not allow too complex a hypothesis space. The regularized risk is of the form

$$\hat{R}_{\text{reg}}(g) = \hat{R}(g) + \lambda\Omega(g), \tag{2.24}$$

where $\Omega(f)$ is a regularization term that measures how complex the model f can be, and λ is a hyperparameter controlling the trade-off between the empirical risk and the regularization term. Common types of regularizers include L_1 (Lasso) and L_2 (Ridge) penalties, which induce sparsity and control the magnitudes of parameters, respectively.

2.3.2 Types of supervisions

Let consider a dataset in **supervised learning** where consists of input–output pairs $\mathcal{D} = \{x_i, y_i)\}_{i=1}^{n}$, such that it denotes an input example x_i and is a corresponding output target vector y_i representing the labels associated with. In this thesis, the output target vector y normally contains the binary reference labels, $y \in \{0, 1\}^C$, of a task using a vocabulary $\mathcal{V} = \{c_i\}_{i=1}^{C}$ made up of C classes. A value of 1 in y indicates that class c_i is present in the input example x, while a value of 0 indicates absence. In the case of multi-class problems, labels are represented as one-hot vectors, where only one element may be equal to 1. In contrast, multi-label problems are represented by labels as multi-hot vectors, allowing more than one element to be equal to 1. Hence, we will refer to such input examples as input representations.

In **unsupervised learning** context, instead, the dataset is a collection of input examples with no corresponding output target vectors, i.e., $\mathcal{D} = \{x_i\}_{i=1}^{n}$. Unlike in supervised learning, the objective of unsupervised learning is to find patterns, structures, or representations of data itself without any prior knowledge about the labels. Learning-based models in this field are typically trained to learn meaningful representations of the input data x_i. Representations, commonly called *latent representations* or *embeddings*, retain only the essential features or structures of the input data in a lower-dimensional space. In unsupervised learning tasks, the model may be trained to cluster, reduce dimensions, or in some other way recognize patterns in the input data x_i, thereby discovering inherent structures in the data in an unsupervised way.

In **reinforcement learning**, a dataset is composed of sequences of experiences, where each experience is defined as a tuple (x_t, a_t, r_t, x_{t+1}), hence

$\mathcal{D} = \{(x_t, a_t, r_t, x_{t+1})\}_{t=1}^{n}$. Here, x_t is the current state, a_t is the action taken by the agent in the state, r_t is the reward gained through the action, and x_{t+1} is the subsequent state onto which the said agent transitions. In reinforcement learning, the goal is to learn a policy that maximizes cumulative rewards over multiple time-steps based on interactions with the environment. This way, reinforcement learning can be seen to concern learning through trial and error, unlike getting patterns from datasets, as in the case of supervised/unsupervised learning. The agent keeps on communicating and learning through the environment, directed by the received reward signals, in the manner in which learning takes place by reinforced signals.

The current thesis deals mainly with supervised and unsupervised learning, but another interesting milestone for future research may be carried out by applying techniques from reinforcement learning to audio signal processing.

This opens up a wide vista of new possibilities for performing such tasks as adaptive noise cancellation, automatic audio enhancement, or even creative applications such as real-time audio effects generation. An example could be training an agent to optimize audio quality within dynamic environments, learning in real-time how to adapt given changes in noise conditions or user preferences.

This would also mean that the development of this direction includes the integration of reinforcement learning frameworks with audio processing techniques, and it can further result in systems capable of not only responding to auditory input but also learning over time how to predict and adapt to complex auditory environments. As reinforcement learning continues to evolve and mature, combining it with audio processing opens great opportunities for future research and practical applications.

To summarize, designing a learning-based approach requires four main components:

- the dataset $\mathcal{D}$, which is composed of input samples and, if necessary, labels;
- the model $g : \mathcal{X} \to \mathcal{Y}$ that performs the mapping from input to output space;
- the loss function $\mathcal{L}$ that permits to find the best solution g^* that minimizes the empirical risk $R(g)$;
- metrics that assess the performance of the model regarding generalization, i.e., on unseen data.

2.3.3 Audio datasets

Depending on the task we want to solve, several audio datasets have been published in recent years. With a focus on audio classification, UrbanSound8K [9] dataset comprises 8732 audio files of at most 4 seconds of duration and divided into the following 10 different classes: *air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music.* As to the ESC-50 [10] dataset, it has 2000 short clips recorded at a sampling frequency of 44.1 kHz grouped into 50 classes of various common sound events: *dog, rain, crying baby, door knock, helicopter, rooster, sea waves, sneezing, mouse click, chainsaw, pig, crackling fire, clapping, keyboard typing, siren, cow, crickets, breathing, door, wood creaks, car horn, frog, chirping birds, coughing, can opening, engine, cat, water drops, footsteps, washing machine, train, hen, wind, laughing, vacuum cleaner, church bells, insects, pouring water, brushing teeth, clock alarm, airplane, sheep, toilet flush, snoring, clock tick, fireworks, crow, thunderstorm, drinking, glass breaking, hand saw.*

TAU Urban Acoustic Scenes 2020 Mobile development dataset [11] was proposed by Tampere University for research in the area of environmental sound classification, including classifying urban areas. It includes 3600 10-second audio recordings with a sampling rate of 44.1 kHz and was recorded across various urban settings, from streets and parks to malls. Moreover, this dataset contains recordings from different mobile devices, adding real variations in the quality and conditions of the sound. Each recording is annotated with the label of the represented urban acoustic scene, thus supporting structured classification both for training and testing machine learning models. Used in DCASE challenges, this dataset has driven the development of audio-based environmental recognition systems under simulated real variability.

However, as stated by machine learning theory with the VC dimension, increasing the sample size permits training a larger model with an enhanced ability to model more complex audio events. Therefore, researchers started collecting larger and larger labeled audio datasets, with greater emphasis on the quality of labels. A major example is AudioSet [12], which is a large-scale dataset curated by Google. It includes more than 2 million human-labeled 10-second audio clips drawn from YouTube videos. Each clip is annotated with labels from a hierarchical ontology of over 600 sound categories that cover a very broad range of sounds, from everyday environmental noises to animal sounds, human activities, musical instruments, and many others. AudioSet is used by many as a resource for creating and evaluating machine learning models for the detection and classification of audio events and related tasks; AudioSet is therefore considered to be a benchmark for different kinds of applications in audio processing.

In the same direction, FSD50K [13] is a large-scale audio events dataset intended for training and testing machine learning systems in the audio event classification task. It consists of more than 50000 audio clips from Freesound, an online collaborative dataset of creative-commons licensed audio sounds. The audio is annotated with labels from the AudioSet ontology, encompassing human activities, musical instruments, animal sounds, and environmental noises. FSD50K has been used widely for benchmarking and model development in various activities involving audio classification, detection, and recognition.

In this thesis, in Chapter6, we propose an annotated audio dataset that has been recorded in a public transportation environment. Specifically, since no SED datasets for bus acoustics and anomalous events were available, we designed the acquisition of background noise in the bus, together with the injection of possible dangerous events to train a SED model in resource-constrained environments.

2.3.4 Neural networks in sound event recognition

In this thesis, most of the models $g \in \mathcal{H}$ used for Sound Event Recognition (SER) are ML classifiers and DNNs. From a mathematical point of view, a learning-based model $g_{\boldsymbol{\theta}}$ is parametrized by a set of weights $\boldsymbol{\theta}$ that are optimized to minimize the empirical risk of the training set. ML classifiers typically employ hand-crafted features, such as MFCC, zero-crossing rate, and other first and second-moment statistics for modeling and classification. However, they have been superseded by deep learning models. DNNs, instead, usually use low-level representations, such as different forms of spectrograms or even raw waveforms, instead of acoustic features used in feature engineering [14].

In this regard, CNNs are neural networks deploying convolution operations instead of general matrix multiplication at least in one layer [15]. A cross-correlation function by sliding filters (kernels) over the input data is performed through convolutional layers of a CNN, thereby producing an output known as a *feature map*. CNNs avoid some of the limitations of prior models, such as MultiLayer Perceptrons (MLPs) [15]. For instance, CNNs allow for sparse interactions because kernels are smaller in size compared to the input. This is in contrast to matrix multiplications involving interactions of all input with all output units. Another characteristic feature of CNNs is weight sharing, where each kernel weight is used at all positions of the input. This makes the processes far more efficient than a dense matrix multiplication. Another implication of this weight sharing is that convolutional layers are equivariant to translation, meaning if a pattern in a feature map shifts in the input, its corresponding feature map will also shift similarly.

Generally, convolutional layers are combined with pooling layers, as well as normalization layers (such as Batch Normalization [16]) and non-linear activation functions (like Rectified Linear Units or the more recent Exponential Linear Unit (ELU) [17]). Pooling layers serve to downsample the feature maps, which reduces the dimensionality handled by the network and enables deeper layers to integrate information over larger areas.

At the end of every CNN, the final layer is strictly related to the targeted task. The *softmax* layer is required for multi-class tasks scenario, whereas the *sigmoid* activation is employed for multi-label classification, which is also known as *tagging*.

CRNNs are a class of CNN that have attached RNN layers, such as Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), at the end of convolutional layers to aggregate the extracted features over time, thus modeling discriminative temporal structures. This type of DNN has been employed in the realm of SED, as it is possible to both extract class-wise features and identify the sound source in time.

An example of well-known 2D CNN for both audio classification and tagging are the PANNs [18], which require Mel spectrograms as input, thus time-frequency representations. A famous 1D CNN is Wavenet [19], which has been designed for audio generation tasks and classification exploiting the raw audio signal in the time domain. Regarding CRNN, in [20, 21] a CNN with two GRU layers is proposed to perform SED.

Finally, Transformer-based methods have been recently used in recognizing sound events due to their superior ability to capture long-range dependencies in audio. Traditional methods involve a convolutional neural network that, by design, operates locally. Transformers, however, use self-attention mechanisms [22] for weighing the importance of each part of the input sequence toward the goal of modeling global relationships. Transformers in SER have been employed to process sequences of audio features like Mel spectrograms, in which the mechanism of self-attention helps the model pick out relevant sound events across time. That is particularly helpful when the sound events are temporally dispersed or overlapping, scenarios difficult for a model such as CNN. Besides, transformers can handle variable-length input sequences, which makes them flexible for different tasks related to SER. A state-of-the-art Transformer-based audio classifier is AST [23], which combines a CNN and the Transformer to extract features and combine them in time, respectively.

It is important to highlight that PANNs and Transformer-based classifiers are computationally expensive, thus, it is unfeasible to deploy these models in resource resource-constrained environment, such as edge nodes in a cloud architecture. In this thesis, we mainly design CNNs and CRNNs as we focus on developing low-complexity learning-based models with low training and inference times.

2.3.5 Loss function

As previously described, the selection of the loss function $\mathcal{L}$ is fundamental for the ERM procedure to find the best hypothesis/model $g_{\boldsymbol{\theta}}$. Similarly to the dataset and the model, choosing the loss function depends on the task.

Generally, the cross-entropy loss $\mathcal{L}_{\text{CE}}$ is employed for audio classification tasks. It corresponds to minimise the statistical difference between the prediction of the model and the truth. Let $\mathbf{y_i} \in \mathbb{Z}^C$ and $\hat{\mathbf{y}}_\mathbf{i} = f_{\boldsymbol{\theta}}(x_i) \in \mathbb{R}^C$ be the ground truth and predicted labels of a single audio sample x_i using the neural network $f_{\boldsymbol{\theta}}$, respectively. Then, the error for this sample is computed as

$$l_{\text{CE}_i} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{2.25}$$

As deep learning models usually compute the error across batches, i.e., with n samples at time, the error that is going to be propagated by means of the *back-propagation algorithm* to correct the weights of the DNN is averaged as follows

$$\mathcal{L}_{\text{CE}} = \mathbb{E}_n[l_{\text{CE}_i}] \tag{2.26}$$

In the case of multi-label classification problems, e.g., audio tagging, the binary cross entropy $\mathcal{L}_{\text{BCE}}$ is used, following the same batch operation of the cross-entropy

$$\mathcal{L}_{\text{BCE}} = \mathbb{E}_n[l_{\text{BCE}_i}] = \mathbb{E}_n[y_i \ln \hat{y}_i + (1 - y_i)\ln(1 - \hat{y}_i)]. \tag{2.27}$$

The next Chapters will also describe some variants of loss function related to audio classification, such as ArcFace [24].

For regression tasks, such as distance estimation in Chapter 6 with $\{y_i, \hat{y}_i\} \in \mathbb{R}$, the most used loss function is the Mean Squared Error (MSE)

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_n[(y_i - \hat{y}_i)^2] \tag{2.28}$$

as it has been shown that it maximizes the mutual information between predicted and ground truth labels. However, other types of regression losses can be used, such as L_1, Huber loss, etc.

2.3.6 Performance assessment

Assessing the correctness of a model's prediction is a critical step in the machine learning field. Although some tasks have a straightforward assessment of performance, for instance, an audio classification performance is measured using the accuracy, some tasks require more complex and articulated indicators. In the following, we provide the metrics used in this thesis.

Audio classification

The validation of an audio classification system is usually based on the accuracy score. More specifically, let TP be the number of true positives, FP be the number of false positives, FN be the number of false negatives, and TN be the number of true negatives. The classification accuracy is evaluated as:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}, \tag{2.29}$$

which can be decomposed into Sensitivity (S_t) and Specificity (S_p):

$$S_t = \frac{TP}{TP + FN}, \tag{2.30}$$

$$S_p = \frac{TN}{TN + FP}. \tag{2.31}$$

Inspired by the state-of-the-art [25], we discard TN from Equation (2.29) and we obtain:

$$Acc_M = \frac{TP}{TP + FP + FN}. \tag{2.32}$$

To visually inspect the model's prediction and to understand the type of false positive errors, it is possible to employ the confusion matrix CF. Specifically, it is a square matrix $C \times C$ in which the accuracy metric can be obtained by

$$Acc = \frac{\sum_{i=1}^{C} \mathrm{CF}(i,i)}{\sum_{i=1}^{C} \sum_{j=1}^{C} \mathrm{CF(i,j)}}, \tag{2.33}$$

where C denotes the number of classes. It has been employed in Chapter 3 to inspect the number of correct and incorrect predictions made by the model, broken down by each class, in a domain shift scenario.

Anomalous sound detection

To evaluate the performance of a model in the unsupervised anomaly detection task, two metrics are generally computed: the AUC and the pAUC, where AUC describes an overall performance level for a binary classifier with a tradeoff between the true positive rate and false positive rate for its classification. In contrast, pAUC focuses on part of the Receiving Operating Curve (ROC) curve, specifically over a defined range of interest where low false positive rates are critical. For this task, compute the pAUC over the False Positive Rate (FPR) range $[0, p]$.

The AUC and pAUC are mathematically expressed as:

$$\text{AUC} = \frac{1}{N_- N_+} \sum_{i=1}^{N_-} \sum_{j=1}^{N_+} \mathcal{H}(A_\theta(x_j^+) - A_\theta(x_i^-)), \tag{2.34}$$

$$\text{pAUC} = \frac{1}{\lfloor pN_- \rfloor N_+} \sum_{i=1}^{\lfloor pN_- \rfloor} \sum_{j=1}^{N_+} \mathcal{H}(A_\theta(x_j^+) - A_\theta(x_i^-)), \tag{2.35}$$

where $\lfloor \cdot \rfloor$ is the flooring function, and $\mathcal{H}(x)$ is the Heaviside step function that returns 1 in the case when $x > 0$ and 0 otherwise. In the above formulae, $\{x_i^-\}_{i=1}^{N_-}$ and $\{x_j^+\}_{j=1}^{N_+}$ is the normal and anomalous test samples respectively, that are sorted in descending order according to their anomaly scores. Here, N_- and N_+ are the number of normal.

This method requires the anomaly scores of all normal test samples as a threshold. It must, therefore, also necessitate anomaly scores of all test samples rather than giving a binary decision output, as most practice goes through. This way, it allows for a detailed assessment of the model in making a classification between a normal and an anomalous sample, especially when the false positive rate needs to be kept at a minimum level.

Anomalous Sound Detection (ASD) systems applied to the real world are easily susceptible to a loss of credibility of their system if the rate of false alarms is too high: the "boy who cried wolf" problem. Thus, when operating at low FPR, it becomes of special importance to optimize True Positive Rate (TPR). Since more weight is to be given to the low-FPR portion of pROC, a pAUC with $p = 0.1$ is generally set.

In this Thesis, we employ these metrics in Chapter 4 to evaluate the effectiveness of our anomaly detection method.

Sound event detection

At the moment, a rigorous quantitative evaluation of SED systems is still not universally accepted by the research community. For this reason, the comparison between the output of the SED algorithm and the ground truth is performed also on fixed length time intervals, thus measuring segment-based metrics [26].

Segment-based metrics evaluate the system prediction and the reference in fixed short time segments. Thanks to this activity representation, it is possible to define intermediate statistics like TN, TP, FP, and FN. To evaluate the robustness of the system, an activity threshold δ is introduced to the predicted activity matrix $\hat{Y}$. In the following, we denote with *loose* threshold the case $\delta = 0.8$ and with *strict* threshold the case with $\delta = 0.9$. By applying one of the thresholds to the predicted activity matrix $\hat{Y}$, the resulting matrix is binary, and intermediate statistics can be evaluated. Precision and Recall [27] are used for measuring the effectiveness of the retrieval. For a generic i-th class event, the precision (P_i) and recall (R_i) are evaluated as:

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i}. \tag{2.36}$$

Then, class-wise Precision (P_c) and Recall (R_c) are calculated by averaging with respect to the number of class events N_{class}:

$$P_c = \mathbb{E}\left[\sum_{i=1}^{N_{class}} P_i\right], \quad R_c = \mathbb{E}\left[\sum_{i=1}^{N_{class}} R_i\right], \tag{2.37}$$

where $\mathbb{E}[\cdot]$ is the expected value. In addition, F-score can be derived as:

$$F_i = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}. \tag{2.38}$$

In the literature, two types of averaging approaches for the F-score are proposed [26].

We define the class-wise F_c as the average of all the F-scores:

$$F_c = \mathbb{E}\left[\sum_{i=1}^{N_{class}} F_i\right]. \tag{2.39}$$

We evaluate the audio-wise F-score F_a by calculating the precision P_j and the recall R_j of a j-th audio recording (ignoring the class events).

Let N_{audio} be the number of audio samples. Then, the metric is computed as:

$$F_a = \mathbb{E}\left[\sum_{j=1}^{N_{audio}} \frac{2 \cdot P_j \cdot R_j}{P_j + R_j}\right]. \tag{2.40}$$

Furthermore, we use the Error Rate (ER) metric to account for the number of wrong predictions in terms of substitution, deletion, and insertion errors. More precisely, let k be a specific time-segment, with its intermediate statistics (TP_k, TN_k, FP_k, and FN_k), in the j-th audio file with K time-frames. We can define the errors as:

$$\begin{aligned} S_k &= \min\left(FN_k, FP_k\right), \\ D_k &= \max\left(0, FN_k - FP_k\right), \\ I_k &= \max\left(0, FP_k - FN_k\right). \end{aligned}$$

The total ER of the j-th audio file is evaluated as:

$$ER_j = \frac{\sum_{k=1}^{K} S_k + \sum_{k=1}^{K} D_k + \sum_{k=1}^{K} I_k}{\sum_{k=1}^{K} N_k}. \tag{2.41}$$

Finally, ER is averaged to the number of audio files in the validation set:

$$ER = \mathbb{E}\left[\sum_{j=1}^{N_{audio}} ER_j\right]. \tag{2.42}$$

It is worth noticing that the ER is not a probability, so its value can be bigger than 1. However, thanks to the use of the activity threshold δ and the fact that a zero predicted activity matrix yields unit ER, all the approaches used for comparisons have an ER lower or equal to 1.

Speaker distance estimation

The performance evaluation of distance estimators has been carried out using the Mean Absolute Error (MAE) ($\mathcal{L}_1$) as the performance measure for the entire test dataset

$$\mathcal{L}_1(y, \hat{y}) = |y - \hat{y}|, \tag{2.43}$$

where the ground truth $y \in \mathbb{R}$ and the prediction $\hat{y} \in \mathbb{R}$ are considered. Additionally, the performance is assessed by calculating the MAE within different distance ranges. This analysis allows us to quantify the relative error of our model concerning source distance. We define the relative MAE ($r\mathcal{L}_1$), which includes the real speaker distance in the evaluation, as follows:

$$\mathrm{r}\mathcal{L}_1(y, \hat{y}) = \frac{\mathcal{L}_1(y, \hat{y})}{y} = \frac{|y - \hat{y}|}{y}. \tag{2.44}$$

For the sake of clarity and brevity, MSE has not been considered in the performance evaluation.

Chapter 3

Low-Complexity Acoustic Scene Classification

3.1 Introduction

ASC is a subproblem in the domains of machine listening and audio signal processing; its purpose is to automatically detect and classify the environmental context or scene existing in an audio signal. Specifically, the objective is to identify the type of environment in which a sound was recorded, such as *airport*, *bus*, *metro*, or other similar settings. The aim is to correctly classify the scene based on the signature and patterns that define the unique auditory characteristics of every environment [28]. ASC can be visually described in Figure 3.1.

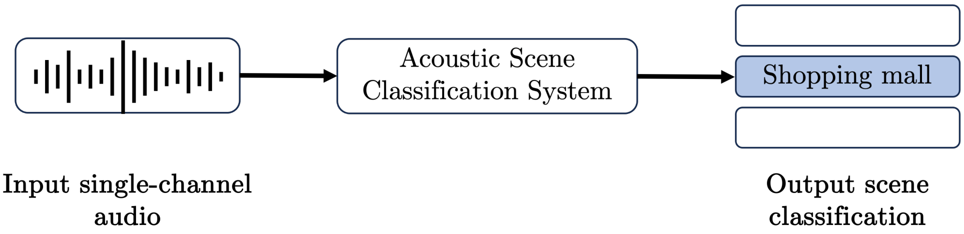

Acoustic Scene Classification System
Shopping mall
Input single-channel audio
Output scene classification

Figure 3.1: Definition of the ASC task.

The audio data used in ASC are usually gathered from different places covering several cities to obtain good sound coverage varying in their environment. The recordings are made with a diversity of devices, each differing in quality and specifications. The diversity in data collection is very important in developing robust ASC models for generalization across different acoustic conditions and device types [29]. The process of classification is based on state-of-the-art algorithms analyzing acoustic features from audio signals, which enable the system to distinguish scenes due to subtle variations of sound patterns.

For real-world applications, a classification method for acoustic scenes is expected to work in very diverse conditions, including audio captured with different devices and as short as possible inference time. The first task on low-complexity acoustic scene classification was defined in 2020 for only three classes and a single device [30], for which many submissions obtained very high performance.

This usually establishes several potential pitfalls to current systems in the adaptation to quite different tasks of ASC under inhomogeneous recording conditions, as the distributions of the data are unlike [29]. System adaptation must therefore be taken as a prime issue of high relevance in this field, since measurements from many environments can barely fit into any reasonable portable device to use, for example, adaptive noise cancellation in headphones. With this background, the aim of this study is to design a robust ASC system, which would adapt well with the variable recording conditions in these cities and different devices.

The recording device's resolution and the density in population, along with the cultural factors and local infrastructure, greatly differentiate the utilization of audio features used by ASC to derive the classes and, thus, degrades the performance. These characteristics will make the feature distributions shift under different recording conditions, where each of those conditions, like a city or a device, represents a different, distinct domain of data. Accordingly, adapting a model trained on data from one set of cities and devices to work on new cities and devices can be cast into a problem of domain adaptation. The other approach could be an increased collection of heterogeneous data to overcome the problem of domain adaptation, although, with an unpredictable test set, this would need a more strong model design to work on the matter in hand.

The contributions of this Chapter can be summarized as follows:

- The use of Chebyshev moments is implemented for the first time in the literature for audio classification in constrained hardware. Unlike other image moments [31], Chebyshev moments are only defined on a discrete set. Therefore, no approximations are required in their implementation. Moreover, their property of symmetry drastically reduces the amount of those moments, which are to be calculated, hence reducing the computational time involved and making them suitable for real-time applications.

- A novel low-complexity neural network, which solely uses Chebyshev moments to perform ASC with a lightweight CNN, is devised. The primary advantage of this method lies in the processing, which is done on the reduced-size CFD, which is obtained using the Fast Fourier Transform (FFT) efficiently, and not on the larger Mel-spectrogram. Performance is assessed through the average accuracy metric on two challenging and widely studied datasets: UrbanSound8K [32] and ESC-50 [33], with cross-validation.

- To overcome the domain shift problem, a further advanced deep learning approach with an attention module equipped with a learned time-frequency representation, called Wavegram, is proposed. This module allows the model to be attentive to only the most relevant features in the input data, providing better adaptation to various conditions of recording and the environment. Moreover, the use of Wavegram representation allows to capture of more diverse and complex time-frequency patterns, enhancing the feature set and making it more robust and discriminative as compared to the traditionally used ones.

- A multi-iteration Fine-Tuning (FT) algorithm is defined to train the model on the source domain, enhancing its generalization capability. Finally, unlabeled data is leveraged in a semi-supervised manner to further refine the model's predictions. Performance is assessed through the average accuracy metric on the IEEE ICME 2024 Grand Challenge development dataset which focuses on the domain shift problem.

3.2 Audio classification using Chebyshev moments

Although MFCC features have been in use for a long time for audio classification, attention to Chebyshev moments can be justified by the probable benefits it may offer. Although these moments proved effective in hand-gesture classification in both image and radar domains [34, 35], to the best knowledge of the authors, they have not been explored so far for the classification of audio signals. Contrary to other image moments [31], the Chebyshev moments are defined only on a discrete set; therefore, in their implementation, no approximations are needed. This fact, other than providing accuracy, makes them highly suitable for real-time applications. Moreover, their intrinsic symmetry reduces the number of moments to be computed, hence considerably reducing the overall computational time and making them an efficient alternative for real-time audio classification tasks. In this Section, a brief introduction of Chebyshev moments is presented, together with the definition of a method to classify sound using the aforementioned polynomials, MFCC, and a machine learning-based classifier. The overall framework can be inspected in Figure 3.2.

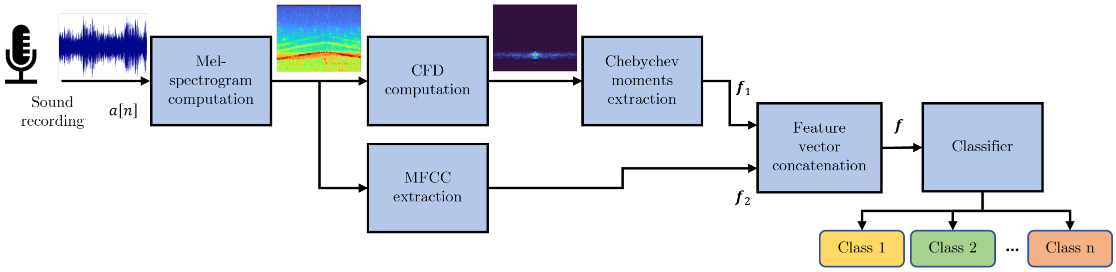


Figure 3.2: Definition of the method using Chebyshev polynomials and MFCC using a ML classifier for the identification of sound sources.

3.2.1 Chebyshev polynomials and moments

Let $f(x, y)$ be a non-negative real-defined image of size $L \times H$. The moments of order $l + h$ of $f(x, y)$ are defined as its projection on the monomials $x^l y^h$, by means of the following integral [31]:

$$M_{l,h} = \iint\limits_{\mathbb{R}^2} x^l y^h f(x, y)\, dx\, dy. \tag{3.1}$$

In general, the moments do not share orthogonality properties since their generating monomials $\{x^l y^h\}$ do not. This condition is, however verified for some widely used polynomials, e.g., Chebyshev [36]. More specifically, Chebyshev moments can be derived as the projection of the image $f(x, y)$ on the related polynomials, as in Equation (3.1), to a discrete polynomial set, that is:

$$C_{l,h} = \frac{1}{\bar{\rho}(l, L)\bar{\rho}(h, H)} \sum_{x=0}^{L-1} \sum_{y=0}^{H-1} c_l(x) c_h(y) f(x, y), \tag{3.2}$$

being $c_l(x)$ the Chebyshev polynomial of order l that can be written as

$$\sum_{x=0}^{L-1} c_l(x)c_h(x) = \rho(l,L)\delta_{l,h}, \tag{3.3}$$

with $0 \leq l \leq L-1$, $0 \leq h \leq H-1$, and $\delta_{l,h}$ the Kronecker delta function that is equal to 1 when $l = h$ and 0 otherwise. Moreover, $\bar{\rho}$ is the normalized amplitude factor given by:

$$\bar{\rho}(l,L) = \frac{\rho(l,L)}{L^{2l}} = (2l)!\binom{L+l}{2l+1}\frac{1}{L^{2l}}. \tag{3.4}$$

Next, the generalized hypergeometric function [37] of order (3, 2) is introduced, that is

$$ {}_3F_2\left(a_1, a_2, a_3; b_1, b_2; z\right) = \sum_{k=0}^{+\infty} \frac{(a_1)_k(a_2)_k(a_3)_k}{(b_1)_k(b_2)_k}\frac{z^k}{k}, \tag{3.5}$$

with $(a)_l$ denoting the Pochhammer symbol [37] given by

$$(a)_l = a(a+1)\cdots(a+l-1) = \frac{\Gamma(a+l)}{\Gamma(a)}, \tag{3.6}$$

where $\Gamma(z) = \int_0^{+\infty} t^{z-1}e^{-t}dt$ is the gamma function. Finally, the Chebyshev polynomials [36] can be expressed using the following equation

$$c_l(x) = (1-L)_l \, {}_3F_2\left(-l, -x, 1+l, ; 1, 1-L; 1\right), \tag{3.7}$$

where $x = 0, 1, 2, \ldots, L-1$.

3.2.2 Feature extraction and classifier

The CFD is devised in [38, 39] as an enhanced tool allowing to better distinguish micro-Doppler-based radar signals with respect to the use of the classic spectrogram. It consists of a DFT applied to pass from a time-frequency to a cadence-frequency domain. Performing a DFT of the spectrogram for each frequency bin allows one to obtain the cadence information, that is, the repetition cycle of each frequency involved in the original signal. As in [40], the CFD is computed from the Mel spectrogram modulus with $f = 64$ filters used in place of the spectrogram in [38, 39], that is

$$\begin{aligned}\Delta[\xi, m] &= \mathscr{F}_{\mathrm{DFT}}\{H_{\mathrm{Mel}_f}\{x[n]\}\} && (3.8)\\ &= \sum_{k=0}^{N_{\mathrm{CFD}}-1} |H_{\mathrm{Mel}_f}[k, m]| e^{-j2\pi k\xi/N_{\mathrm{CFD}}} \quad m = 0, \ldots, N_{\mathrm{DFT}}, && (3.9)\end{aligned}$$

where $|\cdot|$ denotes the modulus of its complex argument, ξ is the cadence frequency, N_{CFD} is the number of frequency bins used in the CFD computation, and N_{DFT} is the number of frequency bin involved in the Mel spectrogram computation.

The extraction of the Chebyshev moments occurs through the projection of the CFD into the orthogonal Chebyshev polynomials by means of

$$C_{l,h} = \frac{1}{\bar{\rho}(l, N_{\mathrm{CFD}})\bar{\rho}(h, N_{\mathrm{DFT}})} \sum_{x=0}^{N_{\mathrm{DFT}}-1} \sum_{k=0}^{N_{\mathrm{CFD}}-1} \bar{c}_l(m)\bar{c}_h(k)|\overline{\Delta}[k, m]|, \qquad (3.10)$$

with $\bar{\rho}$ the normalized amplitude factor, $\bar{c}_l(\cdot)$ and $\bar{c}_h(\cdot)$ the Chebychev polynomial of order l and h. Finally, $\overline{\Delta}(\cdot, \cdot)$ is the CFD normalized to be in the interval $[0, 1]$.

It is herein worth underlining that since the Chebyshev polynomials only depend on the polynomial order (a priori set) as well as on N_{CVD}, they can be a priori computed. This is compliant with real-time applications of the proposed pipeline.

The feature vector obtained from Chebyshev moments $\boldsymbol{f}_1$ is constructed as

$$\boldsymbol{f}_1 = [C_{0,0}, C_{0,1}, \ldots, C_{l,h}]^T . \qquad (3.11)$$

In addition, the MFCC are also extracted. In particular, they are obtained as the amplitudes of the DCT of the logarithm of the Mel spectrogram. Then, the feature vector $\boldsymbol{f}_2$ is constructed taking the mean value of each MFCC over time, say $\overline{\mathrm{MFCC}}$, that is

$$\boldsymbol{f}_2 = \left[\overline{\mathrm{MFCC}}_1, \overline{\mathrm{MFCC}}_2, \ldots, \overline{\mathrm{MFCC}}_{N_{\mathrm{DFT}}}\right]^T . \qquad (3.12)$$

Then, the feature vector $\boldsymbol{f}$ used to train the classifier is obtained by concatenation of the aforementioned feature vectors $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ as

$$\boldsymbol{f} = \left[\boldsymbol{f}_1^T, \boldsymbol{f}_2^T\right]^T . \qquad (3.13)$$

Finally, the audio classification is carried out by machine learning-based classifiers such as KNN and RF.

3.2.3 Experiments

The effectiveness of the proposed architecture based on the joint exploitation of both Chebyshev moments and MFCC to automatically distinguish among different audio sources is assessed in this Subsection. Tests are conducted on two publicly available databases, viz. UrbanSound8K [32] and ESC-50 [33].

Table 3.1: Mean classification accuracy (%) for each feature set on the UrbanSound8K dataset using the 10-fold cross-validation and two different classifiers, namely KNN and RF.

UrbanSound8K [32]

	KNN	RF
Baseline [32]	55.00	66.00
MFCC	37.82	50.91
pseudo-Zernike order 20 [41]	38.39	60.05
Chebychev order 10	37.37	63.65
Chebychev order 20	37.70	62.13
Chebychev order 10 + MFCC (ours)	40.40	**68.55**
Chebychev order 20 + MFCC (ours)	40.10	67.35

Then, to assess the performance of the proposed framework, a 10-fold and 5-fold cross-validation is applied on the UrbanSound8K and ESC-50 datasets, respectively. As to the classifier, both a KNN with the parameter k set equal to 11 and a RF with 500 trees are used. The settings of classifiers are the result of a grid search over a finite set of hyperparameters. Results of tests on UrbanSound8K [9] and ESC-50 [10] are reported in terms of average accuracy in Table 3.1 and Table 3.2, respectively, for the proposed algorithm considering two different values for the moments order, i.e., 10 and 20. For comparison purposes, the results obtained applying other feature sets on both UrbanSound8K and ESC-50 classification are also reported, such as MFCC and Chebyshev moments of order 10 and 20, separately.

On the UrbanSound8k dataset, we obtain the best performance using Chebyshev moments of order 10 in conjunction with MFCC. Specifically, the KNN classifier achieves an accuracy of 40.40%, whereas for the RF, it reaches 68.55%. On ESC-50, which has a smaller number of samples with lots of classes, the effectiveness of the proposed framework can also be appreciated in terms of its discriminative capabilities. The proposed

Table 3.2: Mean classification accuracy (%) for each feature set on the ESC-50 dataset using the 5-fold cross-validation and two different classifiers, namely KNN and RF.

ESC-50 [33]

	KNN	RF
Baseline [33]	32.20	44.30
MFCC	18.15	31.60
pseudo-Zernike order 20 [41]	17.85	40.50
Chebychev order 10	13.45	45.05
Chebychev order 20	13.80	42.45
Chebychev order 10 + MFCC (ours)	16.85	**52.15**
Chebychev order 20 + MFCC (ours)	15.00	50.30

method allows to reach an accuracy of 16.85% with the KNN and 52.15% with the RF.

It is worth noting that the RF classifier consistently outperforms KNN across all feature sets, suggesting that it is better suited for this task or dataset. In addition, combining Chebyshev moments and MFCC yielded the best performance in both datasets, demonstrating the effectiveness of the proposed feature set. Chebyshev features, particularly when combined with MFCC, significantly enhance performance, highlighting their usefulness in this context.

3.2.4 Summary

In this Section, an architecture based on a machine learning approach is devised and analyzed to automatically discriminate different audio signal sources. The proposed framework is based on a concatenation of two different feature vectors. The former is obtained as the Chebyshev moments extracted from the CFD that is, in turn, obtained from the Mel-spectrogram of the incoming audio, and the second comprises the well-known MFCC. Hence, the proposed procedure has a low computational complexity thanks to the symmetry property of the discrete Chebychev moments as well as the fast computation of the CFD with the FFT algorithm. Tests conducted on UrbanSound8K and ESC-50 datasets have shown interesting results demonstrating the effectiveness of the proposed pipeline.

Although the performance demonstrates the effectiveness of using Chebyshev polynomials in constrained, there is still a significant gap between the machine learning-based approach and state-of-the-art in terms of mean classification accuracy, e.g., PANNs [18]. Hence, the next Section will focus on the design of a deep learning-based method employing only Chebyshev moments with improved performance and contained resource requirements.

3.3 Lightweight Convolutional Neural Network using Chebyshev Moments

Even though machine-learning models provide complete control of the extracted features and a limited computational complexity, each specific procedure requires strong theoretical expertise on the application field and difficulties in generalizing the developed methods. Conversely, deep learning has the advantage of being more generally applicable with also higher classification performance. These results are, however, often paid in terms of a higher computational burden and the need for the availability of many data to train the network.

The main features of these two competing strategies are exploited to take advantage of the strengths of both and limit their weaknesses. To this aim, a FCN is designed, which encompasses two convolutional branches for extracting features from the CFD representation and the Chebyshev moments coefficients, respectively. Hence, the proposed pipeline's first branch consists of a few convolutional layers alternating with max-pooling layers that apply a proper transformation of the input CFD.

The main advantage of the proposed framework lies in its limited computational complexity. The processing is performed on the reduced size CFD (efficiently obtained using the FFT) rather than the wider Mel-spectrogram. Additionally, the Chebyshev polynomials can be a priori computed and stored since they only depend on the polynomial order and the CFD size. Beyond the above attentions, the developed 2D FCN is characterized by a few layers with also a very low number of parameters if compared with SOTA architectures.

Performances have been assessed in terms of the average accuracy of performing cross-validation on two widely investigated datasets, namely UrbanSound8K and ESC-50. Results show the effectiveness of the proposed approach in comparison with other existing state-of-the-art approaches with higher computational complexity.

3.3.1 Proposed Approach

Let $g(\cdot)_{\boldsymbol{\theta}} : \mathbb{R}^{N_{\text{CFD}} \times N_{\text{CFD}}} \rightarrow \mathbb{R}^{N_{\text{classes}}}$ be the FCN parametrized with weights $\boldsymbol{\theta}$ that processes the CFD representation obtained from Equation (3.8) and

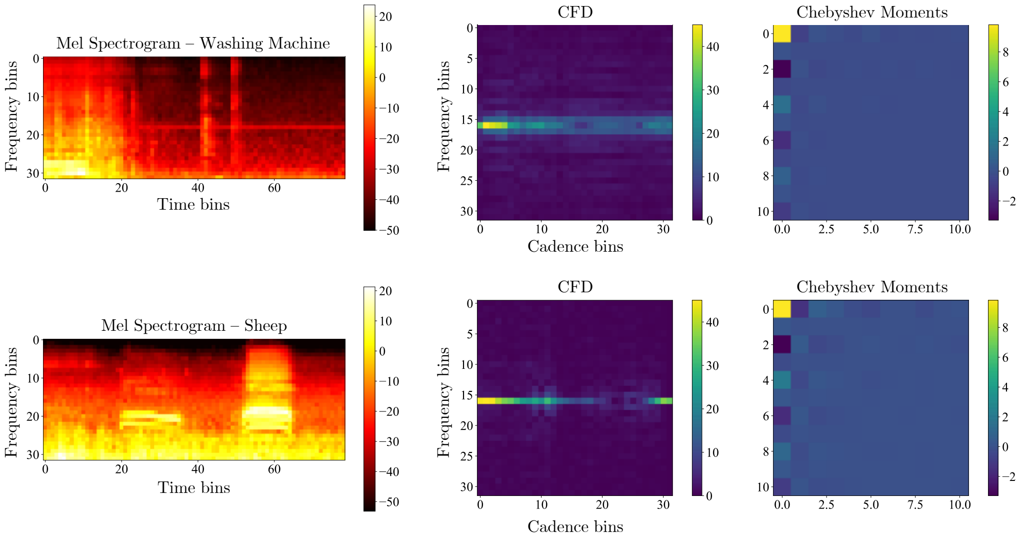


Figure 3.3: Examples of Mel-spectrogram, CFD with $N_{\text{CVD}} = 32$, and Chebyshev moments with $l = 10$ from two audios of the ESC-50 dataset.

predicts the vector of class probabilities, also denoted as class logits, $\hat{\mathbf{y}} \in \mathbb{R}^{N_{\text{classes}}}$. Specifically, $g(\cdot)_{\boldsymbol{\theta}}$ encompasses two neural branches ($g_{\text{CFD}_{\boldsymbol{\theta}}}$ and $g_{\text{Cheb}_{\boldsymbol{\theta}}}$) that are responsible for processing the CFD and the Chebyshev moments, respectively.

In more detail, the branch $g_{\text{CFD}_{\boldsymbol{\theta}}}(\cdot) : \mathbb{R}^{N_{\text{CFD}} \times N_{\text{CFD}}} \to \mathbb{R}^{N_{\text{classes}}}$ consists of three convolutional blocks, labeled as ConvBlock(C_i), where C_i represents the number of output channels. These blocks are used for extracting spatial features from the 2D representation. Each block performs a 2D convolution with 3 × 3 kernels, followed by batch normalization [16] and the activation function called ELU [17], which is defined as

$$\text{ELU}(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases} \tag{3.14}$$

where α is set to 1 to avoid negative values saturation. After the first two blocks, an image downsampling process is performed using the MaxPool$(2, 2)$ function. Then, a linear projection layer is employed to transform the output of the final convolutional block into a feature tensor with several channels equal to the number of classes N_{classes}. The idea is to have a feature map for each class. To this aim, a 1×1 convolutional layer is utilized. Finally, class logits $\hat{\mathbf{y}}_{\text{CFD}} \in \mathbb{R}^{N_{\text{classes}}}$ are obtained by means of the Global

Average Pooling (GAP) operator.

In parallel, the branch $g_{\text{Cheb}_{\boldsymbol{\theta}}}(\cdot) : \mathbb{R}^{N_{\text{CFD}} \times N_{\text{CFD}}} \rightarrow \mathbb{R}^{N_{\text{classes}}}$ extracts the Chebyshev moments from the normalized CFD and provides the class logits $\hat{\mathbf{y}}_{\text{Cheb}} \in \mathbb{R}^{N_{\text{classes}}}$. First, Chebyshev moments are extracted from the CFD following the procedure detailed in Section 3.2.1. Then, the coefficients are arranged in a squared matrix of size $(l+1) \times (l+1)$ to be processed by the network $g_{\text{Cheb}_{\boldsymbol{\theta}}}(\cdot)$.

Regarding the architecture of the Chebyshev branch, it is composed of 2 consecutive ConvBlock with 32 and 64 filters with size 3×3. Similarly to $g_{\text{CFD}_{\boldsymbol{\theta}}}(\cdot)$, a linear projection layer reduces the number of feature maps to the number of classes, and the GAP layer maps to class logits.

Table 3.3: Description of the proposed 2D FCN.

Input: normalized CFD $\overline{\Delta} \in \mathbb{R}^{N_{\text{CFD}} \times N_{\text{CFD}}}$

CVD branch $g_{\text{CFD}_{\boldsymbol{\theta}}}(\cdot)$	**Moments branch** $g_{\text{Cheb}_{\boldsymbol{\theta}}}(\cdot)$
ConvBlock(128)	Chebychev moments extraction of order N_{Cheb}
MaxPool(2, 2)	ConvBlock(16)
ConvBlock(128)	ConvBlock(64)
MaxPool(2, 2)	Projection to N_{classes} channels
ConvBlock(128)	GAP$(\cdot)$
Projection to N_{classes} channels	-
GAP$(\cdot)$	-
Output: class logits $\hat{\mathbf{y}}_{\text{CFD}}$	**Output**: class logits $\hat{\mathbf{y}}_{\text{Cheb}}$

Finally, the prediction of the approach $\hat{\mathbf{y}} \in \mathbb{R}^{N_{\text{classes}}}$ is computed by element-wise multiplication, denoted as $\otimes$, between the two branch estimations as a soft-voting strategy

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_{\text{CFD}} \otimes \hat{\mathbf{y}}_{\text{Cheb}}. \tag{3.15}$$

This training procedure has been applied to provide coherence between the two proposed audio representations. The architecture is trained by means of the Cross-Entropy loss $\mathcal{L}_{\text{CE}}$ between the predicted and the ground truth labels.

Table 5.1 provides a comprehensive overview of the architecture of the FCN. The whole neural network configuration has been tuned by exploiting a hyperparameter grid search optimization.

3.3.2 Experiments

To assess the performance of the proposed framework, a 10-fold and 5-fold cross-validation is applied to the UrbanSound8K [32] and ESC-50 [33] datasets, respectively. The performance on these datasets is assessed using the accuracy metric, evaluating the number of perfect matches between predicted and ground truth labels.

Table 3.4: Mean classification accuracy with 95% confidence interval on the UrbanSound8K dataset using the 10-fold cross-validation.

UrbanSound8K [32]

	Accuracy
Baseline	0.66
CFD Mel16 + order 10	0.72 ± 0.05
CFD Mel16 + order 15	0.73 ± 0.07
CFD Mel16 + order 20	0.67 ± 0.11
CFD Mel32 + order 10	0.72 ± 0.06
CFD Mel32 + order 15	$\mathbf{0.73 \pm 0.05}$
CFD Mel32 + order 20	0.67 ± 0.09
CFD Mel64 + order 10	0.73 ± 0.06
CFD Mel64 + order 15	0.72 ± 0.06
CFD Mel64 + order 20	0.71 ± 0.06

The performance of the proposed approach on UrbanSound8K and ESC50 is depicted in Table 3.4 and Table 3.5, respectively. The best results, which outperform the baselines, are obtained when the CFD is computed from the spectrogram with 32 mel bins. Moreover, it is notable that the best order of Chebyshev moments depends on the scenario and the amount of available data. In fact, with a smaller dataset, i.e., ESC50, the best performance is observed with a greater number of Chebyshev coefficients than in the case of a larger dataset, i.e., UrbanSound8k.

Moreover, it is worth highlighting that the experiments have been carried out only with the Mel spectrogram. Even though the approach is agnostic concerning the audio representation, tests conducted on other time-frequency analyses, such as STFT and MFCC, yielded non-converging training procedures.

Table 3.6 depicts the performance and the computational complexity (for the number of learnable parameters) of well-known deep learning strategies, e.g., CNNs [18,42] and Transformers [23], for audio classification without the use of additional training data such as AudioSet [12]. It is notable how the proposed approach is three orders of magnitude lower than well-known state-of-the-art models for sound recognition. Since the Chebyshev polynomials only depend on the polynomial order as well as on N_{CVD}, they

Table 3.5: Mean classification accuracy with 95% confidence interval on the ESC50 dataset using the 5-fold cross-validation.

ESC50 [33]

	Accuracy
Baseline	44.30
CFD Mel16 + order 10	0.59 ± 0.05
CFD Mel16 + order 15	0.58 ± 0.02
CFD Mel16 + order 20	0.60 ± 0.03
CFD Mel32 + order 10	0.57 ± 0.04
CFD Mel32 + order 15	0.59 ± 0.06
CFD Mel32 + order 20	$\mathbf{0.62 \pm 0.05}$
CFD Mel64 + order 10	0.58 ± 0.04
CFD Mel64 + order 15	0.60 ± 0.04
CFD Mel64 + order 20	0.57 ± 0.02

can be a priori computed. This is compliant with real-time applications of the proposed pipeline.

Moreover, regarding the computational complexity of deep neural networks, as explained in [43], having a smaller number of learnable parameters can help mitigate the risk of overfitting, especially when dealing with small datasets. Overfitting occurs when the model becomes too complex and starts to memorize noise or outliers in the training data. A simpler model with fewer parameters is less prone to overfitting. Additionally, with fewer parameters to update, the optimization process requires less computational resources and time. This can be advantageous when working with limited computational capabilities or large datasets [43].

In addition, thanks to the CFD computation, the size of the input feature is lower than canonical time-frequency representations such as STFT and Mel-spectrogram. In fact, the configuration of the proposed approach that yields the best results encompasses a CFD of size 32×32. Instead, without this domain shift, a canonical time-frequency analysis that computes the same preprocessing yields a 32×64 spectrogram, increasing the overall forward step of neural networks.

However, a drawback of this approach is the loss of the time information. Performing a DFT for each frequency bin in the Mel-spectrogram produces a new frequency-cadence domain, in which the cadence provides information about the amount of repetition of each frequency within the observed signal for all the observation time. Therefore, the original time information is integrated and hence is somehow lost when the second DFT is applied.

Table 3.6: Study on the computational complexity and performance of the proposed approach with CFD on 32-bins Mel-spectrogram in comparison with state-of-the-art architectures. We denote with ↑ when the performance is better when the metric is high and ↓ otherwise. A dash symbol means no experiments have been provided by the authors.

Model	Params (M) ↓	Acc ESC50 ↑	Acc USK8 ↑
CNN-based			
CNN14-PANN [18]	81.06	0.83	**0.79**
AemNet [42]	14.40	0.77	0.77
Transformer-based			
AST [23]	88.10	**0.87**	-
Proposed FCN	**0.32**	0.62	0.73

3.3.3 Summary

In this Section, a new architecture that employs the CFD and the Chebyshev moments for the classification of environmental sound is presented. Specifically, a low-complexity learning-based approach is designed for extracting features and classifying audio from a novel feature set. However, as mentioned in the discussion, the employed representation loses time information, making the architecture not suitable for tasks where time-wise classification is required, such as SED [44]. A possible improvement is to compute the CFD and Chebyshev pipelines on sliding windows of the starting time-frequency representation. By doing so, it is possible to evaluate the features in a time-aware fashion. In conjunction, the employment of more advanced attention-based architecture, such as ViT [22, 45], and large-scale audio datasets, such as AudioSet [12], could improve the effectiveness of the proposed approach while always keeping an eye on the computational cost.

3.4 Tackling the Domain Shift

Domain shift is a critical problem in ASC where models trained on one set of audio conditions underperform when tested on different acoustic environments, for example, trained on advanced recording systems and tested on Commercial-off-the-shelf (COTS) devices [46] and vice versa. An example of the domain shift problem is presented in Fig. 3.4, where an ASC system is trained and tested using different natures of audio recordings from differ-

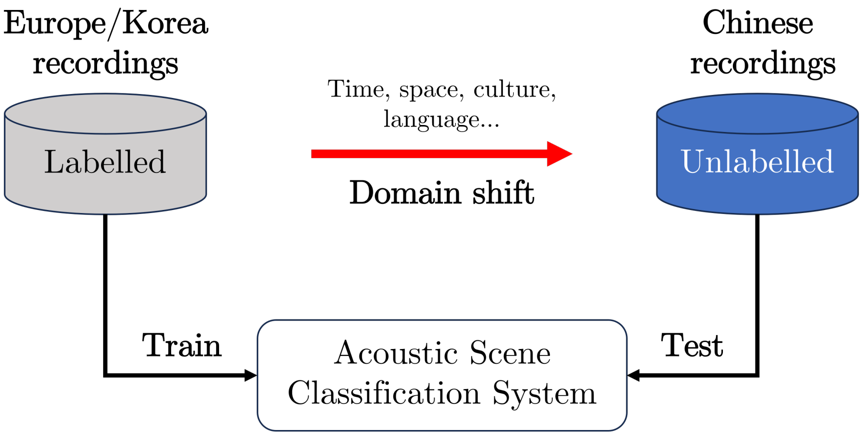


Figure 3.4: Example of the domain shift problem in the context of the IEEE ICME 2024 Grand Challenge.

ent factors, e.g., time, space, and culture. In [46], the authors proposed an unsupervised domain adaptation method that aligns the first- and second-order statistics of all the frequency bands of target-domain acoustic scenes to the ones of the source-domain training dataset. However, there is a lack of methods that exploit large portions of unlabeled raw data to improve the supervised training of deep learning models. A recent study introduced a multi-target domain adaptation technique focusing on reducing the domain gap by treating domain shift as a measurable distance [47].

To tackle the domain shift issue, a deep learning approach is proposed based on an attention module and a learned time-frequency representation, namely Wavegram. Then, a multi-iteration FT process is devised to train the model on the source domain to improve its generalization ability. Finally, unlabeled data is used in a semi-supervised fashion to refine the model's predictions.

3.4.1 Proposed Approach

Initially, a pre-processing stage is used for extracting the complex STFT $\mathrm{STFT}\{\mathbf{x}\}$ from the single-channel audio signal $x[n]$. This transform is performed using a Hann window of duration 32 ms with 50% overlap. Next, a log-Mel spectrogram $X_{\mathrm{Mel}} \in \mathbb{R}^{t \times f}$ is extracted by using a Mel filterbank $\mathrm{H}_{\mathrm{Mel}}\{\cdot\}$ as follows:

$$X_{\mathrm{Mel}} = 20 \log_{10} \mathrm{H}_{\mathrm{Mel}}\{\mathrm{STFT}\{\mathbf{x}\}\}, \tag{3.16}$$

where t and f denote the number of time and frequency bins, respectively.

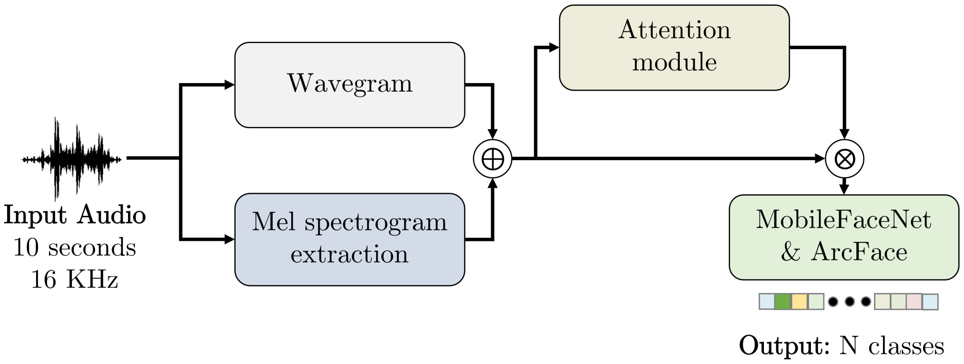


Figure 3.5: Proposed approach for semi-supervised classification of ASC.

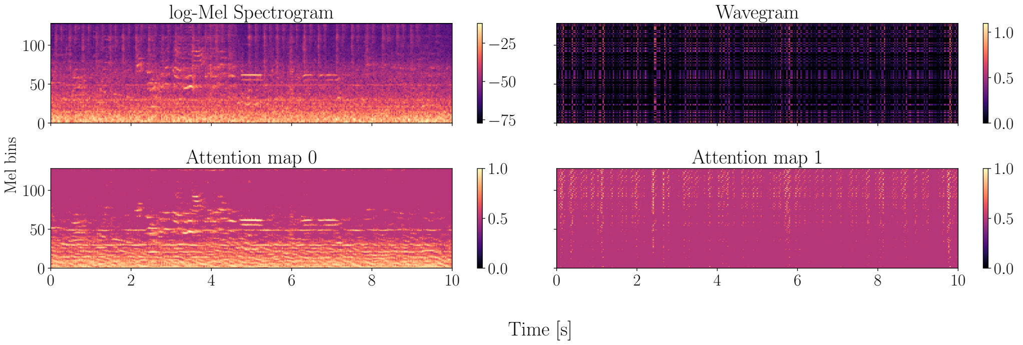


Figure 3.6: Example of acoustic features X and corresponding attention maps $H = f_{\text{ATT}}(X)$ for a *Construction site* audio recording. The first row depicts the log-Mel spectrogram and the Wavegram, respectively. The second row shows the attention maps that are element-wise multiplied with the acoustic features to obtain $\tilde{X}$.

In [18], Wavegram is introduced as a new learned time-frequency representation for audio tagging. In particular, it is designed to capture time-frequency patterns that are generally lost during the extraction of handcrafted filterbanks, e.g., Mel spectrograms [18]. Several methods have been based on Wavegram by applying a 1D convolution that acts as a learnable STFT [48–50]. Next, the features were further processed by layer normalization and 1D convolutions with 3×3 kernels [49,50]. To reduce the computational complexity, Wavegram consists only of a separable 1D convolutional layer with $f = 128$ filters with 1024 neurons each. To mimic the windows' overlap in the STFT computation, stride and padding of 512 samples are applied. The output of the Wavegram is denoted as $X_{\text{Wave}} \in \mathbb{R}^{t\times f}$.

Finally, the log-Mel spectrogram and the output of Wavegram are concatenated along the channel dimension:

$$X = [X_{\text{Mel}}, X_{\text{Wave}}] \in \mathbb{R}^{t\times f\times 2}. \tag{3.17}$$

The attention module is devised to construct an attention map $H \in \mathbb{R}^{+t\times f\times 2}$ utilizing both the log-Mel spectrogram and the Wavegram, similar in [51]. Its objective is to highlight the most significant regions of features for the task of classification. This module is represented as: $\mathrm{f}_{\text{ATT}} : \mathbb{R}^{t\times f\times 2} \rightarrow \mathbb{R}^{+t\times f\times 2}$. It encompasses two separable convolutional blocks with 16 and 64 filters of size 3×3, sequentially. Following these blocks, a convolutional layer of 1×1, i.e., a projection layer, is applied, and a sigmoid activation function maps each pixel to a probability, thus producing a $t \times f \times 2$ attention map. The enhanced acoustic features $\tilde{X} \in \mathbb{R}^{t\times f\times 2}$ result from the element-wise product ($\otimes$) of the time-frequency representations and the attention map, defined as

$$\tilde{X} = f_{\text{ATT}}(X) \otimes X. \tag{3.18}$$

An example of log-Mel spectrogram, Wavegram, and their attention maps is depicted in Fig. 3.6. In the given example, it is notable that the attention model primarily concentrates on time-frequency patterns and the lower frequency bands within the log-Mel spectrogram. Conversely, the Wavegram provides a semantic representation of time and frequency that is specific to each acoustic scene.

The classification layer we employ is ArcFace [24], which combines the cross-entropy loss with the requirement of features to be distributed on a hypersphere with respect to their labels:

$$\mathcal{L}_{\text{AF}}(\boldsymbol{\theta}, \mathbf{y}) = -\mathbf{y}^T \frac{e^{s\cos(\boldsymbol{\theta}+m\mathbf{y})}}{\sum_{i=1}^{c} e^{s\cos(\theta_i+m\hat{y}_i)}}, \tag{3.19}$$

where the vector of angles $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_c]$ corresponds to each class and is derived by computing $\theta_i = \arccos(\mathbf{w}_i^T\mathbf{h})$, reflecting the correlation between the features classified as $\mathbf{h} \in \mathbb{R}^{h\times 1}$ and the ArcFace weights learned as $\mathbf{w}_i \in \mathbb{R}^{h\times 1}$ for the i-th class. The constants $s \in \mathbb{R}^+$ and $m \in \mathbb{R}^+$ denote the scaling and margin coefficients for the ArcFace loss, respectively.

Initially, the model is pre-trained on the TAU Urban Acoustic Scenes 2020 Mobile development dataset [11], similarly to the baseline provided by the organizers [52]. This stage involves adjusting the model's weights to recognize sound patterns and features of the urban scenario.

Then, multiple FT iterations are performed on the labeled *development* dataset of the Grand Challenge. In this work, a FT iteration consists of

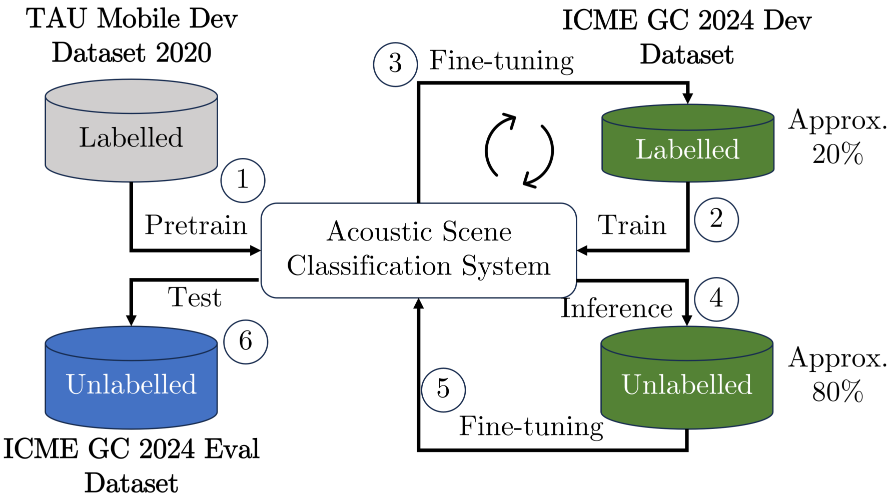


Figure 3.7: Dataset and proposed semi-supervised pipeline.

removing the last classification layer, i.e., ArcFace [24], and keeping the model's weights. This iterative process helps the model to adapt to specific tasks, handle class imbalances, and enhance its ability to generalize to new data. It also offers insights for further model refinement, making the final model more suited to real-world applications [53].

3.4.2 Experiments

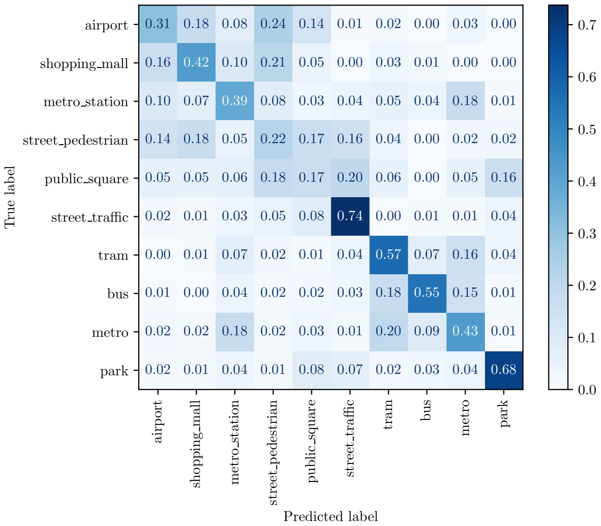
The 2023 Chinese Acoustic Scene (CAS) [52] dataset is an extensive resource foundational to studies on environmental acoustic scenes, containing 10 scenes with a collective length of more than 130 hours. Each of the dataset's 10-second sound clips is accompanied by metadata detailing its recording location and time. Derived from the CAS 2023, the dataset for the ICME 2024 challenge includes development and evaluation parts. The evaluation comprises 1,100 recordings chosen from 12 cities, incorporating 5 cities not previously included to enrich the evaluation process for domain shift scenarios. Due to the nature of the challenge, we randomly split the development dataset into training, validation, and testing sets using a percentage ratio of 80%-10%-10%, respectively.

TAU Urban Acoustic Scenes 2020 Mobile development dataset [11] is used to pre-train the proposed model. The dataset encompasses recordings from 12 European cities across 10 distinct acoustic scenes, captured with 4 different devices. Moreover, synthetic data was generated for 11 mobile devices, drawing on the original recordings. Two of the 12 cities are exclusively included in the evaluation set. The overall length of the dataset is 64 hours. Training, validation, and testing have been carried out following the official splits provided by the organizers of the challenge.

For both CAS 2023 [52] and TAU Urban Acoustic Scene (UAS) 2020 [11], we follow the authors where accuracy is employed to assess the performance of models.

In this work, the sampling frequency is set to $f_s = 16$ kHz to reduce the computational complexity of the approach. The classifier at the end of the DNN is MobileFaceNet [54]. The number of trainable parameters is 874k, highlighting the low-complexity characteristic of the approach. Regarding the training and FT procedure, the model is trained for 100 epochs with batches of size 32. A cosine annealing learning rate is employed with an initial learning rate $\eta_{\text{max}} = 0.001$ with a maximum number of steps $T_{\text{max}} = 100$. Pytorch-Lightning and Weights&Biases are utilized for training and logging, respectively. ArcFace's scale and margin coefficients are set to $s = 8$ and $m = 0.2$, respectively, following [24]. The number of FT iterations on the labeled ICME 2024 development dataset is set to 3 since no improvement in the validation loss has been observed.

Fig. 3.8 reports the confusion matrix of the proposed approach on the TAU Urban Acoustic Scenes 2020 Mobile dataset. Overall, the model achieves an average accuracy of 45.5%, which is consistent with the performance of architectures that are not ensembles of models [11], following the rules of the ICME 2024 Grand Challenge.



	airport	shopping_mall	metro_station	street_pedestrian	public_square	street_traffic	tram	bus	metro	park
airport	0.31	0.18	0.08	0.24	0.14	0.01	0.02	0.00	0.03	0.00
shopping_mall	0.16	0.42	0.10	0.21	0.05	0.00	0.03	0.01	0.00	0.00
metro_station	0.10	0.07	0.39	0.08	0.03	0.04	0.05	0.04	0.18	0.01
street_pedestrian	0.14	0.18	0.05	0.22	0.17	0.16	0.04	0.00	0.02	0.02
public_square	0.05	0.05	0.06	0.18	0.17	0.20	0.06	0.00	0.05	0.16
street_traffic	0.02	0.01	0.03	0.05	0.08	0.74	0.00	0.01	0.01	0.04
tram	0.00	0.01	0.07	0.02	0.01	0.04	0.57	0.07	0.16	0.04
bus	0.01	0.00	0.04	0.02	0.02	0.03	0.18	0.55	0.15	0.01
metro	0.02	0.02	0.18	0.02	0.03	0.01	0.20	0.09	0.43	0.01
park	0.02	0.01	0.04	0.01	0.08	0.07	0.02	0.03	0.04	0.68

Figure 3.8: Confusion matrix of the proposed approach on TAU Urban Acoustic Scenes 2020 development dataset.

Table 3.7 shows the performance of the proposed approach with several

Table 3.7: Comparison of several training setups for test accuracy. All the results are in percentages.

Approach	Acc
from scratch	63.8
1 FT iteration	97.7
2 FT iterations	98.3
3 FT iterations	99.4
3 FT iterations + unlabelled dataset	**100.0**

training setups. Training from scratch yields the worst performance with an average accuracy of 63.8%. Instead, pretraining on the TAU Urban Acoustic Scenes 2020 improves the generalization ability of the model.

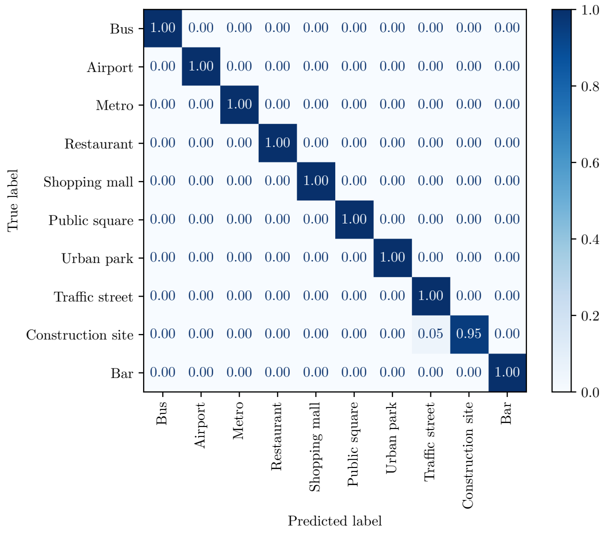


Figure 3.9: Confusion matrix of the proposed approach on ICME 2024 Grand Challenge test set dataset before exploiting the unlabeled dataset.

Moreover, the multi-iteration FT process further enhances the performance, achieving a remarkable 99.4% of accuracy on the test set, as can be inspected from the confusion matrix in Fig. 3.9. With the addition of the unlabeled dataset in the FT, the proposed approach achieves optimal classification performance, showing a diagonal confusion matrix in Fig. 3.10.

The proposed approach on the evaluation dataset achieved an accuracy of 63.1%, improving the performance of the baseline by 3.1%, as can be inspected in Table 3.8. Compared to the baseline, which employs a

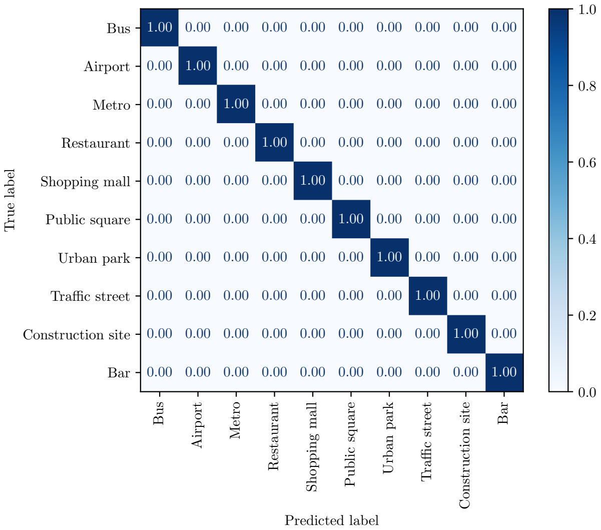


Figure 3.10: Confusion matrix of the proposed approach on ICME 2024 Grand Challenge test set dataset after exploiting the unlabeled dataset.

Table 3.8: Comparison of performance between proposed approach and baseline on the evaluation set. All the results are in percentages.

Approach	Acc	Bus	Airport	Metro	Restaurant	Shopping mall	Public square	Urban park	Traffic street	Construction site	Bar
Baseline	60.0	40.0	54.7	90.0	**69.0**	51.0	**29.0**	46.0	65.0	**68.0**	**87.0**
Our approach	**63.1**	**42.0**	**80.0**	**98.0**	60.0	**69.0**	28.0	**58.7**	**74.0**	61.0	60.0

transformer-based model, our approach utilizes a CNN, thereby offering a more lightweight solution.

3.4.3 Summary

In this Section, a semi-supervised learning approach for ASC that addresses domain shift is proposed for the IEEE ICME 2024 Grand Challenge. Thanks to an attention-based CNN, a learning-based time-frequency representation, and an iterative FT process, our model demonstrated optimal performance on the development dataset of the challenge. On the evaluation set, our methodology gained the fifth position in the competition, outperforming the baseline and providing a low-complexity trade-off. Future works will be focused on involving augmentation strategies in either the pretraining or the finetuning procedure.

3.5 Conclusion

The content of this Chapter is associated with the following publications:

- L. Pallotta, **Michael Neri**, M. Buongiorno, A. Neri, and G. Giunta, "A Machine Learning-Based Approach for Audio Signals Classification using Chebychev Moments and Mel-Coefficients", in: *International Conference on Frontiers of Signal Processing (ICFSP)*, 2022 [40].
- **Michael Neri***, L. Pallotta, and M. Carli "Low-Complexity Environmental Sound Classification using Cadence Frequency Diagram and Chebyshev Moments", in: *International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2023 [55].
- **Michael Neri*** and M. Carli, "Semi-Supervised Acoustic Scene Classification under Domain Shift using Attention-based Separable Convolutions and Angular Loss", in: *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2024 [56].
- **Michael Neri**, "Anomaly Detection and Classification of Audio Signals with Artificial Intelligence Techniques", in: *Science Talks*, 2024 [57].

Chapter 4

Unsupervised Sound Anomaly Detection

4.1 Introduction

Until now, in this Thesis, the developed models perform audio classification using the presence of labels, i.e., in a supervised fashion. However, several audio challenges are intrinsically unsupervised.

In many real-world scenarios, labeled data is hard to come by, expensive, or impossible to acquire. For instance, in industrial environments, the detection of anomalies in machine sounds and the detection of unusual patterns in environmental noise are common problems that one encounters with vast amounts of unlabeled data [58]. Supervised learning methods have exceptionally high performance regarding accuracy with large labeled datasets; however, their applicability in such contexts is greatly limited.

UASD overcomes these limitations by focusing on the identification of deviations from normal behavior but without the need for labeled training data. In place of learning from predefined categories, the model will learn the underlying distribution of normal sound patterns and flag whatever instances deviate farthest from this learned distribution as probable anomalies. This type of approach is much appreciated in different applications, like predictive maintenance, security surveillance, and health monitoring, where the detection of rare and unpredicted events is useful.

In the context of unsupervised anomaly detection, an *anomaly* refers to data patterns that deviate from the expected *normal* behavior [59]. Likewise, ASD is the task of understanding whether a sound is *normal* or not (*anomalous*) [49], as can be visually inspected in Fig. 4.1.

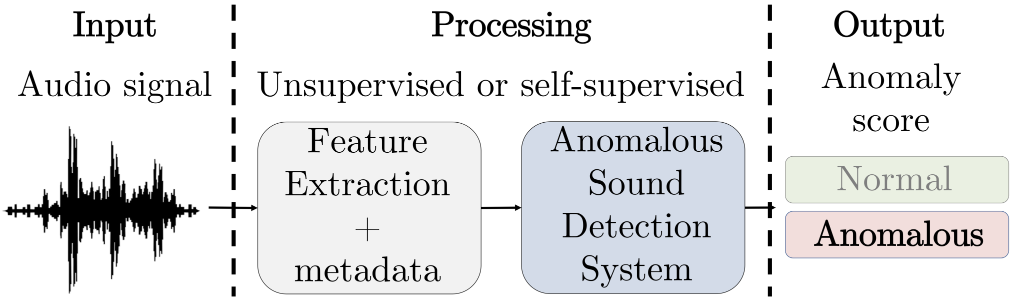

Input
Audio signal
Processing
Unsupervised or self-supervised
Feature Extraction + metadata
Anomalous Sound Detection System
Output
Anomaly score
Normal
Anomalous

Figure 4.1: Definition of the binary sound anomaly detection task.

The UASD challenge lies in correctly modeling what constitutes *normal* behavior in extremely variable and dynamic acoustic environments. The sounds could vary greatly because of operational state changes, environmental noise, and other factors [58, 60]; therefore, a stable baseline cannot be easily established for normality. Further, the notion of what is an anomaly is generally context-dependent, which serves to intensify task difficulty.

UASD is applied in the field of machine condition monitoring [58, 60], medical diagnosis [61], safety and security in urban environments [62], and multimedia forensics [63,64]. Generally, DNNs are employed for ASD due to their ability to identify subtle and unknown anomalous data patterns [65]. Unsupervised or semi-supervised models are generally adopted in ASD problems because of the limited availability of anomalous sounds. SOTA UASD approaches can be classified into two categories [66, 67]: *reconstruction-based* and *classification-based*. In the first scenario, models are based on the hypothesis that only non-anomalous samples, which have been analyzed during training, can be effectively retrieved after lossy compression, e.g., Autoencoder (AE) [68]. In [69], a DNN has been designed to interpolate masked time bins of the log-Mel spectrogram. Similarly, in [70], normalizing flows have been used for estimating the probability density of normal data. However, these models suffer from generalization problems, e.g., an anomalous sample may be correctly reconstructed by an AE [71].

Classification-based approaches, instead, compute the anomaly score exploiting probability-based distances between prediction and ground truth, e.g., cross-entropy. The classification is carried out on metadata, which can be the identification number of a specific machine that produced the sound. The design rationale is that a model cannot successfully classify the metadata associated with a sound if it is anomalous [66, 67, 72]. The use of metadata as an auxiliary loss function allows the modeling of the probability distribution of normal data, namely Inlier Modeling (IM) [73]. One such model, STgram-MFN [48], extracts temporal and spectral features to classify the IDs of machines using ArcFace [24]. Similarly, in [50], two novel angular losses, ArcMix and Noisy-Arcmix, have been designed to enhance the compactness of intra-class distribution during the classification of IDs. Differently, the authors in [74] involved contrastive learning in the pre-training to reduce distances between pairs of feature embeddings from the same machine IDs.

However, it is important to consider the computational complexity in the context of UASD. The response time of an anomaly detector may be critical to limit the damage caused by an anomalous event [68]. Hence, this work also analyzes the computational complexity of SOTA approaches in terms of the number of learnable parameters. Moreover, it is often challenging to interpret why these models flag certain audio segments as anomalies due to their black-box nature. To address this issue, for the first time in the literature, we employ an attention module [51] to provide explanations for the decisions made by the anomaly detection system. The attention mechanism highlights which parts of the input are most influential in the model's anomaly detection, thereby enhancing the interpretability of the model's outputs.

The contributions of this Chapter can be summarized as follows:

- Definition of an attention module focused on identifying time-frequency anomalous pattern detected both in the log-Mel spectrogram and from the learned representation, i.e., Wavegram [18].

- Use of separable convolutions to reduce the computational complexity of the model, decreasing by approximately 13% of the number of learnable parameters concerning the top-tier approaches of the literature;

- Statistical analysis of the attention maps highlights the importance of high-frequency bins in the log-Mel spectrogram as the main cue for the identification of anomalous sounds in this scenario. Moreover, a comparison with SOTA approaches, in terms of performance and computational complexity, is carried out.

4.2 Proposed Anomaly Detector

The goal is to determine whenever a single-channel audio signal $x[n]$ is anomalous without using in training the binary anomaly label $y \in \mathbb{Z}^2$. To

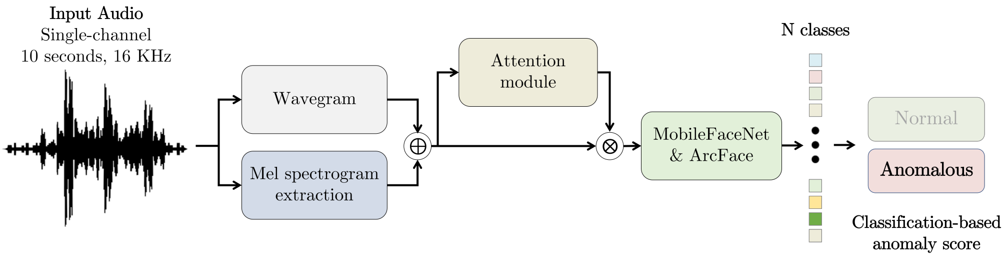


Figure 4.2: Description of the proposed pipeline for UASD.

do so, we employ time-frequency representations as features, namely log-Mel spectrogram and Wavegram, to jointly identify patterns in time and frequency since audio signals are generally non-stationary [14]. In conjunction with an attention module and angular loss, an efficient DNN is proposed for classification-based anomaly detection. The overall architecture is shown in Figure 4.2.

Initially, a pre-processing stage is employed to extract the complex STFT $\text{STFT}\{x[n]\}$ from the audio signal $x[n]$. This transform is performed using a Hann window of length 64 ms with 50% overlap. The selection of the window function is critical since windowing in the time-domain results in a convolution in the frequency domain, disrupting the spectral characteristics of the audio signal. Hann window mitigates this problem thanks to its characteristic of having the localization of spectral energy around the normalized frequency $w = 0$, minimizing spectral leakage [75]. The length and overlap of windows are consistent with those found in the literature for ASD. To enhance time-frequency patterns and as previously done in Chapter 3, a log-Mel spectrogram $X_{\text{Mel}} \in \mathbb{R}^{t \times f}$ is extracted using a Mel filterbank $\text{H}_{\text{Mel}_f}\{\cdot\}$ as $X_{\text{Mel}} = 20 \log_{10} \text{H}_{\text{Mel}_f}(\text{STFT}\{x[n]\})$, where t and f denote the number of time and frequency bins, respectively.

In [18] Wavegram is introduced as a new learned time-frequency representation for audio tagging. In particular, Wavegram is designed to capture relevant time-frequency cues for the classification that may go unnoticed, like hand-crafted log-Mel spectrograms, due to its lossy representation [18]. Within the scope of UASD, several methods have been based on Wavegram by applying a 1D convolution that acts as a learnable STFT [48]. Next, the features have been further processed by layer normalization and 1D convolutions with small kernel sizes [49, 50]. To reduce the computational complexity, in this work, Wavegram consists only of a separable 1D convolutional layer with f strided filters to mimic the windows' overlap in the STFT computation. Finally, the log-Mel spectrogram and the output of Wavegram $X_{\text{Wave}} \in \mathbb{R}^{t \times f}$ are concatenated along the channel dimension

$X = [X_{\text{Mel}}, X_{\text{Wave}}] \in \mathbb{R}^{t \times f \times 2}$. An example of input acoustic features is depicted in Figure 4.3.

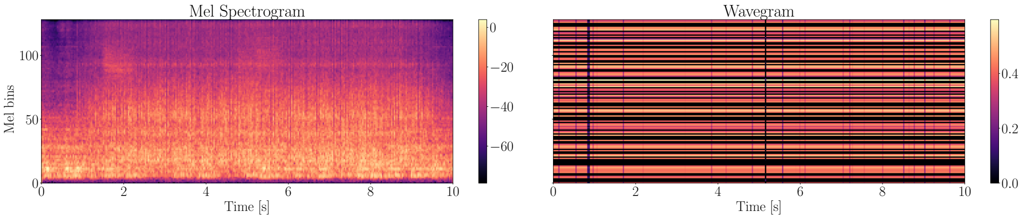


Figure 4.3: Example of acoustic features X_{Mel} and X_{Wave}, respectively, from an anomalous sound of Task 2 DCASE 2020 dataset.

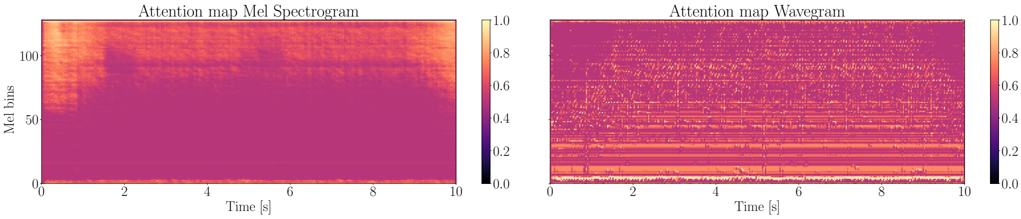


Figure 4.4: Attention maps $H = f_{\text{ATT}}(X)$ obtained from the attention module with acoustic features in Figure 4.3.

The attention module is responsible for learning an attention map $H \in \mathbb{R}^{+t \times f \times 2}$ from the log-Mel spectrogram and the Wavegram. Its objective is to emphasize regions of features that are most informative for the classification task. This module has been extensively analyzed for evaluating the distance between a microphone and a speaker [51]. However, its application in ASD has not been investigated yet. In this work, it is denoted as the function $\text{f}_{\text{ATT}} : \mathbb{R}^{t \times f \times 2} \to \mathbb{R}^{+t \times f \times 2}$. It comprises 2 separable convolutional blocks, having 16 and 64 3×3 filters, respectively. Then, a 1×1 convolutional layer, that acts as a linear projection to reduce the number of channels, with two filters, followed by a sigmoid activation function for mapping each pixel into a probability, is used to map the features to yield the $t \times f \times 2$ attention map. Finally, the weighted acoustic features $\tilde{X} \in \mathbb{R}^{t \times f \times 2}$ are obtained by element-wise multiplication ($\otimes$) between the two time-frequency representations and the attention map as $\tilde{X} = f_{\text{ATT}}(X) \otimes X$. Examples of attention maps are shown in Figure 4.4.

To improve the robustness of the model, we synthetically augment the dataset using mixup [76] in each batch during the training, defined as

$$\begin{cases} x^{ij} = \lambda x^i + (1-\lambda)x^j \\ \mathbf{y}^{ij} = \lambda \mathbf{y}^i + (1-\lambda)\mathbf{y}^j, \end{cases} \tag{4.1}$$

where $(x, \mathbf{y})$ is the tuple describing the waveform x and the one-hot encoded metadata $\mathbf{y} = [y_1, y_2, \dots, y_c]$ with c classes of a single audio recording under analysis, respectively. $i, j \in \{0, 1, \dots, n-1\}$ are randomly selected indexes of training audio samples in the batch with size n, and $\lambda \sim \text{Beta}(\alpha, \alpha)$ is the mixup coefficient. This augmentation can be performed at different levels of the deep learning architecture, e.g., input level or at intermediate feature levels [76]. In this work, the augmentation procedure is applied to input signals before the preprocessing step, following [50].

Finally, to distinguish between anomalous and normal sound, an anomaly score $\mathcal{A}_\theta$ is computed from the predicted metadata and the ground truth. As a classification-based approach, if a sound is misclassified, then it is anomalous since the model is trained to correctly classify normal sounds. We utilize ArcFace [24] as the classification layer,

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{y})_{\text{AF}} = -\mathbf{y}^T \frac{e^{s \cos \boldsymbol{\theta} + m\mathbf{y}}}{\sum_{i=1}^{c} e^{s \cos \theta_i + m\hat{y}_i}}, \tag{4.2}$$

where the angular vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_c]$ is obtained for each class by computing $\theta_i = \arccos(\mathbf{w}_i^T \mathbf{h})$, which is the result of the mapping between the features obtained from the classifier $\mathbf{h} \in \mathbb{R}^{h\times 1}$ and learned ArcFace weights $\mathbf{w}_i \in \mathbb{R}^{h\times 1}$ for the i-th class. The scalars $s \in \mathbb{R}^+$ and $m \in \mathbb{R}^+$ are the scale and margin coefficients for the ArcFace loss, respectively. As introduced in [50], the employed loss function for training the model is

$$\mathcal{L}(\boldsymbol{\theta}, \mathbf{y}, \mathbf{y}^{ij}) = \lambda \mathcal{L}(\boldsymbol{\theta}, \mathbf{y})_{\text{AF}} + (1-\lambda)\mathcal{L}(\boldsymbol{\theta}, \mathbf{y}^{ij})_{\text{AF}}. \tag{4.3}$$

During the testing phase, as the augmentation is not performed, the anomaly score is computed as $\mathcal{A}_\theta(y, \hat{y}) = \mathcal{L}(\boldsymbol{\theta}, \mathbf{y})_{\text{AF}}$.

4.3 Experiments

The Task 2 development dataset of the DCASE 2020 challenge [58] is used to assess the performance of the proposed approach. It encompasses six machines (Fan, Pump, Slider, Valve, ToyCar, and ToyConveyor), and each machine is labeled with a unique identifier to differentiate audio recordings from various machines within the same category. A total of 41 machines with 10 seconds of audio signals are collected. To assess the performance of the proposed approach, we evaluate the AUC and pAUC metrics. The latter is the AUC over a low FPR in the range $[0, p]$ with $p = 0.1$, following [77]. Our approach is trained to classify the $c = 41$ labels derived from machine types and IDs [58, 60]. For the loss and mixup computation, parameters are set as $\alpha = 0.2$, $m = 0.7$, and $s = 40$, following the guidelines provided by their corresponding works [24, 76]. Log-Mel spectrogram and Wavegram output have $t = 313$ and $f = 128$ bins. The classifier is MobileFaceNet [54], which yields a feature vector with dimensionality $h = 128$. The network is optimized using AdamW with a learning rate of 0.0001, epochs of 300, and a batch size of 64. Hyperparameters of the training procedure have been assigned using a grid search optimization procedure.

Table 4.1: Comparison with SOTA methods. **Bold** and underline are used to highlight first and second-best results, respectively.

Methods	Fan		Pump		Slider		Valve		ToyCar		ToyConveyor	
	AUC [%]	pAUC [%]	AUC [%]	pAUC [%]	AUC [%]	pAUC [%]	AUC [%]	pAUC [%]	AUC [%]	pAUC [%]	AUC [%]	pAUC [%]
IDNN [69]	67.71	52.90	73.76	61.07	86.45	67.58	84.09	64.94	78.69	69.22	71.07	59.70
MobileNetV2 [72]	80.19	74.40	82.53	76.50	95.27	85.22	88.65	87.98	87.66	85.92	69.71	56.43
Glow-Aff [70]	74.90	65.30	83.40	73.80	94.60	82.80	91.40	75.00	92.20	84.10	71.50	59.00
GMM + Arcface [66]	87.97	80.66	**95.63**	85.74	99.22	97.55	91.26	84.00	95.28	86.91	69.80	61.21
STgram-MFN [48]	94.04	88.97	91.94	81.75	**99.55**	**97.61**	99.64	98.44	94.44	87.68	74.57	63.60
SW-WaveNet [49]	97.53	91.54	87.27	82.68	98.96	94.58	99.01	97.26	95.49	90.20	81.20	68.20
Noisy-ArcMix [50]	**98.32**	**95.34**	95.44	**85.99**	99.53	97.50	99.95	99.74	96.76	90.11	77.90	67.15
Proposed approach	95.10	87.25	91.97	80.00	99.24	96.10	**99.99**	**99.96**	**96.99**	**90.30**	**84.59**	**73.55**

The performance of the proposed approach compared with those obtained with SOTA architectures are represented in Table 4.1. Overall, the proposed approach shows the best performance in three of the six equipment types (Valve, ToyCar, and ToyConveyor). In the other classes (Fan, Pump, and Slider), the performance is still competitive. Generally, Table 4.1 can be used as a reference for the selection of the approach that is most suitable to the specific use case. Regarding the computational complexity analysis, Table 4.2 highlights the number of parameters and the performance of the proposed approach compared with those of the SOTA. Our system offers a good trade-off between model complexity and performance.

Table 4.2: Number of parameters, average AUC, and average pAUC of SOTA approaches and proposed method.

Methods	Parameters	AUC [%]	pAUC [%]
IDNN [69]	**46 k**	76.96	62.57
MobileNetV2 [72]	1.1 M	84.34	77.74
Glow-Aff [70]	30 M	85.20	73.90
GMM + Arcface [66]	1 M	89.86	82.68
STgram-MFN [48]	1.1 M	92.36	86.34
SW-WaveNet [49]	27 M	93.25	87.41
Noisy-ArcMix [50]	1.1 M	**94.65**	**89.31**
Proposed approach	884 k	93.44	85.71

Table 4.3 shows the selection of parameters regarding the type of features and the dimensionality of the ArcFace layer. The use of Wavegram representation $[X_{\text{Wav}}]$ in conjunction with the log-Mel spectrogram can improve the performance of the proposed model by 1.17% in terms of AUC, albeit being ineffective using it alone. Moreover, the best performance is obtained by setting the dimensionality of the classification layer to $h = 128$.

Table 4.3: Selection of parameters of the proposed approach.

Features	h	AUC [%]	pAUC [%]
Feature study			
$[X_{\text{Mel}}]$	128	92.26	84.55
$[X_{\text{Wav}}]$	128	63.48	54.12
Dimensionality study			
$[X_{\text{Mel}}, X_{\text{Wav}}]$	256	90.87	83.94
$[X_{\text{Mel}}, X_{\text{Wav}}]$	64	91.94	85.00
$[X_{\text{Mel}}, X_{\text{Wav}}]$	128	**93.43**	**85.71**

To better explain which parts of the log-Mel spectrogram are relevant for the ID classification, Figure 4.5 shows the average and standard deviation maps on the testing set of the proposed heatmap.

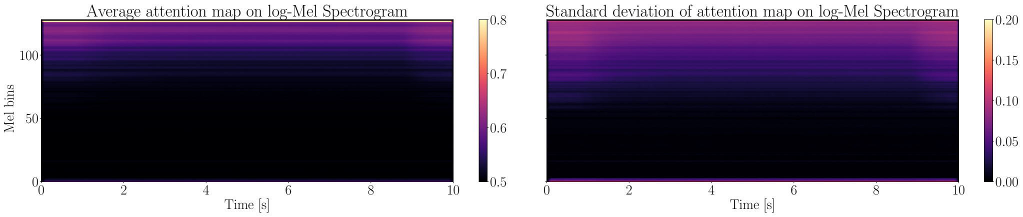


Figure 4.5: Mean and standard deviation of the attention map of the log-Mel spectrogram on testing set.

It is worth highlighting that the most important frequency bins for the identification of anomalies, i.e., ID misclassification, are contained in the range $[1.7, 8]$ kHz of the log-Mel spectrogram. In addition, the range $[0, 71]$ Hz is also relevant. The rest of the log-Mel spectrogram $[0.071, 1.7]$ kHz is assigned a value of 0.5 with zero variance, denoting this region as less important for the ID classification and, thus, less reliable for the identification of anomalies. To assess the impact of both separable convolutions and the attention module, an ablation study has been carried out. Table 4.4 demonstrates the effectiveness of using the attention map in combination with separable convolutions.

Table 4.4: Ablation study.

Methods	Parameters	AUC [%]	pAUC [%]
No both	1 M	90.50	83.62
No separable	1 M	92.25	84.82
No f_{ATT}	882 k	91.72	84.52
Proposed approach	884 k	**93.43**	**85.71**

4.4 Summary and Conclusion

In this work, a learning-based low-complexity approach is proposed to detect anomalous sound in a machine monitoring scenario. To this aim, a DNN is proposed. It exploits an attention module to highlight the most salient time-frequency patterns for identifying machine IDs. Then, an anomaly score is computed from the classification errors between predicted and ground truth metadata. Experimental results demonstrate the validity of the proposed low-complexity model. Future work will focus on the improvement of the attention module, coping with more complex tasks in the realm of sound anomaly detection such as few and one-shot unsupervised anomaly detection.

The content of this Chapter is associated with the following publication:

- **Michael Neri**, and M. Carli,“Low-complexity Unsupervised Audio Anomaly Detection exploiting Separable Convolutions and Angular Loss”, in: *IEEE Sensors Letters*, 2024.

Chapter 5

Low-Complexity Sound Event Detection in Noisy Environments

5.1 Introduction

The objective of a sound event detector is to recognize anomalies in an audio clip and return their onset and offset. In more detail, differently from monophonic audio classification, the objective of a SED system is to identify both the type of event and the exact time of its onset and offset [78]. However, recent state-of-the-art models trained on AudioSet [79] have shown to be unsuitable for human security and safety-oriented applications [78]. Generally, SED systems trained on AudioSet provide accurate start and end time identification of acoustic events, while their Recall may be unsatisfactory. In addition, detecting sound events in noisy environments is a challenging task. This is because, in a real audio signal, several sound sources co-exist, together with uncontrolled environmental and thermal noise. Moreover, the lack of large annotated audio datasets has a significant impact on the performance of SED models, leading to weak generalization capabilities of deep learning-based systems.

To address these problems, in this Chapter, a sound anomaly detection system is proposed, which is based on a FCN that exploits image spatial filtering and an ASPP [80] module called AuSPP. In more detail, our model aims to detect audio events that potentially denote the presence of circumstances threatening public safety and security (e.g., broken glass, gunshots, or shouting). The proposed model has been tested in a public transportation vehicle. The motivation for this choice is twofold. First, modern buses are equipped with on-board sensors able to collect heterogeneous data that can be employed for maintenance, i.e., the Automatic Vehicle Monitoring (AVM) paradigm [81]. Second, the same raw audio information can be exploited for granting passenger security and safety in a noisy urban environment.

To cope with the lack of datasets specifically designed for sound event detection for our scope, an annotated audio SED dataset, SEDDOB, specifically designed for the bus environment, has been devised. In more detail, labelled audios with the onset and offset time of different types of audio events are provided. To the best of our knowledge, this is the first contribution of an audio dataset for human safety in the public transport system.

The performances of the proposed system have been evaluated through segment-based metrics such as error rate, recall, and F1-Score. Moreover, robustness and precision have been evaluated through four different tests. The analysis of the results shows that the proposed sound event detector outperforms both state-of-the-art methods and general-purpose deep learning solutions. In addition, in the proposed approach, the number of learnable parameters depends on the number of class anomalies and on the time resolution. Hence, with respect to state-of-the-art SED approaches, it is possible to customize the behaviour of AuSPP, drastically reducing its complexity and detecting more sound events than state-of-the-art models.

To summarize, the contributions of this Chapter are as follows:

- the definition of a new augmented spectrogram that exploits spatial filters to enhance time-frequency patterns by means of the ASPP module;
- the design of an end-to-end SED that is more lightweight than state-of-the-art approaches. Moreover, the number of parameters depends on the time resolution and on the number of classes, thus being customizable;
- the generation of a new SED dataset for the bus environment. Synthesis of real background recording with anomalous events from state-of-the-art monophonic datasets has been performed.

5.2 Proposed Approach

One of the peculiarities of the proposed model, AuSPP, is the exploitation of a larger dimensional input than state-of-the-art Mel spectrogram-based methods. This choice allows the extraction of a larger number of semantic audio features. More specifically, the information provided by the concatenation of spatial derivatives of the Mel spectrogram helps the model to learn frequency patterns for jointly classifying the audio events and their corresponding onset and offset times.

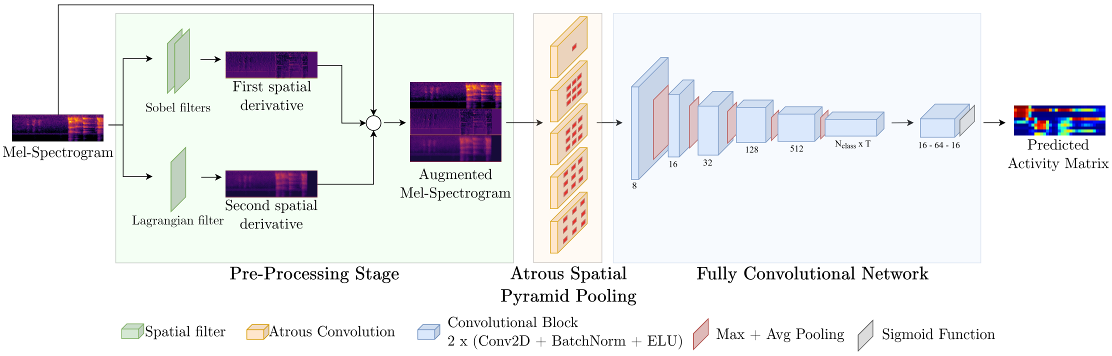


Figure 5.1: Structure of AuSPP, the proposed model for SED.

AuSPP can be partitioned into three main blocks: a pre-processing stage, the ASPP, and a FCN. The first stage is responsible for extracting a time-frequency audio representation, applying spatial filters to the input audio spectrum, and arranging the output into an augmented Mel-spectrogram. Subsequently, the ASPP module is introduced to combine the output of dilated convolutional filters. Finally, a FCN processes the ASPP output to obtain the predicted activity heatmap. The overall architecture is depicted in Figure 5.1.

The Mel spectrogram X is computed by means of the Mel-Filterbank $H_{\mathrm{Mel}_f}(\cdot)$ on the squared magnitude of the STFT of the input single-channel recording $x[n]$

$$X[m,k] = H_{\mathrm{Mel}_f}\{|\mathrm{STFT}\{x[n]\}|^2\}. \tag{5.1}$$

In more detail, the Mel filterbank groups the spectral values of each time frame into f logarithmic bins to model human sound perception. Hence, the Mel spectrogram X has size $M \times f$.

To enhance the audio patterns on the Mel-Spectrogram, Sobel [82] and Langrangian [83] operators have been employed. Let H_u and H_v be the horizontal and vertical first-order spatial derivatives, respectively. Furthermore, let H_l be the second order spatial derivative

$$H_u = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad H_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{5.2}$$

$$H_l = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{5.3}$$

The first Mel spectrogram spatial derivative is calculated by convolution:

$$X'_u = X * H_u, \tag{5.4}$$

$$X'_v = X * H_v, \tag{5.5}$$

$$X' = \sqrt{X'^2_u + X'^2_v}. \tag{5.6}$$

Similarly, the second spatial derivative is obtained as:

$$X'' = X * H_l. \tag{5.7}$$

Finally, the Mel spectrogram is concatenated on the frequency axis with its derivatives in an augmented Mel spectrogram $\hat{X}$ with size $M \times 3f$:

$$\hat{X} = X \circ X' \circ X'', \tag{5.8}$$

where $\circ$ denotes the concatenation function. An example of an augmented spectrogram $\hat{H}$ is depicted in Figure 5.2.

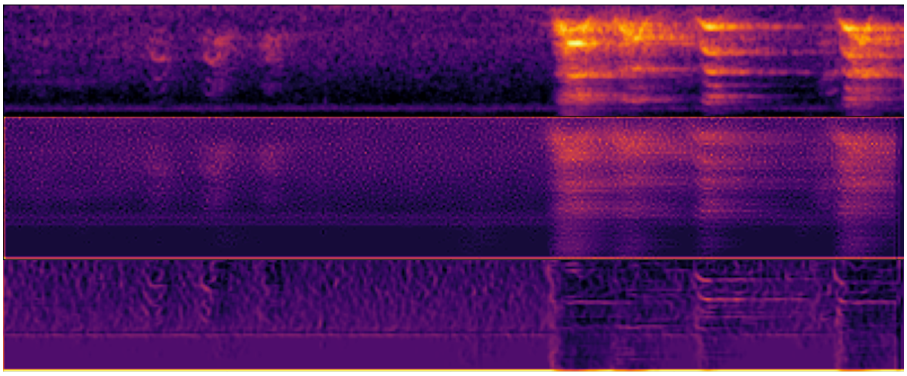


Figure 5.2: Augmented spectrogram $\hat{X}$ of size $M \times 3f$. The x-axis corresponds to the time domain, whereas the y-axis represents the frequency domain. It is worth noticing that the output of the two spatial filters emphasises the edges of the Mel spectrogram, i.e., the most intense frequency bins in terms of magnitude.

In this model, the ASPP module [80] is employed. In more detail, it applies atrous convolutions to the augmented Mel spectrogram to extract more relevant spatial features. Considering two-dimensional signals, given the augmented Mel spectrogram $\hat{X}$, a convolutional kernel W, and a region of interest I, the output of the atrous convolution for the selected region, $Y[i]$, is:

$$Y[i] = \sum_{k} \hat{X}[i + (r \cdot k)]W[k], \tag{5.9}$$

where the atrous rate r is the stride parameter of the convolutional layer of the network, which allows to application of convolutions to the input spectrogram $\hat{X}$ with upsampled zero-padded filters. The variable k accounts for all the possible regions of the image. In the proposed model, the ASPP module is composed of 5 atrous convolutions with kernels of size 3×3 with an atrous rate of $1, 2, 4, 8$, and 16, respectively. This design choice has been considered for capturing new time-frequency patterns across the augmented spectrogram.

Table 5.1: Description of the proposed FCN of AuSPP.

Input: $\hat{X}$ Augmented Mel-Spectrogram $T \times 3f$
ConvBlock(8) Max + Average Pooling 4×2
ConvBlock(16) Max + Average Pooling 4×2
ConvBlock(32) Max + Average Pooling 2×2
ConvBlock(128) Max + Average Pooling 2×2
ConvBlock(512) Max + Average Pooling 2×2
ConvBlock($T \times N_{class}$) Global Max + Average Pooling
Reshape to 1 channel $T \times N_{class}$
ConvBlock(16)
ConvBlock(64)
ConvBlock(16)
ConvBlock(1)
Sigmoid activation function
Output: $\hat{Y}$ Binary Activity Matrix $T \times N_{class}$

Inspired by the network architecture proposed in [18], let ConvBlock(C) be the generic convolutional block with C output channels, shown in Figure 5.3. It is composed of two consecutive Conv2D-BatchNormalization-ELU [84,85] blocks with 3 × 3 kernels. Variable pooling sizes are applied to intermediate ConvBlock(·) outputs to reduce the size of the feature maps. Moreover, a Dropout [86] layer is applied at the end of each ConvBlock(·) to reduce overfitting.

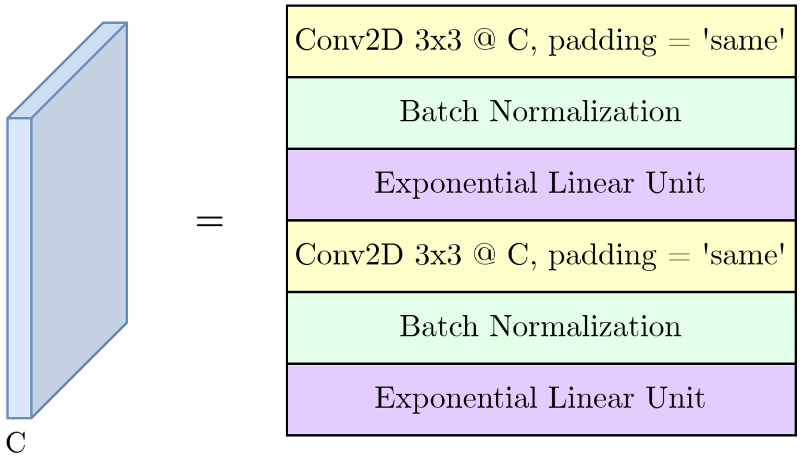


Figure 5.3: General convolutional block with C output channels.

5.3 SEDDOB: Sound Event Detection Dataset On the Bus

A synthetic audio dataset SEDDOB has been designed. As previously mentioned, a large number of high-quality annotated audios is required for training data-driven approaches. In the following, we describe the recording setup and the synthesis of the sound classes.

To collect a typical bus background, a microphone array and a recording unit have been deployed inside a bus. The microphone array has been positioned in 3 different locations (Figure 5.4). This choice accounts for the observation that in the bus used for the recordings, the engine is located in the rear. Therefore, the background sound pressure level and its spectral characteristics change with distance from the engine, which is the main source of background noise. The total length of the recorded background noise is 1 hour. All the recordings have been acquired in dedicated runs in compliance with General Data Protection Regulation (GDPR).

The audio events selected for the classification task are extracted from existing monophonic datasets: UrbanSound8k [9], ESC50 [10], and Au-

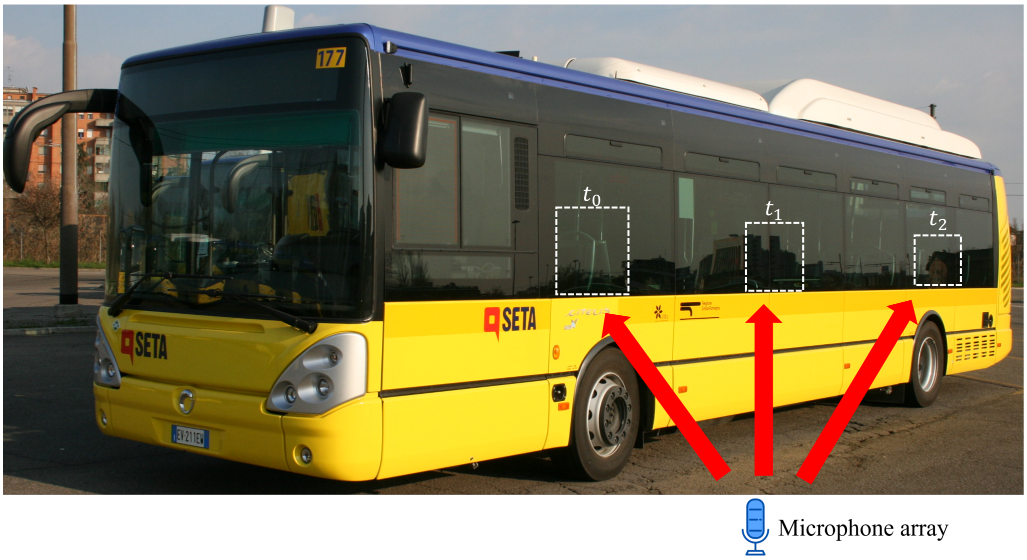


Figure 5.4: Picture of the background recording setup where (t_0, t_1, t_2) are the microphone positions.

dioSet [79]. Silences and clipping audio recordings have been removed in order not to bias the training of the models. In total, 10 audio classes, which can occur in a bus environment and that can relate to threatening events for public safety and security, have been selected: *breaking_glass, car_horn, gunshot, siren, slap, scream, cry, jackhammer, car_alarm, smoke_alarm*. Scaper [87] has been used to generate the proposed dataset by adopting the parameters configuration reported in Table 5.2. The audio length is selected based on the characteristics of human auditory perception. Tests on human subjects confirm that 4 seconds are sufficient to correctly classify events [9]. Furthermore, the distribution of class events, together with their number, onset, and offset times, is set as uniform to create a balanced dataset. In addition, augmentation strategies such as pitch shift, time stretch, and variable SNR have been introduced for generalization purposes.

5.4 Experiments

To properly test the performance of our proposed SED model, we first created and then adopted two different datasets, namely ”full dataset” and ”reduced dataset,” respectively, containing 10 and 4 audio classes. We sample a reduced dataset to examine the model’s performance under a simpler classification task, hence providing insight into its robustness, adaptability, and efficiency in handling a smaller set of audio classes.

Table 5.2: SEDDOB characteristics.

Audio Settings	
Number of soundscapes	10000 samples
Fixed duration	$4s$
Sampling frequency f_s	16000 Hz
Anomalies Statistics	
Minimum number of events	0
Maximum number of events	4
Distribution of the number of events	Uniform
Distribution of class anomalies	Uniform
Augmentation Statistics	
Minimum SNR	-5dB
Maximum SNR	0dB
Distribution SNR	Uniform
Minimum pitch shift	-0.5 semitones
Maximum pitch shift	0.5 semitones
Distribution pitch shift	Uniform
Minimum time stretch	0.9
Maximum time stretch	1.1
Distribution stretch	Uniform

We have also tried different detector resolutions, 20ms and 50ms, across audio classes. This was to investigate how time-granularity affects the complexity and accuracy for the detection of sound events. From these, we saw how different time resolutions would influence the model in the detection of sound events and hence gave deeper insight into the trade-offs between temporal precision and the computational demand of the model—something critical to the optimization of real-time applications.

We also ran experiments varying the threshold activity levels, which quantify the model's confidence in its detections of sound events. In particular, we defined *loose threshold* and *strict threshold* as 0.8 and 0.9, respectively. The loose threshold was set to make the model detect events more sensitively and thus probably cover more subtle occurrences at the expense of increasing false positives. On the other hand, the strict threshold was supposed to give more confidence that only the most certain detections would be taken into consideration, thereby reducing false positives but probably missing some real events. We benchmarked this at each of these different thresholds to show how detection confidence influences model accuracy and general reliability if we want an overall assessment of model behavior under the influence of operating conditions.

In these systematic variations, we make changes at an increasing level of complexity of the dataset, temporal resolution, and threshold settings to ensure a stringent test of the SED model's capabilities to arrive at meaningful conclusions about its performance for a wide range of realistic scenarios. Both datasets are split into 10 folds to use different portions of the data for training and testing. The AuSPP weights are updated utilizing the Adam optimizer with learning rate λ. The STFT, Mel filterbank, and training hyperparameters for each model are listed in Table 5.3.

Table 5.3: STFT parameters and training hyperparameters for all the models.

STFT Settings	
Minimum frequency	50 Hz
Maximum frequency	8000 Hz
Hop size H	160 samples
Window size N	512 samples
Window function $\mathbf{w}$	Hanning
N_{DFT}	512 samples
Mel filterbank Settings	
f	64 bins
Training Hyperparameters	
Learning rate λ	0.001
Batch size	16 samples
Maximum epochs	50 epochs
Learning rate scheduler	Reduce on Plateau of validation loss
Scheduler patience	4 epochs
Scheduler factor	0.1

We present the results of four tests where the ability to generalize and the robustness of the proposed framework are compared to state-of-the-art techniques [18,88–91]. Since SEDDOB is split into 10 folds, we provide the mean values for the aforementioned metrics.

As shown in Tables 5.4, 5.5, 5.6, 5.7, some metrics such as Acc, S_t, P_t are not meaningful for rare anomalous events activity since they are biased by the high value of TN. Moreover, they do not take into account:

- multiple class events must be recognized in the same time frame (*class errors*);
- the wrong predictions of onset and offset timestamps for a correct class event (*time errors*).

Reduced dataset, small time frame, and loose threshold

This test is performed for evaluating the AuSPP performance on a reduced number of audio classes with a high time-resolution. In more detail, 10.000 audio tracks with 4 anomalous events (*breaking_glass, car_horn, gunshot, and siren*) are used for training and testing. The temporal resolution is set to 20ms, and the *loose* threshold δ is set to 0.8 to filter low probabilities from the predicted activity matrix $\hat{Y}$. As shown in Table 5.4, AuSPP outperforms both general and SED deep learning approaches for all the metrics, except for the precision metric.

However, for a security-driven system, the proposed model is preferable since it has a larger recall than the state-of-the-art PANNs (in this work, we select the best model, CNN14 [18]), with an improvement of 7.68%. From the results, the model pre-trained on AudioSet [79] does not outperform the other models even if it has been trained on a large annotated audio dataset. This behaviour could be caused by the reduced number of audio events to be classified. As a drawback, AuSPP shows a smaller value of precision than the other models.

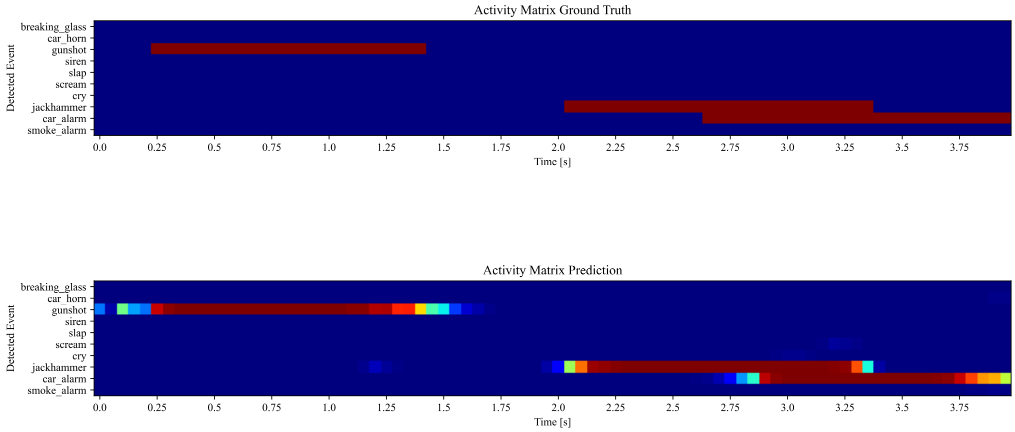


Figure 5.5: Example of a ground truth Y and a successful predicted $\hat{Y}$ from AuSPP activity matrices, respectively. In this example, two events are overlapping but the proposed model succeeds in distinguishing them with high probabilities.

More specifically, predicted onset and offset timestamps of events from AuSPP are less accurate, as it can be seen in Figure 5.5. This can be due to the low complexity of the model to state-of-the-art models.

Full dataset, small time frame, and loose threshold

In this case, a dataset of 10.000 audio samples is generated with all 10 audio class events. Hence, AuSPP ability to recognize multiple class anomalies is tested together with the other models. The results reported in Table 5.5 show that AuSPP achieves comparable results to the version of CNN14 [18] pre-trained on AudioSet [79]. It is worth noticing that AuSPP, with 75% fewer parameters than CNN14, does not require additional data. Therefore, AuSPP can be employed in edge computing devices where computational resources are limited.

Full dataset, small time frame, and strict threshold

Exploiting the aforementioned extended dataset, this test allows to analyze the predictions of all the models by adopting the *strict* threshold ($\delta = 0.9$). With this test, we evaluate which model is more confident, i.e. higher probability, of its predicted activity matrix. Table 5.6 shows that the proposed AuSPP outperforms state-of-the-art approaches with a Recall improvement of 4.74% over CNN14 [18], the second-best model. Similarly, with the other tests, our model is less precise with respect to the state-of-the-art pre-trained model on AudioSet.

Full dataset, large time frame, and strict threshold

Finally, we increase the time-frame from $20ms$ to $50ms$ and re-train all models in order to assess their performances. As shown in Table 5.7, AuSPP outperforms state-of-the-art SED models with a recall improvement of 5.16%. It is worth noticing that the proposed method achieves better performance without the ASPP module. Hence, using a larger time frame, it is preferable to avoid using the module since no improvements can be obtained.

Table 5.4: Performance of SOTA models and of the proposed approach on 4 classes with threshold $\delta = 0.8$ and time frame $T = 20ms$. We denote with ↑ when the performance is better when the metric is high and ↓ otherwise.

		Segment-based							
	Parameters ↓	$F2_c \uparrow$ (%)	$ER \downarrow$	$F2_g \uparrow$ (%)	$R_c \uparrow$ (%)	$P_c \uparrow$ (%)	$Acc \uparrow$ (%)	$Acc_M \uparrow$ (%)	$S_t \uparrow$ (%)
General-purpose models									
VGG16 [88]	145.36M	68.11	0.45	70.48	56.23	95.99	94.83	55.16	99.77
VGG19 [88]	150.67M	65.18	0.48	52.91	52.91	95.84	94.46	51.91	99.77
ResNet18 [89]	14.93M	73.79	0.38	75.69	63.24	95.16	95.54	61.81	99.69
ResNet34 [89]	25.04M	67.61	0.44	69.79	56.86	94.80	94.83	55.53	99.69
ResNet50 [89]	26.21M	73.33	0.38	75.32	62.84	95.44	95.50	61.42	99.70
ResNet101 [89]	45.20M	64.72	0.48	66.60	53.28	95.28	94.47	52.10	99.73
MobileNetV2 [91]	**7.55M**	71.37	0.41	73.48	59.95	96.47	95.25	58.90	99.78
MobileNetV3 [91]	9.53M	41.60	0.69	44.38	31.67	86.01	92.16	31.03	99.83
DenseNet121 [90]	11.76M	61.96	0.49	63.83	51.27	95.31	94.29	50.25	99.76
DenseNet169 [90]	18.60M	63.83	0.48	65.84	52.89	92.04	94.42	51.70	99.73
SED models									
CNN14 [18]	83.46M	69.00	0.44	71.22	56.89	**97.12**	94.96	56.13	99.83
Pre-trained CNN14 [18, 79]	83.46M	71.09	0.42	73.16	59.34	96.26	95.22	58.59	99.81
Wavegram-LogMel-CNN [18]	82.69M	70.78	0.41	72.96	59.70	95.11	95.14	58.27	99.69
Our Model									
AuSPP w/o ASPP	7.78M	74.48	0.36	76.48	65.13	93.50	95.60	62.78	99.53
AuSPP + ASPP	7.78M	**76.12**	**0.34**	**77.67**	**67.02**	92.95	**95.76**	**64.46**	99.47

Table 5.5: Performance of SOTA models and of the proposed approach on 10 classes with threshold $\delta = 0.8$ and time frame $T = 20ms$. We denote with ↑ when the performance is better when the metric value is high and ↓ otherwise.

		Segment-based							
	Parameters ↓	$F2_c \uparrow$ (%)	$ER \downarrow$	$F2_g \uparrow$ (%)	$R_c \uparrow$ (%)	$P_c \uparrow$ (%)	$Acc \uparrow$ (%)	$Acc_M \uparrow$ (%)	$S_t \uparrow$ (%)
General-purpose models									
VGG16 [88]	145.36M	49.63	0.61	54.39	37.01	89.58	97.08	36.06	99.90
VGG19 [88]	150.67M	43.46	0.67	48.56	31.53	86.27	96.85	30.81	99.92
ResNet18 [89]	14.93M	60.42	0.51	62.88	48.25	88.79	97.54	46.87	99.87
ResNet34 [89]	25.04M	28.25	0.78	30.15	21.58	51.44	96.32	20.78	99.88
ResNet50 [89]	26.21M	53.43	0.57	55.52	42.47	88.13	97.27	41.17	88.13
ResNet101 [89]	45.20M	28.10	0.78	30.44	21.30	64.67	96.36	20.56	99.92
MobileNetV2 [91]	**7.55M**	55.38	0.56	59.52	42.35	93.42	97.31	41.19	99.88
MobileNetV3 [91]	9.53M	1.19	0.99	1.48	0.74	8.35	95.47	0.72	99.99
DenseNet121 [90]	11.76M	33.94	0.74	36.24	25.70	64.88	96.51	24.48	99.86
DenseNet169 [90]	18.60M	26.21	0.80	28.04	19.50	53.42	96.28	18.97	99.94
SED models									
CNN14 [18]	83.46M	61.16	0.51	64.15	47.59	94.60	97.53	46.36	99.88
Pre-trained CNN14 [18, 79]	83.46M	**69.99**	**0.42**	**71.93**	57.05	**94.92**	**97.95**	**55.73**	99.88
Wavegram-LogMel-CNN [18]	82.69M	63.22	0.49	65.72	50.19	92.73	97.60	48.48	99.84
Our model									
AuSPP w/o ASPP	16.67M	64.14	0.47	66.48	52.57	86.85	97.56	49.32	99.69
AuSPP + ASPP	16.67M	68.21	**0.42**	70.23	**57.31**	88.15	97.77	53.77	99.69

Table 5.6: Performance of SOTA models and of the proposed approach on 10 classes with threshold $\delta = 0.9$ and time frame $T = 20ms$. We denote with ↑ when the performance is better when the metric is high and ↓ otherwise.

	Parameters ↓	$F2_c \uparrow$ (%)	$ER \downarrow$	$F2_g \uparrow$ (%)	$R_c \uparrow$ (%)	$P_c \uparrow$ (%)	$Acc \uparrow$ (%)	$Acc_M \uparrow$ (%)	$S_t \uparrow$ (%)
General-purpose models									
VGG16 [88]	145.36M	33.76	0.77	37.01	21.97	89.02	96.46	21.85	99.98
VGG19 [88]	150.67M	26.70	0.82	29.83	16.96	84.89	96.22	16.86	99.98
ResNet18 [89]	14.93M	55.10	0.59	57.83	40.37	96.95	97.27	39.92	99.95
ResNet34 [89]	25.04M	16.52	0.88	17.56	11.49	43.86	95.95	11.34	99.97
ResNet50 [89]	26.21M	47.58	0.64	49.49	35.22	85.82	97.02	34.76	99.95
ResNet101 [89]	45.20M	11.37	0.92	12.10	7.98	36.98	95.80	7.91	99.99
MobileNetV2 [91]	**7.55M**	41.39	0.71	44.71	28.18	96.15	96.74	27.97	99.97
MobileNetV3 [91]	9.53M	3.76	0.98	4.14	2.23	16.53	95.54	2.23	99.99
DenseNet121 [90]	11.76M	19.74	0.86	21.00	13.98	49.00	96.07	13.84	99.98
DenseNet169 [90]	18.60M	18.30	0.87	19.47	12.61	42.46	96.00	30.36	99.97
SED models									
CNN14 [18]	83.46M	44.58	0.69	47.36	30.57	97.55	96.84	30.36	99.97
Pre-trained CNN14 [18, 79]	83.46M	56.37	0.58	58.87	41.31	**97.58**	97.32	41.06	99.97
Wavegram-LogMel-CNN [18]	82.69M	48.09	0.65	51.00	33.93	96.73	96.97	33.57	99.95
Our model									
AuSPP w/o ASPP	16.67M	58.59	0.54	61.17	45.12	90.35	97.35	43.47	99.82
AuSPP + ASPP	16.67M	**59.49**	**0.53**	**61.98**	**46.05**	91.13	**97.40**	**44.40**	99.83

Table 5.7: Performance of SOTA models and of the proposed approach on 10 classes with threshold $\delta = 0.9$ and time frame $T = 50ms$. We denote with ↑ when the performance is better when the metric is high and ↓ otherwise.

	Parameters ↓	$F2_c \uparrow$ (%)	$ER \downarrow$	$F2_g \uparrow$ (%)	$R_c \uparrow$ (%)	$P_c \uparrow$ (%)	$Acc \uparrow$ (%)	$Acc_M \uparrow$ (%)	$S_t \uparrow$ (%)
General-purpose models									
VGG16 [88]	145.36M	35.28	0.76	38.69	23.30	87.77	96.42	23.13	99.97
VGG19 [88]	150.67M	31.93	0.78	35.34	20.95	88.04	96.30	20.79	99.98
ResNet18 [89]	14.93M	49.55	0.63	52.27	36.18	91.85	97.00	35.82	99.96
ResNet34 [89]	25.04M	20.21	0.86	21.39	14.12	48.25	95.92	13.90	99.93
ResNet50 [89]	26.21M	53.39	0.59	55.51	39.97	94.83	97.15	39.47	99.94
ResNet101 [89]	45.20M	31.86	0.76	33.47	23.17	66.65	96.39	22.92	99.97
MobileNetV2 [91]	**7.55M**	42.26	0.70	45.58	28.85	95.93	95.68	28.63	99.97
MobileNetV3 [91]	9.53M	1.79	0.989	2.00	1.07	12.38	95.36	1.07	99.99
DenseNet121 [90]	11.76M	32.33	0.77	34.61	21.92	81.71	96.34	21.74	99.97
DenseNet169 [90]	18.60M	33.28	0.76	35.34	23.60	70.91	96.41	23.39	99.97
SED models									
CNN14 [18]	83.46M	44.66	0.68	47.49	30.66	**97.91**	96.76	30.47	99.97
Pre-trained CNN14 [18, 79]	83.46M	56.42	0.58	58.80	41.34	97.68	97.25	41.09	99.97
Wavegram-LogMel-CNN [18]	82.69M	46.24	0.67	48.75	32.06	96.86	96.80	31.78	99.96
Our model									
AuSPP w/o ASPP	7.78M	59.89	**0.53**	62.57	**46.50**	91.84	97.36	44.94	99.84
AuSPP + ASPP	7.78M	**59.96**	**0.53**	**62.61**	46.41	92.40	**97.37**	**44.97**	99.85

5.5 Conclusion

In this Section, we propose AuSPP, a lightweight deep learning model that applies spatial filters to Mel-spectrogram to predict different anomaly classes with the corresponding onset and offset time. Moreover, we introduce SEDDOB, a human safety-oriented dataset that provides high-quality annotated audio waveforms for detecting anomalies on buses. To assess the performance of AuSPP, four tests on SEDDOB demonstrate that our AuSPP outperforms state-of-the-art general-purpose deep learning approaches with a reduced number of audio event classes. In addition, AuSPP achieves comparable results concerning SED models pre-trained on large-scale audio datasets with fewer learnable parameters.

However, the results show that significant improvements can be achieved. All considered models suffer from false alarms on normal recordings, significantly impacting on the task of ensuring human safety. As a possible solution, further studies on a coarse-to-fine approach for increasing the quality of audio prediction - in terms of recall and F-score metrics - could be performed. Moreover, tests could be conducted on a real scenario exploiting edge computing devices. Finally, model interpretability could be performed.

The content of this Chapter is associated with the following publication:

- **Michael Neri**, F. Battisti, A. Neri, and M. Carli, “Sound Event Detection for Human Safety and Security in Noisy Environments”, in: *IEEE Access*, 2022.

Chapter 6

Continuous Speaker Distance Estimation in Single-Channel Audio Signals

6.1 Introduction

In the previous Chapter on SED, we focused on developing models that can accurately identify and classify various sound events within an acoustic environment. The ability to detect and differentiate sound events, such as human speech, is a crucial step toward more advanced audio analysis tasks. However, identifying speech alone is not always sufficient, especially in applications where understanding the spatial context of the speaker is essential, e.g., speech separation and enhancement [92].

First, it is important to highlight that a SED model can identify the presence of speech, but it cannot distinguish between real and AI-generated ones. Regarding fake audio, in recent years, the development of generative deep learning architectures has led to increased concerns about the deepfake problem. Deepfakes are synthesized using AI algorithms - such as GANs, CNNs, and DNNs [93, 94] - to generate artificial media contents that are difficult to distinguish from real ones. As a result, this technology may be used with malicious intent to perpetrate attacks on people and institutions. Therefore, interest in deepfake generation and recognition is spreading, resulting in a strong interest in the research community. In addition, recognition of the synthesis method of a deepfake audio can provide information about the forger. However, to the best of our knowledge, this problem is in an embryonic stage [95].

Given that AI-generated speech, also known as audio deepfake, has gained increasing interest due to its potential misuse in security contexts, the job of audio analysis must go further than simply determining whether speech sounds real or has been synthetically generated. Another very important area of audio processing involves the ability to estimate a speaker's distance.

Accurately estimating the distance of a speaker has significant implications for various applications. In smart home systems, knowing how far a person is from a microphone can improve the system's ability to respond appropriately to voice commands. In surveillance and security contexts, distance estimation can help in assessing the potential threat level or identifying the location of an individual within a monitored space. Moreover, in teleconferencing systems, distance estimation can enhance the audio experience by adjusting the gain or applying spatial audio effects based on the speaker's proximity.

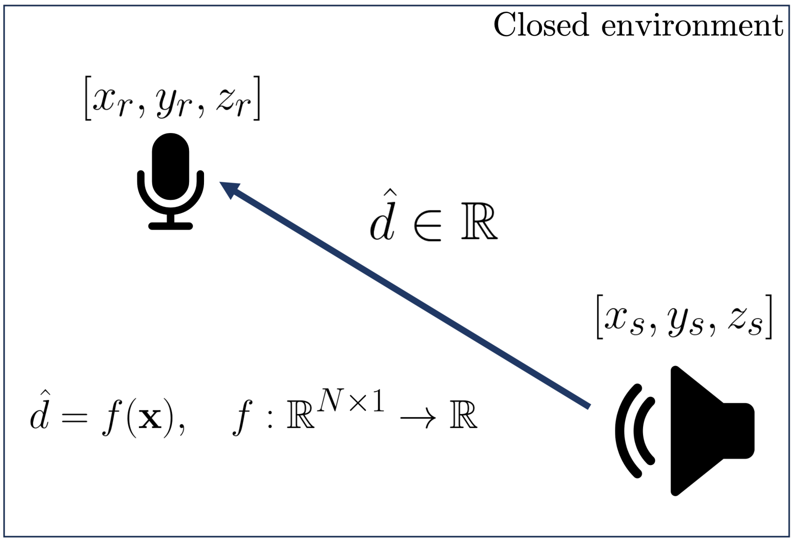


Figure 6.1: Definition of the continuous speaker distance estimation task.

It is often performed in conjunction with Direction of Arrival (DoA) estimation, in which only the direction information about the source position is obtained. Both tasks are useful in many practical applications, including increasing the robustness of automatic speech recognition [96] by enhancing the performance of acoustic echo cancelers [97] and autonomous robotics [98, 99]. Despite both DoA and speaker distance are estimated using multi-channel audio in most practical scenarios [100], the latter has been largely under-researched [101]. Firstly, speaker distance estimation is widely regarded as a more difficult task due to distance cues vanishing with the increased space between the sound source and the receiver. Secondly, DoA offers sufficient information in many downstream spatial filtering tasks. However, many applications such as source separation, acoustic monitoring, and context-aware devices would still benefit from full information about the sound source position, hence the need for further investigations on Source Distance Estimation (SDE).

In this direction, *speaker distance estimation* or, more generally, *source distance estimation* are considered tough challenges. Figure 6.1 visually explains the task, where a microphone in position $[x_r, y_r, z_r]$ captures the sound produced by a source/speaker at position $[x_s, y_s, z_s]$ in a closed environment. Specifically, we aim at estimating the distance $d \in \mathbb{R}^+$ between them, with solely the information of the position of the microphone, as it is generally known a-priori by design.

To solve these problems, state-of-the-art approaches involved binaural signals and hand-crafted features, i.e., estimation of DRR [102] and RIR, to mimic the human auditory system and to infer the distance of a generic source. In more detail, Vesa designed multiple Gaussian Mixture Models (GMMs) that extract features from the correlation between binaural channels to classify the distance [103, 104]. Georganti *et al.* exploited the standard deviation of the difference between the binaural signals to train an ensemble of GMMs and Support Vector Machines (SVMs) [105]. However, these methods suffer from complex hyperparameter tuning, impacting the generalization capabilities in environments that have differing acoustic conditions from the training dataset. Further, recent works provided experiments for a limited set of distances and rooms [106, 107]. Moreover, the employment of DNN in this topic has not been properly investigated by the research community.

Most methods for both DoA and distance estimation rely on arrays with more than two microphones [108]. Multichannel data allows for exploiting spatial cues such as Interchannel Time Differences (ITDs) and Interchannel Level Differences (ILDs) to provide information for efficient DoA estimation, positively affecting distance estimation as well [99]. However, using multiple microphones poses certain limitations in terms of budget and physical portability. To tackle this problem, some studies investigated using binaural recordings for that purpose, allowing for decreasing the number of channels to two by exploiting the human hearing cues [92, 109]. However, the simplest scenario of estimating distance from a single microphone has been largely under-researched [110]. Moreover, the vast majority of studies focus on a classification approach, in which the distance is discretized into a set of disjunctive categories, e.g., “far” and “near”, allowing for easier model training and a higher accuracy [105,111]. However, using pre-defined categories does not allow for continuous estimation, which puts limits on the precision of the obtained sound source position.

Only a few works have addressed the problem of speaker distance using single-channel audio [112]. The rationale behind the use of monophonic microphone signals or fixed beamformers is to reduce time and space complexity in low-power systems with limited computational resources [111].

One of the first works on this scope employed low-level features such as Linear Predictive Coding (LPC), skewness, and kurtosis of the spectrum to classify the distance bin of a speaker [111]. Regarding DNN approaches, Patterson *et al.* classified far and near speech. With this information, they were able to perform sound source separation from single-channel audio signals [110].

To the best of our knowledge, single-channel *speaker distance estimation* has never been estimated but classified in quantised values. Most of the studies involved binary classification [109,110], i.e., far and near, or classification into distance bins of a short range (5 classes from 0 to 3 meters) [111]. In addition, the use of DNNs in this novel scenario is under-researched.

In this Chapter, a new task of *speaker distance estimation* is proposed to estimate continuous distance values rather than discrete bins, differently from previous state-of-the-art works.

The effectiveness of our approach is assessed by simulating different configurations of room shapes, materials, and locations of the microphone and the speaker. By doing so, the method generalizes to rooms and locations that are not present in the training set. Moreover, the idea is to exploit reverberation cues, thus without any a-priori knowledge of the room, to estimate the distance. This characteristic is fundamental as it is generally unfeasible to collect the acoustic parameters of a room. Our results demonstrate that the proposed technique provides speaker distance estimation with an absolute error in the order of centimeters in noiseless conditions. The ablation study demonstrates that phase-related features, i.e., applying the $\sin(\cdot)$ and $\cos(\cdot)$ functions to the raw phase of the STFT, are the most representative features for estimating the speaker distance, implicitly modelling the RIR. Finally, we show that additive noise makes the distance estimation task significantly more challenging.

The contributions of this Chapter can be summarized as follows:

- Definition of a learning-based synthetic speech classifier which analyzes MFCC and GTCC to enhance the artifacts caused by deepfake generators. ParalMGC achieves the highest value on our validation accuracy split on the 2022 IEEE Signal Processing Cup dataset.

- The task of *speaker distance estimation* is proposed to estimate continuous distance values rather than discrete bins, differently from previous state-of-the-art works.

- A deep learning-based baseline for solving the SDE is devised. Our approach consists of a CRNN, which processes acoustic features such as the sine and cosine of the STFT phase of the single-channel audio recording. In addition, further experiments demonstrate how phase features are effective in high SNR scenarios.

- To better estimate the speaker distance estimation, an attention module is proposed. Specifically, it enables the explainability of predictions, providing time-frequency patterns that are employed for the distance estimation.

- We conduct extensive experiments using audio recordings in controlled environments with three levels of realism (synthetic room impulse response, measured response with convolved speech, and real recordings) on four datasets (our synthetic dataset, QMULTIMIT, VoiceHome-2, and STARSS23).

- Experimental results show that the model achieves an absolute error of 0.11 meters in a noiseless synthetic scenario. Moreover, the results showed an absolute error of about 1.30 meters in the hybrid scenario. The algorithm's performance in the real scenario, where unpredictable environmental factors and noise are prevalent, yields an absolute error of approximately 0.50 meters. All the codes and datasets are available in a dedicated public repository.

6.2 Synthetic Speech Attribution

Several works related to the problem of classifying the algorithm that generated a given audio signal consider deepfake audio detection. In [113], a deepfake detection approach using GMMs and Maximum A Posteriori Probability (MAP) is considered. The authors propose to classify an input audio as genuine or fake based on its comparison with two reference GMM distributions. However, this approach is applicable only if the data points follow mixtures of a Gaussian distribution. For this reason, several approaches exploiting advanced machine learning techniques and neural networks have been developed.

In [114], a machine learning-based classification approach is presented. The system exploits bispectral analysis to detect high-order correlations within audios. According to the authors, these features are not easily counterfeited by a human spoken audio, thus simplifying the comparison between real and fake audios. Bicoherence magnitude and phase are extracted, and low-level features are employed by the Quadratic-SVM classifier. Unfortunately, machine learning approaches suffer from a lack of generalization capabilities due to their model-driven nature.

In [115], a CNN architecture is presented to detect the audio generated by a GANs architecture. In this study, MFCCs are extracted as features in the pre-processing stage. These feature sets are then processed to distinguish images of manipulated faces from authentic ones by combining CNNs and RNNs. Other works consider different types of neural networks, such as ResNet [116] and SENet [117]. ResNet uses skip connections between the input and the output in order to avoid the vanishing gradients problem. The SENet adds to the ResNet model a squeezing operation, which produces a channel descriptor by aggregating feature maps across their spatial dimension.

Borrelli *et al.*, in [118], have tried to deal with the problem of the synthetic human speech attribution. In particular, they designed an architecture that can determine whether a specific speech is synthetic or not, and in the case of a positive answer, identify which algorithm generated it. The short- and long-term traces have been extracted from audios in input and fed to three different classifiers: random forest, linear SVM, and a radial basis function kernel SVM.

Transfer learning methodologies involve the use of pre-trained neural networks [119]. Nowadays, these techniques are popular since they ease the training process through an initial set of weights adapted from a previous task, improving the generalization capabilities of the model. Transfer Learning approaches have been successfully applied in spoofing audio detection. Martín-Doñas *et al.* [119] considers the pre-trained Wav2Vec2 architecture as a feature extractor, which elaborates the raw audio waveform through a CNN feature encoder and several transformer-based blocks. The extracted features are then combined with the memory states of each transformer block and fed as input to an MLP network.

6.2.1 Proposed Approach

Several 2D audio features have been analyzed to extract discriminative characteristics of human speech and synthesis artifacts. Specifically, the features in the time-frequency and cepstral domain have been considered. To this aim, two types of spectrograms have been analyzed and processed by an image processing pipeline: Mel-spectrogram and Bark-spectrogram. In this

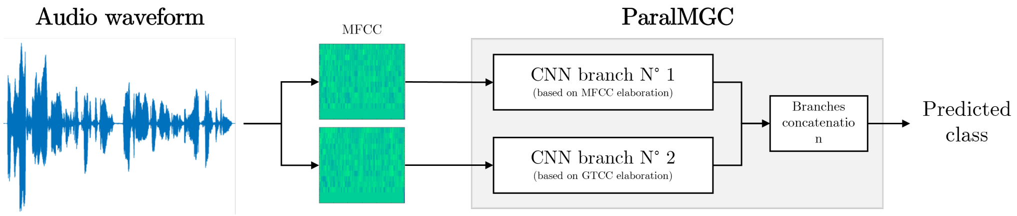


Figure 6.2: Structure of the proposed framework.

kind of representation, a STFT analysis is computed first.

Let $\mathbf{x} : [0, \dots, L-1] \to \mathbb{R}$ be a raw single-channel audio track with L samples and f_s be the sampling frequency in Hertz. Moreover, let $\mathbf{w} : [0, \dots, N-1] \to \mathbb{R}$ be a window function of N samples and $H \in \mathbb{N}$ be the hop size which determines the overlap between two consecutive time frames in samples. Then, the complex-valued STFT of the input signal X_{STFT} is evaluated as

$$X_{STFT}[m,k] = \sum_{n=0}^{N-1} \mathbf{x}[n+mH]\mathbf{w}[n]e^{\frac{-2\pi i k n}{N}}, \tag{6.1}$$

where $m \in [0, \dots, M-1]$ and $k \in [0, \dots, K]$ are the time and frequency bins, respectively. The Mel-Spectrogram X_{Mel} is computed by means of the Mel-Filterbank $H_{Mel}(\cdot)$ on the squared magnitude of the STFT

$$X_{Mel}[m,k] = H_{Mel}(|X_{STFT}|^2). \tag{6.2}$$

Similarly to the Mel-spectrograms, Bark-spectrograms are obtained by using Bark filterbanks $H_{Bark}(\cdot)$

$$X_{Bark}[m,k] = H_{Bark}(|X_{STFT}|^2). \tag{6.3}$$

The bark scale can be used to measure the critical band at which loudness becomes significantly different, while the Mel scale is suitable for pitch perception and phonetic information [120].

Another representation, which is widely used in the literature due to its high discriminating power in the field of sound classification, is based on the MFCCs [115]. MFCCs are extracted by computing the DCT on the logarithm of the amplitude of the Mel-Spectrogram, as shown in Equation (6.4).

$$MFCC[m,k] = \mathrm{DCT}_2(\log X[m,k]_{Mel}), \tag{6.4}$$

where $MFCC[m,k]$ refers to the (k^{th}) MFCC coefficient evaluated for the (m^{th}) audio frame.

Two additional features can be derived from MFCCs to achieve a better description of the signal: MFCC delta and MFCC delta delta. MFCC delta represents the difference between the cepstral vectors extracted from one frame and the cepstral vectors extracted from the previous one. MFCC delta is computed as the difference between two MFCC deltas, referred to as two adjacent audio frames. Another possibility is to exploit the GTCCs [7].

While MFCCs are computed considering a Mel-filterbank, GTCCs are extracted considering GammaTone filters: once the Equivalent Rectangular Bandwidth (ERB) spectrum is obtained, the coefficients extraction procedure follows the same procedure as MFCCs. To the best of our knowledge, this is the first time that GTCCs are used for audio deepfake classification.

Another possible representation, often used for music signals analysis, is the Chromagram [121]. The Chroma feature is obtained by filtering the STFT to represent the audio pitch in a detailed way. Pitch is related to how the human hearing system perceives different sounds characterized by different frequencies. Pitch can be decomposed into two different components, which are referred to as tone height and Chroma.

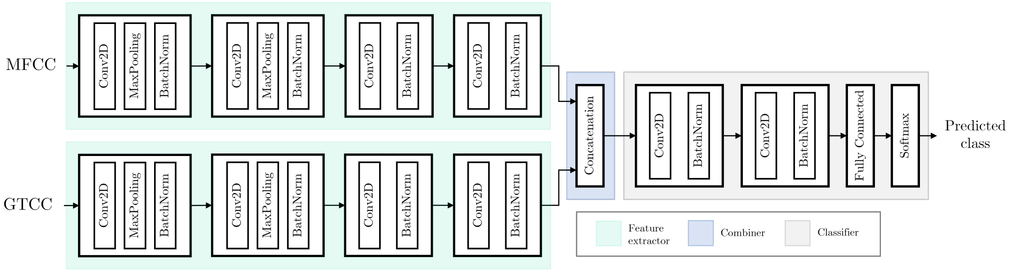


Figure 6.3: Structure of ParalMGC.

In addition, we propose a deep learning model that exploits MFCC and GTCC features. In more detail, it consists of two CNN parallel branches (as shown in Figure 6.2) with four convolutional blocks, characterized by the same parameters except for the number of filters: the deeper the convolutional layer, the higher the number of filters. This choice makes it possible to address two problems: the reduction of the dimensions of the feature set extracted by the convolutional layers and the need for capturing more complex combinations of patterns [122, 123].

Each of these blocks is composed by a convolutional layer with kernel size 5 × 5 and a Rectified Linear Unit (ReLU) activation function. Max Pooling is then applied to reduce the dimensionality of the feature maps obtained by the convolutional layers, to discard redundant information, and to reduce the complexity of the network [18]. The pooling is used only in the first two blocks of the structure so as not to excessively reduce the

Table 6.1: Description of the proposed ParalMGC.

Input: Spectrograms X_{Mel} and X_{Bark}

<table>
<tr><td>Conv2D(1, 16, 5, 2, “same”)
ReLU
Max Pooling 2 × 2, stride = 2
BatchNorm</td><td>Conv2D(1, 16, 5, 2, “same”)
ReLU
Max Pooling 2 × 2, stride = 2
BatchNorm</td></tr>
<tr><td>Conv2D(16, 32, 5, 2, “same”)
ReLU
Max Pooling 2 × 2, stride = 2
BatchNorm</td><td>Conv2D(16, 32, 5, 2, “same”)
ReLU
Max Pooling 2 × 2, stride = 2
BatchNorm</td></tr>
<tr><td>Conv2D(32, 64, 5, 2, “same”)
ReLU
BatchNorm</td><td>Conv2D(32, 64, 5, 2, “same”)
ReLU
BatchNorm</td></tr>
<tr><td>Conv2D(64, 128, 5, 2, “same”)
ReLU
BatchNorm</td><td>Conv2D(64, 128, 5, 2, “same”)
ReLU
BatchNorm</td></tr>
<tr><td>Conv2D(128, 256, 5, 2, “same”)
ReLU
BatchNorm</td><td>Conv2D(128, 256, 5, 2, “same”)
ReLU
BatchNorm</td></tr>
<tr><td colspan="2">Concatenation on the channel axis</td></tr>
<tr><td colspan="2">Conv2D(512, 128, 5, 2, “same”)
ReLU
BatchNorm</td></tr>
<tr><td colspan="2">Conv2D(128, 256, 5, 2, “same”)
ReLU
BatchNorm</td></tr>
<tr><td colspan="2">Flatten to 1<i>D</i> and Fully Connected
Softmax activaction function</td></tr>
</table>

Output: $\hat{Y}$ Classification

dimensionality of the feature maps. Finally, batch Normalization is applied to counteract the internal covariate shift, avoid overfitting, and solve the problem of vanishing gradients.

After the four convolutional blocks, a fully connected layer with some neurons equal to the number of classes is added. The outputs of this layer are then converted into the probabilities that the instance belongs to a certain class through the softmax function. Thus, the output of the network will be the predicted class that maximizes the aforementioned probabilities.

In this direction, we have designed three different models composed by parallel branches, which allow to enrichment of the information extracted by a single serial network. The first one, ParalMGC, takes in input both MFCCs and GTCCs of the audio under analysis. These features are processed by two different branches, each of which consists of the CNN ar-

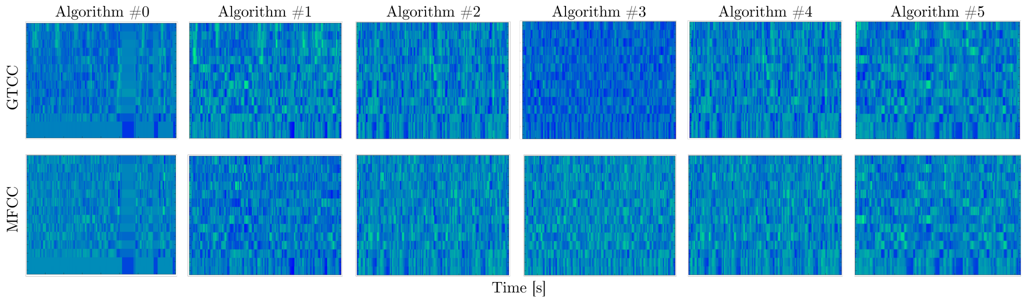


Figure 6.4: Examples of 2D MFCC and GTCC features for each synthetic audio class.

chitecture without the final dense layer. The two feature maps derived by the two branches are then concatenated and processed by two convolutional blocks.

Similarly, DParalMC considers two different branches, each taking in input the MFCCs of an audio. The first branch is again the CNN architecture deprived of the final layer, while the second one is very similar to the first since it is equal but the convolutional layers are dilated (with a dilation factor equal to two) to capture long - term correlations of genuine and synthetic human speech. Again, the two feature maps are concatenated and processed by two convolutional blocks, each consisting of a convolutional 2D layer with ReLU as an activation function, batch normalization, and pooling.

In addition, we implement a more complex neural network, DParalMGC, with three different branches. In particular, DParalMGC consists of two branches elaborating MFCCs (equal to the two composing ParalMC) and the last processing GTCCs (similar to the second one of ParalMGC). Finally, a convolutional block processes the set of features obtained by concatenating features extracted from each branch: in this case, only one convolutional block is considered after the combination of the information provided by the different branches so as to reduce the already high complexity of the network and thus to prevent overfitting.

Figure 6.3 represents the overall architecture of ParalMGC. Further details on the network parameters are listed in Table 6.1 where Conv2D(C_{in}, C_{out}, k, s, padding) is a generic 2D convolutional layer with C_{in} and C_{out} input and output channels, respectively, k is the kernel size, s is the stride, and padding is the padding technique employed.

6.2.2 Experiments

The dataset published for the 2022 IEEE Signal Processing Cup has been used. The synthetic dataset consists of four groups of audio files. The first group contains 5000 tracks whose labels are associated with five known generative algorithms. The second group consists of 1000 audio files generated by an unknown synthesis algorithm. Both groups have been used for training the developed models. For the first group, the labels varied from 0 to 4, while each audio of the second group is labelled as 5. Regarding the training and the validation of the developed model, the labelled audios have been grouped into a single balanced dataset of 6000 audio tracks. The 80% of these audio waveforms has been used for the training process, while the remaining 20% has been used for validation. This *ad-hoc* split of the dataset has been done since the test does not have available labels.

All the methods have been compared, considering the accuracy achieved on the audio signals of the validation set.

The parameters characterizing the feature extraction have been chosen to maximize the accuracy on the validation set. In our case, the best window length for the computation of the STFT is quite short (512 samples) with $H = 192$ overlapping samples. A wider window in the frequency domain results in the smoothing of the signal's higher frequencies, thus causing a low-pass filtering effect. For this reason, a narrower window guarantees better performance. Furthermore, 40 MFCCs and GTCCs are extracted from each audio. Figure 6.4 shows an example of each algorithm's features. Extracting a larger number of coefficients leads to information redundancy and, thus, performance degradation.

Given the limited dimensions of the training set, for the architectures that have shown the best performances on the validation dataset, data augmentation has been performed to increase the ability of the network to generalize. Different techniques, widely used in the literature [124], have been applied to augment data from the training set, such as time shifts, volume gain, pitch shift, stretch in time, white noise addition, and amplitude inversion (which modifies the phase of the spectrogram). This increases the number of listeners to be trained and trains the networks with more sophisticated instances that make the architecture more capable of generalization.

The weights of the ParalMGC network have been initialized throughout the Glorot [125] uniform initialization, which consists in setting the weights in random values drawn from a uniform distribution with zero mean and standard deviation, which depends on the dimensions of the features in input and output of the layer. Adam optimizer has been considered both with default parameters and with learning rate scheduling, obtained multiplying every 40 epochs for 0.2 the current learning rate, starting with a value of

0.001. The network has been trained considering as loss function as the multi-class categorical cross-entropy.

Table 6.2: Results of the implemented models on the validation set.

Proposed models	Accuracy
MFCC + CNN	96.7%
MFCC + LSTM	97.1%
GTCC + CNN	95.4%
Chroma + CNN	90.4%
Chroma + LSTM	85.7%
MelSpectrum + CNN	91.3%
Bark-Spectrum + CNN	92.0%
VGGish features + CNN	93.6%
ParalMGC (ours)	**98.1**%
ParalMC (ours)	97.9%
DParalMGC (ours)	98.0%

Additional models, concerning the ones previously described, have been tested for comparison purposes. According to the results, ParalMGC outperforms the other developed models. The model takes both MFCCs and GTCCs of the audio as input, and two different branches process the two different feature sets. Their outputs are then concatenated and provided as input to the classifier. According to Table 6.2, MFCCs constitute the best feature set and GTCCs allow to explore a different domain (the Gammatone one), which enriches the information provided by the MFCCs.

6.2.3 Summary

The synthetic human speech attribution (or audio deepfake) is a well-known open problem in the audio research community. To solve it, several models have been developed and tested on three different datasets. Despite the challenging task, the proposed ParalMGC model has proven to be effective in achieving promising performances. Table 6.2 shows that the proposed model reaches an extremely high accuracy (98.1%) on the validation set. However, real human speech recordings are not present in the dataset. Hence, tests on a more extended dataset of audio deep fake, ASV spoof 2021 [126], will be carried out in the future. Another main problem is the open-set scenario. More specifically, classifying new synthetic speech is challenging for supervised-driven approaches. In this direction, a few-shot continual learning, i.e., learning from new audio deepfake algorithms from few samples, has been considered as a future work.

6.3 Feature Selection for Acoustic Modeling in Synthetic Dataset

The previous Section focuses on whenever a speech recording is real or synthetically generated. Now, it is possible to perform SDE on genuine speech data, which involves determining the distance between a sound source and the receiver. When compared to the DoA estimation, SDE is an area that has received significantly less attention and is generally considered more challenging. This is primarily since the accuracy of distance estimation declines rapidly for small-sized arrays commonly used in practice, even for relatively short distances from the center of the array (up to 3-4 m). Several factors contribute to this phenomenon, including:

- the decrease in DRR and SNRs as the source distance increases;
- the reduction in inter-channel level differences and constant inter-channel time differences as the source transitions from a spherical wave to a plane wave captured by the array.

The majority of studies related to SDE show results in conjunction with the DoA estimation task. Extensive research has been conducted on this subject for various acoustic systems that commonly use distributed microphone arrays. These systems encompass a range of setups, such as intelligent loudspeakers [127], spherical microphones [128], triangular configurations [129], and arrays of acoustic sensors [130]. Simpler audio formats, including binaural recordings, have been investigated to a much lesser extent, including few studies with classical machine learning methods [99, 131] and very limited research related to deep learning [92, 109].

Regarding SDE modeling in isolation, most of the research has been focused on parametric approaches and manually crafted features. These methods often utilize information such as the DRR [102], RIR [132], or signal statistics and binaural cues such as the Interchannel Intensity Difference (IID) [99]. In some cases, classical machine learning techniques have been employed to leverage statistical features. For instance, a study by Brendel *et al.* estimated the coherent-to-diffuse power ratio to determine the source-microphone distance via GMMs [101]. Vesa utilized GMMs trained with Magnitude Squared Coherence (MSC) features to incorporate information about channel correlation [103, 104]. In [133], the authors used MSC on top of other features to train classifiers with methods such as KNNs or Linear Discriminative Analysis (LDA). Georganti *et al.* introduced the Binaural Signal Magnitude Difference Standard Deviation (BSMD-STD) and trained GMMs and SVMs using this feature [134]. Most of these methods rely on compound algorithms that require careful tuning to adapt to varying acoustic conditions.

Until now, the exploration of source distance estimation using DNNs has been quite limited. Yiwere *et al.* employed an approach inspired by image classification, utilizing CRNNs trained on log-mel spectrograms to classify three different distances in three distinct rooms [106]. Although the models demonstrated promising outcomes for data within the same environment, their performance significantly deteriorated when dealing with recordings from different rooms. In another endeavor, Sobghdel *et al.* introduced relation networks to address this challenge through few-shot learning, which exhibited enhancements over conventional CNNs [107]. Both studies conducted tests within a limited range of specific distances, encompassing a proximity of up to 3-4 meters at most. In [109], the authors conducted experiments for data covering distances for up to 8 m, however, the model was classifying them into two binary classes denoted as “far” and “near”.

Additionally, only a few works have addressed the topic of speaker distance estimation using single-channel audio. One of the first works employed low-level features such as LPC, skewness, and kurtosis of the spectrum to classify the distance of a speaker [111]. Venkatesan *et al.* proposed both monaural and binaural features to train GMMs and SVMs [112]. Regarding DNN approaches, Patterson *et al.* classified “far” and “near” speech to perform sound source separation from single-channel audios [110].

To the best of our knowledge, single-channel source distance estimation has been scarcely addressed as a regression problem, prioritizing classification approaches to ease model training. In addition, there are very few studies investigating the use of DNNs in this task.

For the above reasons, a learning-based approach for the continuous estimation of the distance of the speaker is proposed. A first step towards continuous sound source distance estimation is proposed in the next Section [135], where a CRNN is defined for estimating static speaker distance in simulated reverberant environments from a single omnidirectional microphone.

However, this study was evaluated only on simulations, while in the last Section, various degrees of realism are investigated, from simulated RIRs to synthetic data with measured RIRs, to fully real recordings with distance-annotated sources. Hence, the potential of the method in a real-world scenario is demonstrated in the last Section. In addition, the preliminary study was based on a simpler architecture without an investigation of what architectural components contributed the most to the SDE, while here, the architecture is refined and enhanced, with better overall performance and specific choices investigated in an ablation study.

6.3.1 Definition of the Baseline

Let $x[n]$ be a single-channel audio representing the speech of a single speaker captured by a microphone in a room with an unknown RIR $h[n]$. The objective of this work is to estimate the continuous-valued speaker distance $\hat{y} \in \mathbb{R}$ from the single-channel audio, that is the mapping $g(\cdot) : \mathbb{R}^{1 \times L} \to \mathbb{R}$, where L is the number of samples of the audio recording. To this aim, acoustic features are extracted using the STFT. To model temporal, spatial, and spectral features, a CRNN is employed for the experiments. This type of model has shown promising results in many studies for Sound Event Localization and Detection (SELD) tasks [20, 21]. The architecture is depicted in Figure 6.5.

In more detail, a feature extractor is applied to $x[n]$ to obtain the complex STFT. The transform $\text{STFT}\{x[n]\}$ is computed with a Hanning window of length 32 ms and 50% overlap with sampling frequency $f_s = 16$ KHz. Then, magnitude $|\text{STFT}\{x[n]\})|$ and phase $\angle\text{STFT}\{x[n]\})$ of the STFT are computed. In addition, we extract the sin&cos features for each time-frequency point of the STFT raw phase, i.e., $\sin(\angle\text{STFT}\{x[n]\})$ and $\cos(\angle\text{STFT}\{x[n]\})$, to model the early and late reverberation cues that characterize the audio. The sin&cos phase representation avoids phase wrapping and is advantageous over raw phase in several tasks [136, 137]. Finally, the STFT magnitude and the sin&cos features are arranged in a $T \times F \times 3$ tensor to be processed by the three convolutional layers, where F and T are the number of frequency and time bins, respectively. By doing so, the three input features are assigned to the convolutional channel dimension.

Each convolutional block consists of a 2D convolutional layer containing $P = 128$ 1×3 filters followed by a batch normalization. Max and average pooling operations are computed in parallel along the frequency dimension to be summed. The applied activation function is the ELU [17]. The max and average pooling rate of each layer is $MP = \{MP_1, MP_2, MP_3\} = \{8, 8, 2\}$.

Two bidirectional GRU layers, with $\tanh(\cdot)$ as the activation function, are applied to the feature maps from the convolutional layers. It has shown promising results in audio and speech processing tasks with fewer parameters than LSTM networks [138]. In fact, in the proposed setup, most of the

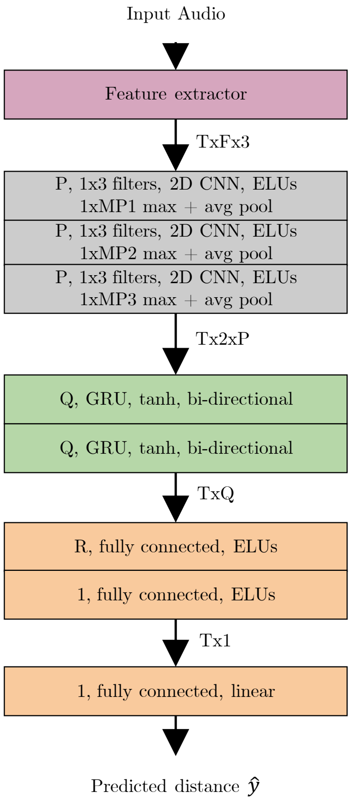

Input Audio
Feature extractor
TxFx3
P, 1x3 filters, 2D CNN, ELUs
1xMP1 max + avg pool
P, 1x3 filters, 2D CNN, ELUs
1xMP2 max + avg pool
P, 1x3 filters, 2D CNN, ELUs
1xMP3 max + avg pool
Tx2xP
Q, GRU, tanh, bi-directional
Q, GRU, tanh, bi-directional
TxQ
R, fully connected, ELUs
1, fully connected, ELUs
Tx1
1, fully connected, linear
Predicted distance $\hat{y}$

Figure 6.5: Model architecture for speaker distance estimation. The shape of each output is reported for the sake of clarity.

information regarding reverberation is extracted from time-wise features. In this implementation, each GRU has $Q = 128$ neurons for each time bin T. Finally, three fully connected layers are used to return the predicted distance $\hat{y} \in \mathbb{R}$. More specifically, the first linear layer projects time-wise features from the last GRU into a matrix of dimension $T \times R$ with $R = 128$. Then, the second linear layer maps from $T \times R$ to a vector of size $T \times 1$. Finally, the last fully connected layer is employed to regress the predicted distance $\hat{y} \in \mathbb{R}$.

The overall architecture is optimized using the MSE loss, as it maximizes the mutual information between predicted and ground truth distances for 60 epochs using the ADAM optimizer with 16 samples per iteration and learning rate $\lambda = 0.001$. Tests on L_1 loss, i.e., MAE, yielded non-converging training processes.

6.3.2 Synthetic Dataset

The dataset used for experiments follows the same setup as in [139]. Briefly, anechoic speech recordings obtained from the TIMIT dataset [140] are convolved with the simulated omnidirectional RIRs from an image-source room simulator for shoebox geometries [141].

This simulator allows for frequency-dependent wall absorption and directional encoding of image sources in 5th order Ambisonics format. The elevation range between the source and the receiver spanned from $-35°$ to $35°$. To compile a list of materials and their respective absorption coefficients for each surface type (ceiling, floor, and wall), we refer to widely used acoustical engineering tables [142]. For each unique simulated room with its room-source-distance configuration, a random material is assigned to each surface, resulting in 2912 possible material combinations. Compared to directly randomizing the target RT60 for each simulated room, this randomization approach allows us to avoid matching unnatural reverberation times to specific room volumes (e.g., a very long RT60 for a small room) and ensure a more natural distribution of reverberation times.

The final distribution of reverberation times exhibits a median, 10th percentile, and 90th percentile of 0.83 s, 0.42 s, and 2.38 s, respectively. Furthermore, the positions of the sound sources are uniformly distributed in terms of the azimuth angle relative to the receiver.

The experiments include 2500 audio files of 10 s duration at 16 kHz in compliance with the speech dataset. The samples are assigned to 5 folds to assess the performance in a 5-fold cross-validation fashion. By doing so, each iteration assigns 1500, 500, and 500 audios to training, validation, and testing sets, respectively. It is worth noticing that the room characteristics and the speaker are different across all the sets. Therefore, no information from similar room patterns and speech utterances can be exploited during training.

Table 6.3: Parameters for data generation.

Parameter	**Random ranges**
Room width and length	$[3.0, 15.0]$ m
Room height	$[2.0, 7.0]$ m
# of materials (wall, floor, ceiling)	$13, 7, 8$
Source - receiver height	$[1.5, 2.2]$ m
Source-to-surface distance	> 0.5 m
Source-to-receiver distance	> 1.0 m

To assess the performance of the proposed approach under different noise levels, real background noise is added to the synthetic dataset. Specifically, environmental noise recordings from the WHAM! [143] dataset, captured in various urban settings such as restaurants, cafes, and bars, are employed. Random segments of the same length as the simulated speech recordings are injected, mirroring the same split as the WHAM! dataset, with several SNRs levels ([50, 40, 30, 20, 10, 5, 0] dB). Table 6.3 depicts the range of random parameters for data generation.

6.3.3 Experiments

Clean Speech

Table 6.4: Experimental results on clean speech. All the errors are in meters.

SNR = $+\infty$

Distance ranges	**Errors**
E_{MAE}	0.22 ± 0.03
$E_{MAE_{[1,2)}}$	0.13 ± 0.03
$E_{MAE_{[2,4)}}$	0.16 ± 0.04
$E_{MAE_{[4,6)}}$	0.30 ± 0.05
$E_{MAE_{[6,8)}}$	0.30 ± 0.08
$E_{MAE_{[8,10)}}$	0.44 ± 0.14
$E_{MAE_{[10,14)}}$	0.45 ± 0.18

The proposed approach efficiently estimates speaker distance with an average error of 22 cm, as can be inspected in Table 6.4. However, it is notable from the scatter plot shown in Figure 6.6 that the predictions are slightly underestimated when the speaker is more than 7 meters from the microphone. This behavior is expected since at such distances, the late reverberant portion of the signal is dominant compared to the direct and early reflection portion of the signal. These dominant late reverberation cues are statistically diffuse [144], i.e., short-term magnitudes and phases resemble noise, and it may be difficult for the model to extract effective information from them. The reason for late reverberant cues dominating is that the intensity of the direct and early echo portion of the microphone signal I_s is inversely proportional to the square of the distance d, i.e. $I_s \propto \frac{1}{d^2}$, while the signal power of the late reverberant component remains more-or-less independent of the source and receiver position.

Overall, the model fits the dataset well, as pointed out by the high value of the coefficient of determination R^2 between predicted and true distances, which is evaluated as:

$$R^2 = 1 - \frac{\mathrm{SSR}}{\mathrm{SST}}, \tag{6.5}$$

where SSR and SST are the sum squared residuals and the total sum of squares, respectively.

Table 6.4 depicts the errors in a noiseless environment. In addition, Figure 6.6 shows the scatter plot between the predicted and ground truth distance to evaluate the bias of the model.

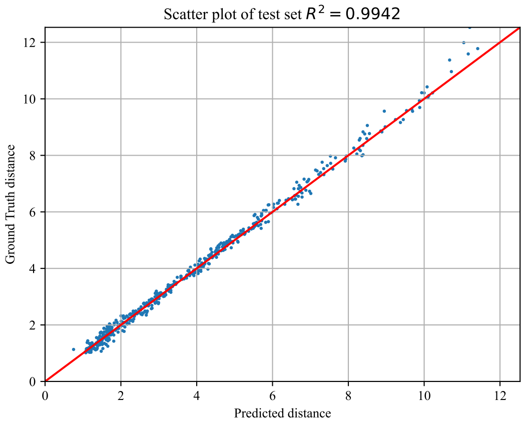


Figure 6.6: Scatter plot of predicted and ground truth distances with STFT and sin&cos as features set with SNR = $+\infty$.

Ablation Study of Acoustic Features

Table 6.5: Ablation studies of acoustic features. All the errors are in meters.

	E_{MAE}	$E_{MAE_{[1,2)}}$	$E_{MAE_{[2,4)}}$	$E_{MAE_{[4,6)}}$	$E_{MAE_{[6,8)}}$	$E_{MAE_{[8,10)}}$	$E_{MAE_{[10,14)}}$
|STFT|	0.37 ± 0.04	0.24 ± 0.04	0.32 ± 0.04	0.37 ± 0.02	0.54 ± 0.10	0.63 ± 0.11	0.78 ± 0.19
sin&cos	$\mathbf{0.16 \pm 0.01}$	$\mathbf{0.10 \pm 0.01}$	$\mathbf{0.11 \pm 0.02}$	$\mathbf{0.24 \pm 0.06}$	$\mathbf{0.19 \pm 0.02}$	$\mathbf{0.31 \pm 0.08}$	$\mathbf{0.35 \pm 0.14}$
sin&cos + |STFT|	0.22 ± 0.03	0.13 ± 0.03	0.16 ± 0.04	0.30 ± 0.05	0.30 ± 0.08	0.44 ± 0.14	0.45 ± 0.18

To study the impact of each feature, an ablation study has been performed. Table 6.5 depicts the MAE for each distance bin with their confidence interval. It is worth noticing that most of the distance information is represented by the phase of the STFT. Using only magnitude yields poor performance over all the distance bins. Contrarily, the configuration using only the sin&cos input reaches better performances to the proposed feature set. However, the same phase-only features perform poorly in the noisy scenario, yielding higher errors, i.e., random guess, than the combination of magnitude with the sin&cos phase.

Noisy Speech

To evaluate the performance of the proposed approach at different levels of noise, real background noise is injected into the simulated dataset. To this aim, the WHAM! [143] dataset has been exploited. In more detail, the environmental noise recordings from WHAM! dataset were collected at various urban locations such as restaurants, cafes, and bars. In addition, this database was split into training, validation, and testing sets. Following the same split provided by the WHAM! dataset, background noise samples are injected randomly into the simulated dataset. More precisely, training noise affects only training samples of the simulated dataset. This behavior also occurs for validation and testing scenarios. By doing so, the proposed approach does not infer information from the same background noise in the training split.

To assess the effectiveness of the method, 7 SNR values have been defined to measure the quality of the predictions concerning noise strength. Table 6.6 depicts the MAEs with confidence intervals for each SNR scenario and for each distance bin. The comparison between noiseless and noisy scenarios highlights the large discrepancy of these results. This is mostly due to the phase information, i.e. $\sin(\angle\text{STFT}\{\mathbf{x}\})$ and $\cos(\angle\text{STFT}\{\mathbf{x}\})$, which is disrupted by the background noise.

Table 6.6: Experimental results on noisy speech with fixed SNR and STFT and sin&cos as features set. All the errors are in meters.

	E_{MAE}	$E_{MAE_{[1,2)}}$	$E_{MAE_{[2,4)}}$	$E_{MAE_{[4,6)}}$	$E_{MAE_{[6,8)}}$	$E_{MAE_{[8,10)}}$	$E_{MAE_{[10,14)}}$
SNR $= 50_{dB}$	0.90 ± 0.24	0.51 ± 0.17	0.75 ± 0.15	0.84 ± 0.26	1.22 ± 0.38	2.01 ± 0.82	3.11 ± 1.01
SNR $= 40_{dB}$	1.16 ± 0.09	0.67 ± 0.11	0.95 ± 0.13	1.16 ± 0.11	1.58 ± 0.18	2.45 ± 0.51	3.96 ± 0.54
SNR $= 30_{dB}$	1.20 ± 0.06	0.75 ± 0.08	0.98 ± 0.09	1.09 ± 0.06	1.64 ± 0.15	2.69 ± 0.32	4.08 ± 0.48
SNR $= 20_{dB}$	1.25 ± 0.06	0.75 ± 0.03	1.03 ± 0.09	1.12 ± 0.08	1.66 ± 0.14	2.68 ± 0.33	4.47 ± 0.50
SNR $= 10_{dB}$	1.34 ± 0.02	0.89 ± 0.09	1.11 ± 0.02	1.21 ± 0.12	1.71 ± 0.15	2.77 ± 0.28	4.55 ± 0.23
SNR $= 5_{dB}$	1.37 ± 0.07	0.97 ± 0.09	1.13 ± 0.11	1.17 ± 0.08	1.70 ± 0.22	2.80 ± 0.21	4.86 ± 0.37
SNR $= 0_{dB}$	1.50 ± 0.05	1.34 ± 0.10	1.21 ± 0.12	1.06 ± 0.12	1.92 ± 0.22	3.39 ± 0.54	5.44 ± 0.71

Moreover, it is worth noticing that the performance of the proposed method in low distances, i.e., up to 6 meters, are similar across all SNRs. However, from that distance and beyond, the error increases rapidly. That may be due to direct sound and early distinct echoes having more energy than the late reverberant sound for the majority of our room scenarios.

In addition, phase-based features, which have been proved to be the most important information in our clean speech analysis, are severely corrupted even by a tiny amount of noise. For example, direct sound and echo patterns, which are highlighted by transients in the clean signal, are smeared in time due to the noise, losing phase coherence across frequencies.

6.3.4 Summary

A novel approach that provides continuous-valued speaker distance estimation from single-channel audio in reverberant rooms has been proposed. The results of our study have demonstrated that the distance estimation can be performed using phase-based features, i.e., the sin&cos of the STFT phase and a CRNN with an average error of 22 cm in a noiseless scenario. However, the presence of even low-energy ambient noise can affect drastically the performance of the proposed method. As for future work, one potential direction is to address how background noise can impact phase-based features. As demonstrated from the results, these features are heavily affected by background noise, impacting the model performance.

6.4 Generalization of Distance Speaker Estimators

This Section describes the key improvement over the results obtained with the CRNN architecture proposed in the previous Section. Importantly, we introduce an attention module designed for identifying the most relevant time-frequency patterns from the input features in the *speaker distance estimation* task. For a fair evaluation of the proposed method, we performed experiments on synthetic data in noiseless and noisy environments. This allowed for good measurement of performance in controlled environments. Additional evaluations were carried out for the CRNN with a designed hybrid dataset that consists of measured RIRs convolved with anechoic speeches and with two real-world recording datasets. In addition, two datasets encompassing real recordings were employed to assess the performance in scenarios where no control of the environment is possible. Finally, a cross-dataset analysis across scenarios, both with and without fine-tuning, demonstrates how the nature of RIR can impact the distance estimation task.

6.4.1 Attention Module and Baseline Revision

First, acoustic features are extracted from the single-channel audio. As done in the previous Section, 3 maps (magnitude of the STFT, sinus, and cosinus of the STFT phase) are obtained with shape $T \times F$, where T and F are the time and frequency bins, respectively. Then, the maps are stacked along the channel dimension, resulting in a feature tensor of size $T \times F \times 3$. To highlight the feature regions that are most informative for distance estimation, an attention map is learned from the three-channel tensor, which is then element-wise multiplied with the input feature tensor.

One of the main contributions of this work is the definition of an attention module that computes an attention map $H \in \mathbb{R}^{+T \times F \times 3}$ from the audio features. The objective of this learned matrix is to emphasize the regions of the features that are most informative for the estimation of the distance. Specifically, this module is the function $\mathrm{f}_{\mathrm{ATT}} : \mathbb{R}^{T \times F \times 3} \rightarrow \mathbb{R}^{+^{T \times F \times 3}}$. Its structure is composed of 2 convolutional blocks, having 16 and 64 3×3 filters, respectively. Then, a 1×1 convolutional layer with three filters, followed by a sigmoid activation, is used to map the features to yield the $T \times F \times 3$ attention map. Finally, the output acoustic features $\tilde{X} \in \mathbb{R}^{T \times F \times 3}$ are obtained by element-wise multiplication ($\otimes$) between the input acoustic features and the attention map as

$$\tilde{X} = f_{\mathrm{ATT}}(X) \otimes X. \tag{6.6}$$

Regarding the convolutional layers, we optimized the architecture to be low-complexity. In more detail, the structure of each block involves a 2D convolutional layer comprising P_i 1×3 filters, i.e., along the frequency axis with values of 8, 32, and 128 assigned to the respective layers, differently from the previous version where all the convolutional layers encompassed 128 filters. We denote these filters as *frequency kernels*, whereas 3×1 filters are named *time kernels*. Square kernels, known for their capability to capture time-frequency patterns, are commonly used in convolutional layers applied to spectrograms due to their effectiveness in capturing local patterns and structures along the frequency axis. In this work, the proposed model adopts rectangular filters, and temporal information is modeled by recurrent layers at the end of the model. Rectangular filters can be more parameter-efficient compared to square kernels. Since the former has fewer parameters than square kernels of the same receptive field size, they can lead to a more compact model, making training and inference more computationally efficient and potentially reducing the risk of overfitting, especially when working with limited data.

Examples of noiseless and noisy spectrograms and attention maps are

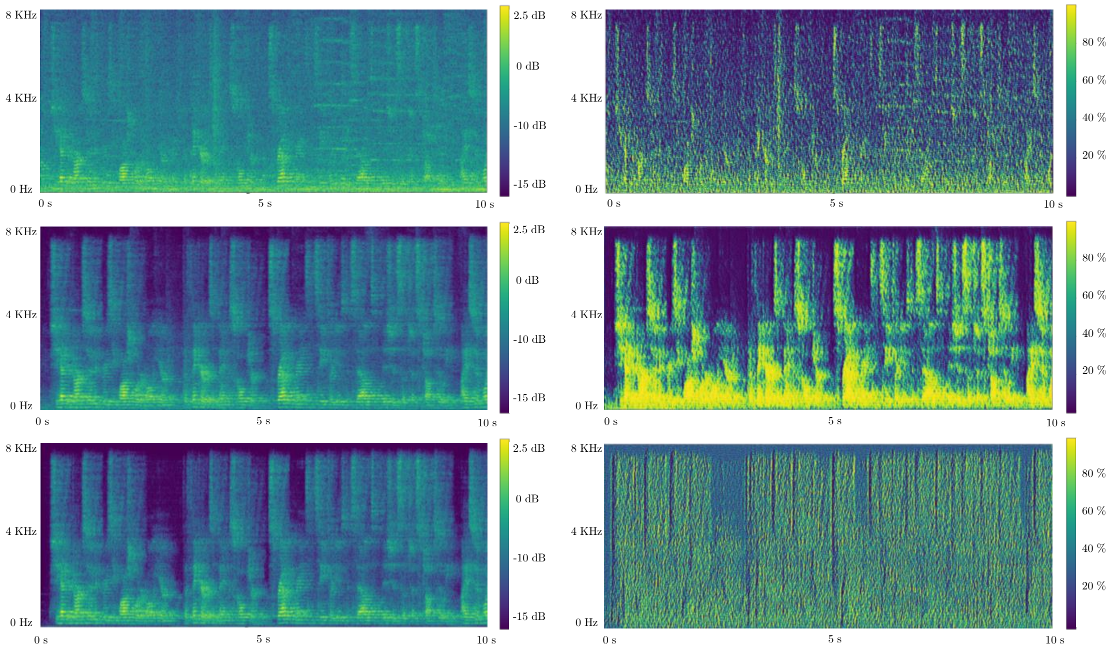


Figure 6.7: Example of spectrograms and attention maps on a speaker talking at 10 meters. First row is at SNR = 0 dB, second row is at SNR = 30 dB, and last row is at SNR = $+\infty$, i.e., noiseless scenario.

depicted in Figure 6.7. It is worth highlighting how the attention module focuses differently on the parts of the signal where the speech is most likely to stand out from the noise or where the characteristics of the speech are still recognizable. The attention map in a noiseless case is evenly distributed across the entire frequency range since no noise interferes.

To process the feature maps from the convolutional layers, two bidirectional GRU layers are utilized with $\tanh(\cdot)$ as the activation function. These layers have exhibited promising results in audio and speech processing tasks, demonstrating parameter efficiency compared to LSTM networks [138].

The output of the CNN with shape $T \times 2 \times P$ is stacked along the channel dimension to produce a $T \times Q$ matrix to be fed to the recurrent layers. Then, in the proposed configuration, the extraction of reverberation-related information primarily relies on integrating information over time with the recurrent layers. Within this implementation, two bi-directional GRUs with $Q = 2P = 128$ neurons each for every time frame are employed.

Then, to predict the distance, three fully connected layers are employed, where an independent mapping between each time frame is performed in each layer. Firstly, the initial linear layer projects time-wise features from the last GRU onto a matrix of dimensions $T \times R$, where $R = 128$. Subsequently, the second linear layer independently maps each time frame of the $T \times R$ matrix onto a vector of size $T \times 1$, denoted as the time-wise distance estimation $\hat{\mathbf{y}}$. Specifically, this vector represents the distance estimation for each time frame. Finally, the last fully connected layer is employed to perform regression and thus estimate the predicted distance, denoted as $\hat{y} \in \mathbb{R}$.

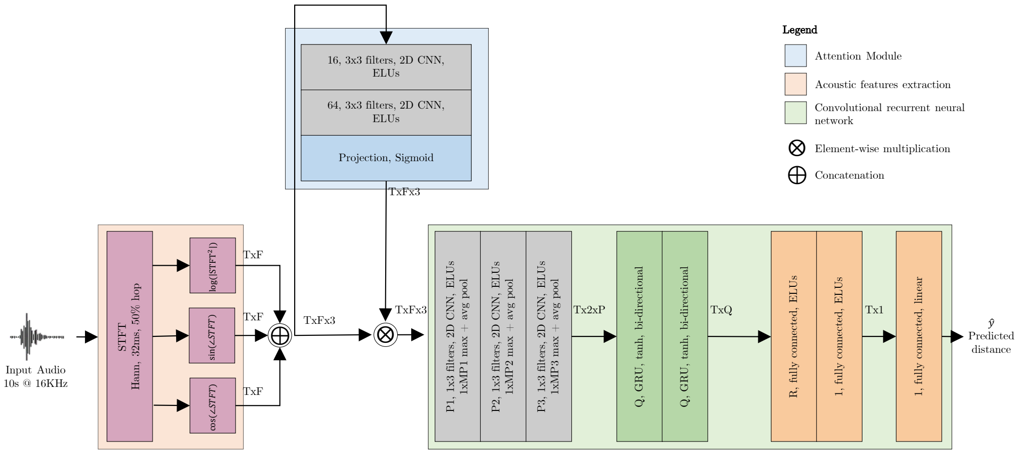

16, 3x3 filters, 2D CNN, ELUs
64, 3x3 filters, 2D CNN, ELUs
Projection, Sigmoid
TxFx3
Legend
Attention Module
Acoustic features extraction
Convolutional recurrent neural network
Element-wise multiplication
Concatenation
Input Audio 10s @ 16KHz
STFT Hann, 32ms, 50% hop
log(|STFT²|)
sin(∠STFT)
cos(∠STFT)
TxF
TxF
TxF
TxFx3
TxFx3
P1, 1x3 filters, 2D CNN, ELUs 1xMP1 max + avg pool
P2, 1x3 filters, 2D CNN, ELUs 1xMP2 max + avg pool
P3, 1x3 filters, 2D CNN, ELUs 1xMP3 max + avg pool
Tx2xP
Q, GRU, tanh, bi-directional
Q, GRU, tanh, bi-directional
TxQ
R, fully connected, ELUs
1, fully connected, ELUs
Tx1
1, fully connected, linear
ŷ Predicted distance

Figure 6.8: Proposed architecture for speaker distance estimation.

The overall revised model can be inspected in Figure 6.8.

The MSE loss is used to train the DNN system. Let $y \in \mathbb{R}$ be the true distance of a static sound source. In addition, let $\mathbf{y} \in \mathbb{R}^{T\times 1}$ be the vector consisting of frame-wise ground truth distances. Then, the loss used in the training phase for a single sample is

$$\mathcal{L}(y, \hat{y}, \mathbf{y}_t, \hat{\mathbf{y}}_t) = (y - \hat{y})^2 + ||\mathbf{y}_t - \hat{\mathbf{y}}_t||^2, \tag{6.7}$$

where the loss is averaged across the batch dimension to be exploited by the backpropagation algorithm. Thanks to the imposition of the loss, the model predicts a distance for each time bin and, from this information, a single-valued distance. Having two losses in a static source scenario operates as a regularization term since it forces the proposed approach to return coherently both time-wise and single-distance estimations. However, in the context of dynamic sound sources, it is important to highlight that only frame-wise loss is required.

6.4.2 Hybrid Dataset

The RIRs used in the hybrid dataset, contained in the C4DM RIR database [145], were measured in three rooms located at Queen Mary, University of London, London, UK. A Genelec 8250A loudspeaker was employed as the source for measuring all IRs, while each receiver position was measured using both an omnidirectional DPA 4006 and a B-format Soundfield SPS422B.

A collection of 130 RIRs was captured in a classroom with dimensions $7.5 \times 9 \times 3.5$ m (236 m^3) and consist of reflective surfaces such as a linoleum floor, painted plaster walls, ceiling, and a sizable whiteboard.

The second room, denoted as the Octagon, is a Victorian structure that was finalized in 1888. Presently serving as a conference venue, the walls of this building still showcase book-lined interiors, complemented by a wooden floor and plaster ceiling. As the name implies, this room features eight walls, each measuring 7.5 m in length, and a domed ceiling towering 21 m above the floor, resulting in an estimated volume of 9500 m^3. In the center of the room, a total of 169 RIRs were measured.

The third room is The Great Hall, which possesses a seating capacity of approximately 800. It encompasses a stage and seating sections both on the floor and on a balcony. To capture the audio, the microphones were positioned within the cleared seating area on the floor, spanning an area of approximately 23×16 m. The microphone placements mirror the layout used for the Octagon, encompassing 169 RIRs over a 12×12 m region.

Following the same setup of the synthetic dataset, anechoic speech recordings are convolved from TIMIT [140], and real background noises from WHAM! [143] are added with the measured RIRs, generating the hybrid QMULTIMIT dataset. For each RIR, 5 random speech recordings are selected from the TIMIT dataset, yielding 2340 audio files. RIRs are randomly divided into training, validation, and testing splits following a percentage ratio of 70-10-20. Finally, the MAE errors averaged across all the distance bins are provided.

6.4.3 Real Datasets

VoiceHome-2 [146]. This dataset is specifically made for distant speech processing applications in domestic environments. It consists of short commands for smart home devices in French, collected in reverberant conditions and uttered by twelve native French speakers facing the microphone. The data is recorded in twelve different rooms corresponding to four houses, with fully annotated geometry, under quiet or noisy conditions. More precisely, VoiceHome-2 includes everyday noise sources (with no annotations regarding their SNRs) such as competing talkers, TV/radio, footsteps, doors, kitchenware, and electrical appliances. Five speaker positions per room, comprising standing and sitting postures, are selected to encompass a broad range of angles and distances concerning the microphone array, which maintains a single, fixed position throughout all the room recordings. The sound is then captured by a microphone array consisting of eight Micro-ElectroMechanical Systems (MEMS) placed near the corner of a cubic baffle. For this study, only the first channel has been extracted. Regarding room acoustics, to obtain the impulse responses, recordings of a 6-second chirp from 0 to 8 kHz were processed. The chirp was played by a loudspeaker in each room, and recordings were performed for two different positions of the microphone array and seven to nine different positions of the loudspeaker. These positions spanned a range of angles and were distributed logarithmically across distance. The recordings were then convolved with the inverse chirp to obtain the estimated room impulse responses. Additionally, in each of the twelve rooms, five complex, everyday noise scenes relevant to the function of the room were recorded. These scenes included background speech, television sounds, footsteps, meal preparation noises, shutters opening or closing, water flowing, and more. The recordings were made at the same two array positions as mentioned earlier. It is important to note that the noise sources varied across the different homes.

In total, VoiceHome-2 encompasses 752 audio recordings, lasting approximately 10 seconds for all the twelve rooms and the five noise scenes. It is important to highlight that the experiments in this work do not involve any information regarding either raw RIR or injected noise, emulating an on-field recording. The dataset is then randomly split using a percentage ratio of 70-10-20 training, validation, and testing splits, respectively, for the experiments.

STARSS22 [147]. The dataset includes recordings of human interaction scenes with spatio-temporal event annotations for thirteen target classes, primarily focusing on speech. It is part of the DCASE Challenge 2022 Task 3 development set. The recordings were made at two sites, Tampere University in Finland and Sony headquarters in Japan, in a total of eleven rooms maintaining a consistent organization and procedure regarding equipment, recording, and annotations. The dataset utilizes the Eigenmike spherical microphone array, offering two spatial formats. One format involves a tetrahedral sub-array of omnidirectional microphones mounted on a rigid spherical baffle. The corpus is more challenging compared to the other datasets due to the natural movement and orientation of multiple speakers during discussions, as well as the presence of intentional and unintentional sound events other than speech. It also contains diffuse and directional ambient noise at significant levels. Finally, audio data from a single microphone of the Eigenmike array has been processed, extracting 2934 two-second single-speech excerpts that do not overlap with other annotated directional sources. As done before with the other datasets, STARSS22 is split using a percentage ratio of 70-10-20 training, validation, and testing splits, respectively.

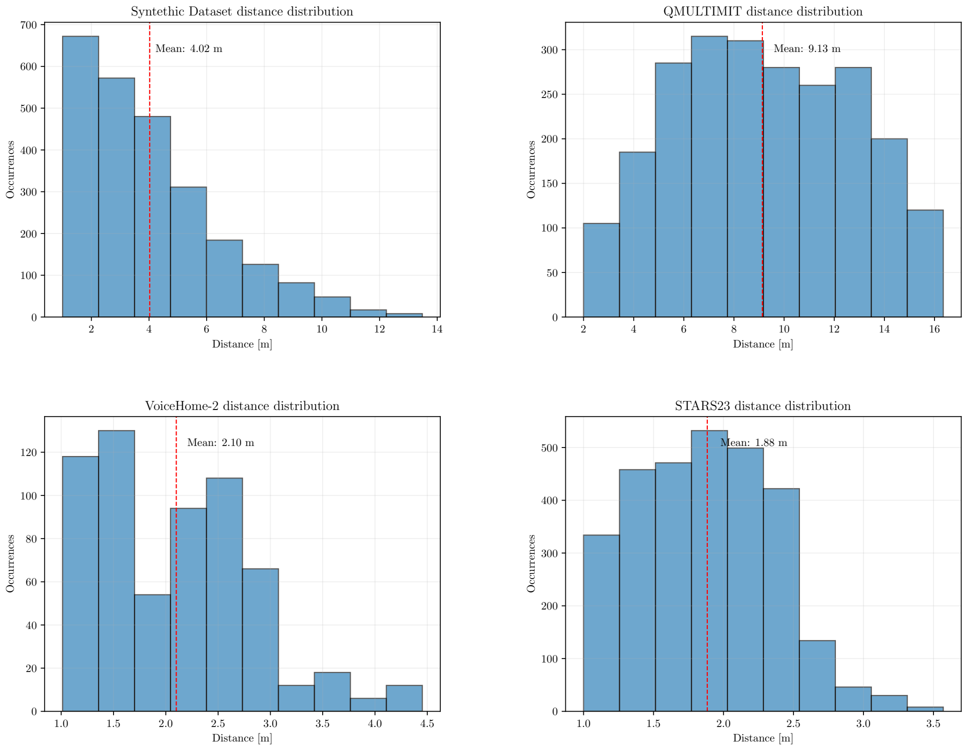


Figure 6.9: Distributions of distances in each dataset.

It is worth noticing that, as can be inspected in Figure 6.9, real dataset distances are differently distributed with respect to the synthetic and hybrid ones. The motivations of this behavior are as follows:

- in many real-world scenarios, as in STARSS23 [148], sound sources are not always at a fixed distance from the recording device;
- different recording environments can introduce variations in the speaker distance distribution. For example, in a controlled studio setting, speakers may be positioned at specific distances from the microphone to achieve the desired sound characteristics. In contrast, field recordings or recordings made in everyday settings can have a wider range of distances due to the uncontrollable nature of the environment. Indeed,

in this context, VoiceHome-2 [146] has been recorded in a domestic environment whereas STARSS23 [147] has been collected in office-like environments;

- audio datasets are often curated to suit specific applications or scenarios. For instance, a dataset focused on speaker recognition in far-field scenarios may deliberately include more examples with distant speakers to simulate real-world challenges. On the other hand, a dataset for speech enhancement in close-proximity situations may prioritize examples with close speaker distances. VoiceHome - 2 has been curately designed for enhancing distant-microphone speech, whereas STARSS23 focuses on SELD, yielding dissimilar distance distributions.

Accordingly, with the distributions of distances in real scenarios, the distance bins used are $\{[1, 2), [2, 2.5), [2.5, 3), [3, 3.5), [3.5, 4), [4, 4.5)\}$ meters. The final MAE errors are averaged using a percentage ratio of 70-10-20 training, validation, and testing splits, respectively.

6.4.4 Experiments

This Section describes how the performance assessment of the proposed approach has been carried out. To validate the work, three levels of realism have been addressed in the scope of speaker distance estimation:

- **Synthetic**: simulated RIRs of an image-source room simulator are convolved with anechoic speech;
- **Hybrid**: measured RIRs are convolved with anechoic speech;
- **Real**: on-field reverberant speech recordings.

Figure 6.9 depicts the histograms of distances in each dataset employed in the experimental results.

Clean Speech

The proposed approach efficiently estimates speaker distance with an average error of 11 cm in a noiseless scenario, as can be inspected from Table 6.7. Since there is no other published method that attempts regression-based SDE with a single microphone, for comparison purposes, results on binaural SDE are presented following the recently published work of [100]. The binaural estimation model is similar to the CRNN model used herein. A similar simulator, range of acoustic conditions, and number of rooms was used in [100] as herein. The same spectrogram and binaural features are also used as in the original work. The binaural estimation results (86 cm) we obtain are, on average, better than the ones in [100] (151 cm), with the improvement most likely attributed to the use of the attention layers. However, the most striking difference is that of the monophonic omnidirectional results (11 cm) versus the binaural ones (86 cm). It seems that the complex frequency, direction, and orientation-dependent effects imposed by Head-Related Transfer Functions (HRTFs) make it harder for the model to associate spectrotemporal reverberation patterns with the source distance. However, a definite conclusion on differences between single-channel omnidirectional versus binaural SDE requires further study.

An increasing trend of the errors with respect to the distance is notable. This behavior is expected due to the dominant influence of the late reverberant component compared to the direct and early reflection components of the signal at long distances. These late reverberation cues exhibit statistical diffusion [144], meaning that short-term magnitudes and phases resemble noise-like characteristics. Consequently, extracting meaningful information from these dominant late reverberation cues may pose challenges for the model in effectively estimating speaker distance.

Such behaviour is demonstrated in Figure 6.10. Considering that the balance between direct speech energy versus early and late reverberant energy is exemplified in the DRR, measured from the simulated RIRs, it is clear that dominance of the reverberation at low DRRs impacts negatively distance estimation. There seems to be an optimum balance where both direct sound and reverberation contribute to estimation, after which direct sound can start to mask reverberation-related cues for higher DRRs, with a subsequent small drop in performance. A closer investigation of distance estimation at very high DRRs or very small distances at the near-field of the microphone is left for future work.

Table 6.7: Hyperparameters selection on the synthetic dataset with clean speech. The gray row highlights the proposed approach.

Kernels	# params	# GRUs	Average		[1, 2)		[2, 4)		[4, 8)		[8, 14)	
			$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$
Binaural [100]	650 k	2	0.86 ± 0.10	0.29 ± 0.05	1.06 ± 0.35	0.72 ± 0.22	0.70 ± 0.13	0.25 ± 0.05	0.81 ± 0.10	0.15 ± 0.02	1.34 ± 0.61	0.13 ± 0.05
Time	123 k	0	0.55 ± 0.02	0.18 ± 0.01	0.50 ± 0.04	0.35 ± 0.03	0.50 ± 0.03	0.18 ± 0.01	0.57 ± 0.03	0.11 ± 0.01	0.79 ± 0.09	0.08 ± 0.01
Squared	149 k	0	0.70 ± 0.02	0.23 ± 0.01	0.59 ± 0.04	0.42 ± 0.03	0.68 ± 0.04	0.24 ± 0.01	0.71 ± 0.04	0.13 ± 0.01	1.03 ± 0.12	0.11 ± 0.01
Frequency	123 k	0	0.86 ± 0.03	0.30 ± 0.01	0.83 ± 0.05	0.60 ± 0.04	0.80 ± 0.04	0.28 ± 0.02	0.86 ± 0.04	0.16 ± 0.01	1.17 ± 0.14	0.12 ± 0.01
Time	353 k	1	0.16 ± 0.01	0.05 ± 0.00	0.15 ± 0.01	0.11 ± 0.01	0.13 ± 0.01	0.05 ± 0.00	0.19 ± 0.01	0.03 ± 0.00	0.27 ± 0.03	0.03 ± 0.00
Squared	379 k	1	0.15 ± 0.01	0.05 ± 0.00	0.13 ± 0.01	0.09 ± 0.01	0.11 ± 0.01	0.04 ± 0.00	0.16 ± 0.01	0.03 ± 0.00	0.27 ± 0.04	0.03 ± 0.00
Frequency	353 k	1	0.13 ± 0.01	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.08 \pm 0.01}$	0.10 ± 0.01	0.04 ± 0.00	0.13 ± 0.01	$\mathbf{0.02 \pm 0.00}$	0.24 ± 0.04	$\mathbf{0.02 \pm 0.00}$
Time	650 k	2	0.13 ± 0.01	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.12 \pm 0.01}$	0.09 ± 0.01	0.10 ± 0.01	0.04 ± 0.00	0.13 ± 0.01	$\mathbf{0.02 \pm 0.00}$	0.24 ± 0.07	$\mathbf{0.02 \pm 0.01}$
Squared	676 k	2	$\mathbf{0.11 \pm 0.00}$	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.08 \pm 0.01}$	$\mathbf{0.09 \pm 0.00}$	$\mathbf{0.03 \pm 0.00}$	0.12 ± 0.01	$\mathbf{0.02 \pm 0.00}$	0.18 ± 0.03	$\mathbf{0.02 \pm 0.00}$
Frequency	650 k	2	$\mathbf{0.11 \pm 0.00}$	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.08 \pm 0.01}$	0.10 ± 0.00	$\mathbf{0.03 \pm 0.00}$	$\mathbf{0.11 \pm 0.01}$	$\mathbf{0.02 \pm 0.00}$	$\mathbf{0.16 \pm 0.02}$	$\mathbf{0.02 \pm 0.00}$

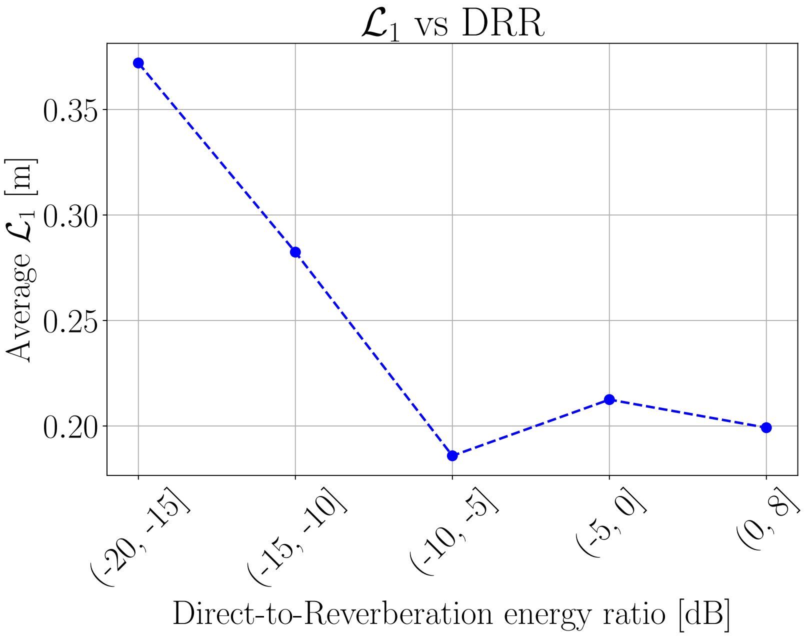


Figure 6.10: Relation between DRR and $\mathcal{L}_1$.

Moreover, the results of the study demonstrate that the GRU layers play a crucial role in the model's performance. The GRU layers likely contribute to the model's ability to capture sequential patterns and dependencies effectively. Additionally, the study found that using rectangular kernels, as opposed to square kernels, in combination with GRU layers improves the model's efficiency. In this scenario, the rectangular kernels are better at capturing different types of patterns and features in the data, leading to more effective and efficient information processing within the model. This statement, however, does not hold when no GRU layers are present.

In addition, it is worth noting that using a single GRU layer slightly impacts the overall performance of the proposed approach, approximately halving the number of learnable parameters.

Noisy Speech

To assess the quality of the predictions about noise strength, seven SNR values have been specifically chosen during training. More precisely, a separate model is trained from scratch for each SNR level.

Table 6.8 depicts the results where a notable discrepancy between the noiseless and noisy scenarios becomes evident. This divergence is primarily attributed to the disruptive influence of background noise on the phase information [135], which has also been demonstrated in speech enhancement

Table 6.8: Experimental results on noisy synthetic data with fixed SNR and frequency kernels. The gray row highlights the proposed approach.

SNR	Feature set	$\mathcal{L}_1$	r$\mathcal{L}_1$
50 dB	w/|STFT|	0.48 ± 0.02	0.14 ± 0.01
	w/sinus and cosinus	$\mathbf{0.37 \pm 0.02}$	$\mathbf{0.11 \pm 0.01}$
	|STFT| + sinus and cosinus	0.41 ± 0.02	0.12 ± 0.00
40 dB	w/|STFT|	0.77 ± 0.03	$\mathbf{0.21 \pm 0.01}$
	w/sinus and cosinus	$\mathbf{0.71 \pm 0.03}$	$\mathbf{0.21 \pm 0.01}$
	|STFT| + sinus and cosinus	0.87 ± 0.04	0.24 ± 0.01
30 dB	w/|STFT|	$\mathbf{1.11 \pm 0.04}$	$\mathbf{0.30 \pm 0.01}$
	w/sinus and cosinus	1.51 ± 0.06	0.45 ± 0.02
	|STFT| + sinus and cosinus	1.14 ± 0.04	0.31 ± 0.01
20 dB	w/|STFT|	$\mathbf{1.20 \pm 0.04}$	$\mathbf{0.33 \pm 0.01}$
	w/sinus and cosinus	1.76 ± 0.06	0.56 ± 0.02
	|STFT| + sinus and cosinus	1.21 ± 0.05	$\mathbf{0.33 \pm 0.01}$
10 dB	w/|STFT|	1.30 ± 0.05	0.36 ± 0.01
	w/sinus and cosinus	1.70 ± 0.06	0.56 ± 0.02
	|STFT| + sinus and cosinus	$\mathbf{1.27 \pm 0.05}$	$\mathbf{0.35 \pm 0.01}$
5 dB	w/|STFT|	1.34 ± 0.05	0.38 ± 0.01
	w/sinus and cosinus	1.73 ± 0.06	0.58 ± 0.02
	|STFT| + sinus and cosinus	$\mathbf{1.26 \pm 0.05}$	$\mathbf{0.34 \pm 0.01}$
0 dB	w/|STFT|	1.47 ± 0.05	0.44 ± 0.02
	w/sinus and cosinus	1.77 ± 0.06	0.61 ± 0.02
	|STFT| + sinus and cosinus	$\mathbf{1.39 \pm 0.05}$	$\mathbf{0.42 \pm 0.02}$

studies [149].

It is worth noting from Figure 6.11 that the performance of the proposed method remains consistent across all SNR levels for distances up to 6 meters. However, beyond this distance, the error increases rapidly. This behavior can be attributed to the quadratic inverse relationship between distance and sound intensity, i.e., $I_s \propto \frac{1}{d^2}$. Due to this physical behavior, the direct sound and early distinct echoes exhibit similar energy levels compared to the late reverberant cues, hindering long-distance information.

Hybrid Speech

As done with the synthetic dataset, five SNR values have been selected to assess the performance of the proposed architecture by training a separate

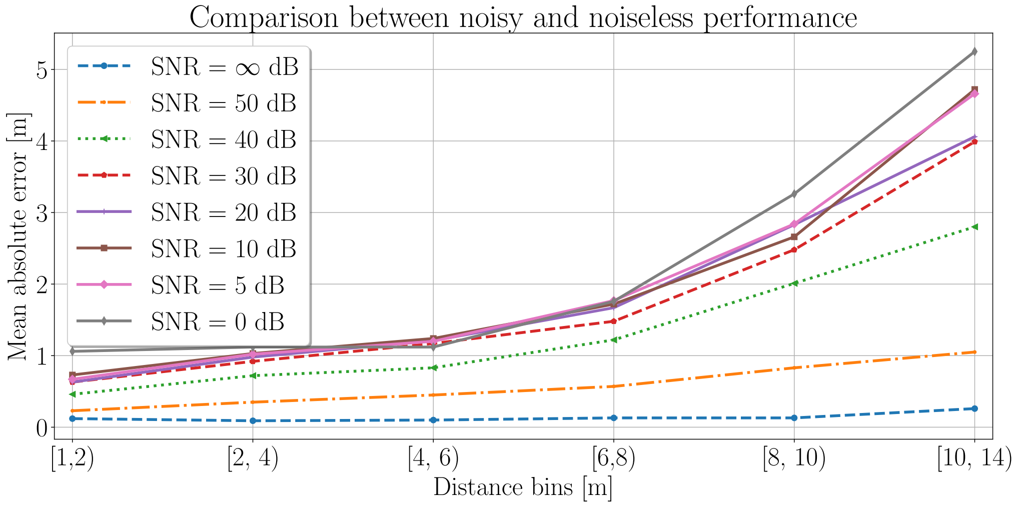


Figure 6.11: Comparison between noisy and noiseless performance of the proposed approach on the synthetic dataset.

model from scratch for each SNR level. Table 6.9 shows the experimental results, highlighting the superiority of the chosen configuration.

The notation $[30, +\infty)$ dB denotes the results of the model both in noiseless case and with at most 30 dB of SNR. It is worth noting that, differently from the synthetic scenario, the impact of background noise is smaller even at low SNR. Comparing Table 6.8 with Table 6.9, it is evident how synthetic RIRs are more affected by noise at higher SNR with respect to measured ones.

Interestingly, the use of only sinus and cosinus maps yields poor performance at all SNRs levels, whereas the STFT magnitude is essential for the task. This result agrees with the previous study [135] where the use of only sinus and cosinus features in noisy audio recordings is ineffective.

Real speech

Table 6.10 and Table 6.11 depict the results on VoiceHome - 2 [146] and STARSS23 [147], respectively.

Table 6.9: Distance estimation errors for the QMULTIMIT hybrid dataset. Gray row highlights the proposed approach. All features are used if not mentioned

SNR	Hyperparameters	# GRUs	$\mathcal{L}_1$	r$\mathcal{L}_1$
[30, +∞) dB	*Time*	0	2.49 ± 0.16	0.28 ± 0.02
	Squared	0	2.38 ± 0.15	0.25 ± 0.02
	Frequency	0	2.97 ± 0.17	0.33 ± 0.03
	Time	1	1.58 ± 0.12	0.16 ± 0.01
	Squared	1	1.52 ± 0.12	0.15 ± 0.01
	Frequency	1	1.68 ± 0.12	0.17 ± 0.01
	Time	2	1.70 ± 0.12	0.17 ± 0.01
	Squared	2	**1.48 ± 0.13**	**0.14 ± 0.01**
	Freq. w/|STFT|	2	1.67 ± 0.13	0.17 ± 0.01
	Freq. w/sinus and cosinus	2	2.17 ± 0.14	0.23 ± 0.02
	Frequency	2	1.52 ± 0.12	0.15 ± 0.01
20 dB	*Time*	0	2.22 ± 0.15	0.24 ± 0.02
	Squared	0	2.36 ± 0.15	0.25 ± 0.02
	Frequency	0	2.88 ± 0.17	0.32 ± 0.02
	Time	1	1.67 ± 0.12	0.16 ± 0.01
	Squared	1	**1.46 ± 0.12**	**0.14 ± 0.01**
	Frequency	1	1.71 ± 0.12	0.17 ± 0.01
	Time	2	1.66 ± 0.13	0.16 ± 0.01
	Squared	2	1.60 ± 0.13	0.16 ± 0.01
	Freq. w/|STFT|	2	1.64 ± 0.13	0.16 ± 0.01
	Freq. w/sinus and cosinus	2	1.98 ± 0.13	0.21 ± 0.02
	Frequency	2	1.48 ± 0.11	**0.14 ± 0.01**
10 dB	*Time*	0	2.23 ± 0.14	0.24 ± 0.02
	Squared	0	2.20 ± 0.14	0.24 ± 0.02
	Frequency	0	2.55 ± 0.14	0.28 ± 0.02
	Time	1	1.71 ± 0.12	0.17 ± 0.01
	Squared	1	1.58 ± 0.13	0.16 ± 0.01
	Frequency	1	1.60 ± 0.12	0.16 ± 0.01
	Time	2	1.65 ± 0.12	0.16 ± 0.01
	Squared	2	1.56 ± 0.13	**0.15 ± 0.01**
	Freq. w/|STFT|	2	**1.55 ± 0.12**	**0.15 ± 0.01**
	Freq. w/sinus and cosinus	2	1.97 ± 0.12	0.21 ± 0.01
	Frequency	2	1.65 ± 0.13	0.17 ± 0.01
0 dB	*Time*	0	2.54 ± 0.14	0.28 ± 0.02
	Squared	0	2.74 ± 0.15	0.30 ± 0.02
	Frequency	0	3.01 ± 0.15	0.33 ± 0.02
	Time	1	1.75 ± 0.12	0.18 ± 0.01
	Squared	1	1.83 ± 0.12	0.19 ± 0.01
	Frequency	1	1.82 ± 0.13	0.19 ± 0.01
	Time	2	2.46 ± 0.15	0.23 ± 0.01
	Squared	2	1.98 ± 0.12	0.21 ± 0.02
	Freq. w/|STFT|	2	**1.63 ± 0.13**	**0.17 ± 0.01**
	Freq. w/sinus and cosinus	2	2.24 ± 0.13	0.25 ± 0.02
	Frequency	2	1.66 ± 0.13	**0.17 ± 0.01**
−10 dB	*Time*	0	3.03 ± 0.14	0.34 ± 0.03
	Squared	0	3.03 ± 0.14	0.33 ± 0.02
	Frequency	0	3.04 ± 0.14	0.33 ± 0.02
	Time	1	3.02 ± 0.14	0.33 ± 0.02
	Squared	1	3.01 ± 0.14	0.33 ± 0.03
	Frequency	1	3.00 ± 0.14	0.33 ± 0.03
	Time	2	3.06 ± 0.14	0.34 ± 0.03
	Squared	2	2.57 ± 0.13	0.28 ± 0.02
	Freq. w/|STFT|	2	**2.28 ± 0.13**	**0.25 ± 0.02**
	Freq. w/sinus and cosinus	2	3.01 ± 0.14	0.33 ± 0.03
	Frequency	2	2.34 ± 0.13	**0.25 ± 0.02**

Table 6.10: Distance estimation errors for the VoiceHome - 2 dataset. The gray row highlights the proposed approach. All features are used if not mentioned

Hyperparameters	# GRUs	Average		$[1, 2)$		$[2, 3)$		$[3, 4.5)$	
		$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$
Time	0	0.95 ± 0.10	0.49 ± 0.06	1.00 ± 0.14	0.73 ± 0.11	0.69 ± 0.12	0.28 ± 0.05	1.20 ± 0.25	0.32 ± 0.06
Squared	0	0.90 ± 0.11	0.46 ± 0.07	0.90 ± 0.16	0.69 ± 0.14	0.57 ± 0.11	0.23 ± 0.04	1.34 ± 0.26	0.35 ± 0.07
Frequency	0	0.83 ± 0.09	0.43 ± 0.06	0.85 ± 0.13	0.63 ± 0.11	0.67 ± 0.13	0.27 ± 0.05	1.02 ± 0.20	0.27 ± 0.05
Time	1	0.76 ± 0.09	0.38 ± 0.05	0.73 ± 0.12	0.55 ± 0.10	0.47 ± 0.10	0.19 ± 0.04	1.19 ± 0.23	0.32 ± 0.06
Squared	1	0.74 ± 0.09	0.40 ± 0.07	0.85 ± 0.15	0.65 ± 0.13	0.43 ± 0.09	$\mathbf{0.17 \pm 0.04}$	0.96 ± 0.20	0.26 ± 0.05
Frequency	1	0.74 ± 0.08	0.37 ± 0.05	0.73 ± 0.12	0.54 ± 0.10	0.53 ± 0.10	0.21 ± 0.04	1.06 ± 0.21	0.28 ± 0.05
Time	2	0.64 ± 0.08	$\mathbf{0.31 \pm 0.05}$	$\mathbf{0.59 \pm 0.12}$	$\mathbf{0.44 \pm 0.10}$	0.49 ± 0.09	0.20 ± 0.03	0.94 ± 0.21	0.25 ± 0.05
Squared	2	0.70 ± 0.10	0.35 ± 0.06	0.67 ± 0.14	0.51 ± 0.12	$\mathbf{0.43 \pm 0.12}$	$\mathbf{0.17 \pm 0.05}$	1.11 ± 0.21	0.29 ± 0.05
Freq w /|STFT|	2	0.66 ± 0.08	0.33 ± 0.05	0.63 ± 0.13	0.48 ± 0.11	0.47 ± 0.10	0.19 ± 0.04	0.98 ± 0.17	0.27 ± 0.05
Freq w /sinus and cosinus	2	0.91 ± 0.11	0.46 ± 0.07	0.88 ± 0.14	0.68 ± 0.13	0.52 ± 0.11	0.21 ± 0.04	1.49 ± 0.21	0.40 ± 0.05
Frequency	2	$\mathbf{0.63 \pm 0.08}$	0.32 ± 0.05	0.64 ± 0.11	0.48 ± 0.10	0.48 ± 0.11	0.19 ± 0.04	$\mathbf{0.80 \pm 0.20}$	$\mathbf{0.21 \pm 0.05}$

Following the same rationale of the synthetic and hybrid scenarios, the selected configuration outperforms the other models. The results obtained from the analysis of real data demonstrate the clear superiority of the proposed model in accurately estimating distances. Across both datasets, the proposed model consistently outperforms different configurations of the models, showcasing its robustness and effectiveness. However, it is worth noting that a few outliers surfaced in the results, particularly within the VoiceHome - 2 dataset, where large confidence intervals are present. This occurrence can be attributed to the limited size of the datasets as the model overfits the training dataset. With a larger dataset, these outliers are expected to be mitigated, and the model's performance is likely to become even more reliable and precise. This observation underscores the potential for further advancement in distance estimation when working with more extensive datasets.

Table 6.11: Distance estimation errors for the STARSS23 dataset. The gray row highlights the proposed approach. All features are used if not mentioned

Hyperparameters	# GRUs	Average		[1, 2)		[2, 2.5)		[2.5, 3)	
		$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$
Time	0	0.51 ± 0.03	0.23 ± 0.01	0.30 ± 0.04	0.16 ± 0.02	0.55 ± 0.03	0.24 ± 0.01	0.76 ± 0.10	0.29 ± 0.04
Square	0	0.50 ± 0.03	0.22 ± 0.01	0.29 ± 0.04	0.16 ± 0.02	0.53 ± 0.03	0.23 ± 0.01	0.85 ± 0.09	0.33 ± 0.03
Frequency	0	0.51 ± 0.03	0.23 ± 0.01	0.35 ± 0.05	0.19 ± 0.03	0.54 ± 0.03	0.24 ± 0.01	0.76 ± 0.10	0.29 ± 0.04
Time	1	0.45 ± 0.02	0.20 ± 0.01	0.26 ± 0.03	$\mathbf{0.14 \pm 0.02}$	0.49 ± 0.03	0.21 ± 0.01	0.70 ± 0.08	0.27 ± 0.03
Square	1	$\mathbf{0.42 \pm 0.02}$	$\mathbf{0.19 \pm 0.01}$	0.33 ± 0.04	0.18 ± 0.02	$\mathbf{0.42 \pm 0.03}$	$\mathbf{0.18 \pm 0.01}$	0.62 ± 0.09	0.24 ± 0.03
Frequency	1	0.46 ± 0.02	0.20 ± 0.01	0.30 ± 0.04	0.16 ± 0.02	0.48 ± 0.03	0.21 ± 0.01	0.69 ± 0.08	0.26 ± 0.03
Time	2	0.46 ± 0.02	0.21 ± 0.01	$\mathbf{0.27 \pm 0.03}$	0.15 ± 0.02	0.49 ± 0.03	0.22 ± 0.01	0.69 ± 0.09	0.26 ± 0.03
Square	2	0.50 ± 0.02	0.22 ± 0.01	0.34 ± 0.04	0.19 ± 0.02	0.51 ± 0.03	0.23 ± 0.01	0.79 ± 0.09	0.30 ± 0.03
Freq w /|STFT|	2	0.46 ± 0.02	0.21 ± 0.01	0.28 ± 0.03	0.15 ± 0.02	0.49 ± 0.03	0.21 ± 0.01	0.71 ± 0.09	0.27 ± 0.03
Freq w /sinus and cosinus	2	0.46 ± 0.02	0.20 ± 0.01	0.28 ± 0.03	0.16 ± 0.02	0.48 ± 0.03	0.21 ± 0.01	0.74 ± 0.09	0.28 ± 0.03
Frequency	2	$\mathbf{0.42 \pm 0.02}$	$\mathbf{0.19 \pm 0.01}$	0.33 ± 0.05	0.18 ± 0.03	0.43 ± 0.03	0.19 ± 0.01	$\mathbf{0.55 \pm 0.09}$	$\mathbf{0.21 \pm 0.04}$

Table 6.12: Cross-dataset generalization tests without finetuning.

		Test w/o finetuning		
		Synthetic	**Hybrid**	**Real**
Training	**Synthetic**	0.11 ± 0.00	4.28 ± 0.45	4.14 ± 0.08
	Hybrid	6.80 ± 0.59	1.52 ± 0.12	3.76 ± 0.56
	Real	2.26 ± 0.38	8.22 ± 0.54	0.42 ± 0.02

Table 6.13: Cross-dataset generalization tests with finetuning.

		Test w/ finetuning		
		Synthetic	**Hybrid**	**Real**
Training	**Synthetic**	0.11 ± 0.00	1.57 ± 0.23	0.47 ± 0.05
	Hybrid	0.18 ± 0.04	1.52 ± 0.12	0.45 ± 0.05
	Real	0.11 ± 0.02	1.54 ± 0.22	0.42 ± 0.02

Tests have been carried out in a cross-corpus training-testing setup, e.g., synthetic-hybrid, synthetic-real, hybrid-real, VoiceHome-STARSS. The model yields very large errors in case no finetuning is performed, as it can be inspected in Table 6.12. This behavior highlights the discrepancy of feature patterns among different acoustic scenarios, levels of acoustical realism, and different distance distributions. If the model is fine-tuned to a different realistic scenario, the performance is slightly worse that the case when the model starts with random weights. The results of this situation is shown in Table 6.13.

To demonstrate the effectiveness of the attention module, an ablation study is performed on all the scenarios. First, performance assessment is carried out without the module. Then, instead of returning a $T \times F \times 3$ matrix, a spectrogram attention map, i.e., $T \times F$, is learned by a module. Then, an element-wise multiplication is performed between the magnitude of the STFT and the attention map.

These three modalities are analyzed in Table 6.14, depicting the errors for each bin with their confidence intervals. Predicting an attention map for each feature provides better distance estimation on average. Moreover, the results demonstrate that all the approaches perform similarly in the short range, up to 8 meters. Conversely, applying the attention map on each of the feature maps in the feature set produces better outcomes in the long range with respect to the other two cases. When the speaker is far from the microphone, the learned attention maps enhance the features set, facilitating the extraction of features of the convolutional layers. Indeed, as the distance between the speaker and the microphone increases, detecting these patterns becomes more challenging due to their reduced salience [144].

Moreover, an ablation study has been carried out also on the hybrid and real data, as it can be inspected in Table 6.15. The attention map yields the best performance in the hybrid case when it is only applied to the STFT magnitude channel. This fact highlights the ineffectiveness of phase features in this specific use case. Instead, the results demonstrate the superiority of the attention map applied to all the channels in the real scenario.

Table 6.14: Ablation study of attention map using frequency kernels on synthetic data with clean speech. The gray row highlights the proposed approach.

Attention	Average		[1, 2)		[2, 4)		[4, 8)		[8, 14)	
	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$
None	0.14 ± 0.01	0.05 ± 0.00	0.13 ± 0.01	0.09 ± 0.01	0.12 ± 0.01	0.04 ± 0.00	0.15 ± 0.01	0.03 ± 0.00	0.28 ± 0.05	0.03 ± 0.00
on spectrogram	0.12 ± 0.00	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.08 \pm 0.01}$	$\mathbf{0.10 \pm 0.01}$	0.04 ± 0.00	0.13 ± 0.01	$\mathbf{0.02 \pm 0.00}$	0.22 ± 0.03	$\mathbf{0.02 \pm 0.00}$
on everything	$\mathbf{0.11 \pm 0.00}$	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.08 \pm 0.01}$	$\mathbf{0.10 \pm 0.00}$	$\mathbf{0.03 \pm 0.00}$	$\mathbf{0.11 \pm 0.01}$	$\mathbf{0.02 \pm 0.00}$	$\mathbf{0.16 \pm 0.02}$	$\mathbf{0.02 \pm 0.00}$

Table 6.15: Ablation study of attention map using frequency kernels on hybrid and real data. Gray row highlights the proposed approach.

Attention	QMULTIMIT		VoiceHome - 2		STARSS22	
	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$	$\mathcal{L}_1$	r$\mathcal{L}_1$
None	2.01 ± 0.06	0.21 ± 0.01	0.78 ± 0.09	0.40 ± 0.06	0.45 ± 0.02	0.20 ± 0.01
on spectrogram	$\mathbf{1.87 \pm 0.06}$	$\mathbf{0.19 \pm 0.01}$	0.73 ± 0.10	0.36 ± 0.06	0.45 ± 0.02	0.20 ± 0.01
on everything	1.90 ± 0.06	0.20 ± 0.01	$\mathbf{0.63 \pm 0.08}$	$\mathbf{0.32 \pm 0.05}$	$\mathbf{0.42 \pm 0.02}$	$\mathbf{0.19 \pm 0.01}$

6.4.5 Discussion

From the results of the noisy scenario in the synthetic dataset, it is important to highlight that even a minimal amount of noise severely corrupts phase-based features, which have been identified as the most critical information in our analysis of clean speech. For instance, the presence of direct sound and echo patterns, characterized by transients in the clean signal, becomes blurred over time due to the presence of noise and late reverberation, resulting in a loss of phase coherence across frequencies. This behavior, however, does not occur in the hybrid dataset where the effect of high SNR in the recordings does not correspond to a similar increase in estimation performance. That may be due to the recordings of the RIRs having a level of inherent measurement noise, which limits the effective SNR that we can achieve in the hybrid simulations.

The imposition of the loss in Equation (6.7) is required for predicting a time-wise distance vector. Due to the lack of baselines and datasets in the literature, only a single value of distance of the sound source is assigned for each time bin to ease the distance tracking task. Generally, this characteristic in audio datasets is referred to as *weak labels* [150]. Without time-wise distance references, denoted as *strong labels*, the model faces challenges in fine-tuning its predictions, decreasing its overall performance. This scenario has been studied in the literature for tasks that require a fine temporal resolution output, such as SED [44] and SELD [151].

Furthermore, it is important to acknowledge that certain portions of the audio data encompass segments where speech information is absent or indiscernible. Consequently, this scarcity of informative speech content can considerably undermine the effectiveness and reliability of predictors.

In this direction, the proposed attention module can improve the ability of the model (Tab. 6.15) to identify the speech information that is relevant for the estimation of the distance. However, it is important to note that the attention module is learned by the model itself, without any direct supervision.

To address these limitations, a potential avenue for improvement emerges, centering around the generation of more comprehensive and fine-grained labels. By augmenting the dataset with *strong labels* that introduces both speech activity and speaker distance estimation, the model may acquire a better understanding of the room acoustics. In addition, this augmentation enables the model to leverage additional contextual cues and refine its predictions, enhancing its performance in accurately estimating speaker distances and capturing the dynamics of speech activity.

Moreover, one of the key areas for improvement is the availability of larger datasets of real recordings with a greater number of rooms and various speaker-microphone configurations. A larger dataset would enable the model to learn more diverse and representative acoustic characteristics, leading to improved performance in distance estimation tasks. Moreover, it could also improve the generalization ability of the approach, as it has been demonstrated how the performance of the proposed model is dependent on the nature of the audio recording (synthetic, hybrid or real). Additionally, by including different room types and microphone placements, the model can better generalize across various real-world scenarios. Furthermore, the use of a transformer-based [152] approach could be explored, leveraging a larger amount of data. Transformer models have shown remarkable success in various natural language processing tasks and have the potential to capture complex patterns and dependencies in acoustic data. Exploiting transformer architectures could enhance the model's ability to estimate distances accurately.

Another possibility for future research is the integration of time-wise distance ground truth, as previously mentioned in the discussion section. By considering temporal information in addition to spatial cues, the model could potentially estimate the distance of a sound source more accurately. This would provide valuable insights in scenarios where multiple sound sources are present. Estimating and tracking the distance of a moving source is an application of interest that is scarcely explored in the literature.

6.4.6 Summary

This work has explored the task of speaker distance estimation in noisy and reverberant environments. Multiple configurations, in terms of kernel size and recurrent layers of the model, have been provided, motivating the proposed architecture. The use of rectangular filters across the frequency dimension and the presence of GRUs layers yields the best performance in terms of distance errors. The experimental results obtained from the proposed model have demonstrated remarkable precision in scenarios where several types of RIRs are employed. In a noiseless synthetic scenario where RIRs have been generated with a room-source simulator, the model has achieved an absolute error of only 0.11 meters. With recorded RIRs, an absolute error of about 1.30 meters has been obtained. In the real scenario with on-field recordings, where unpredictable environmental factors and noise were prevalent, the model yielded an absolute error of approximately 0.50 meters. These results underscore the model's resilience and its capacity to effectively manage various realistic scenarios. Variations in performance across these scenarios can be attributed to differences in the distribution of acoustic parameters, such as the distance from the sound source. Analysis on moving sound sources in single-channel recordings will be carried out as a future work.

6.5 Conclusion

The content of this Chapter is associated with the following publications:

- **Michael Neri**, A. Ferrarotti, L. de Luisa, A. Salimbeni, and M. Carli, "ParalMGC: Multiple Audio Representations for Synthetic Human Speech Attribution", in: *10th European Workshop on Visual Information Processing (EUVIP)*, 2022 [64].

- **Michael Neri**, A. Politis, D. Krause, M. Carli, and T. Virtanen, "Single-Channel Speaker Distance Estimation in Reverberant Environments", in: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023 [135].

- **Michael Neri**, A. Politis, D. Krause, M. Carli, and T. Virtanen, "Speaker Distance Estimation in Enclosures from Single-Channel Audio", in: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024 [51].

Chapter 7

Conclusions and Future Perspectives

7.1 Introduction

This Thesis was structured around the exploration of AI techniques applied to sound-based scene understanding. Each one of the Chapters presented important improvements in the development of low-complexity models that can face real acoustic challenges. The Thesis began with an introduction to the motivation and objectives of the research (Chapter 1). Specifically, it highlighted the main challenges involved in the design of computationally efficient AI models, which are suitable for real-time processing in resource-constrained environments.

Then, the background chapter outlined the bases upon which the research had been conducted (Chapter 2). It discussed basic principles related to how audio signals are typically represented in the state-of-the-art. These basic principles included the fields of room acoustics and the realms of time-frequency transformations. Moreover, this Chapter introduced the main AI methodologies -machine learning, deep learning, and neural networks- to be used in sound event recognition. At the same time, this Chapter helps the reader theoretically to be prepared for the models and techniques that will be further carried out in this Thesis.

Chapter 3 addressed the problem of ASC, proposing a novel model using Chebyshev polynomials and moments for feature extraction. This method encompasses a good balance between performance and computation efficiency and, thus, is suitable to be deployed in an environment where real-time processing is crucial.

Chapter 4 changed the focus to UASD and proposed an attention-based model to improve the detection of anomalies from infrequent and subtle variations in sound. Then, Chapter 5 delved into SED in noisy environments, which is critical for real-world applications such as public safety monitoring. It proposed a low-complexity SED model with Atrous Spatial Pyramid Pooling, which is used to enhance feature extraction. An entirely new dataset called SEDDOB, which simulates real-world noisy environments in public transportation, was introduced.

Chapter 6 presented the challenge of estimating the distance of the speakers using single-channel audio signals. In addition, a framework that combines CRNNs with attention mechanisms for estimating speaker distances in reverberant and noisy environments was introduced. The robustness of this model was evidenced through experiments on synthetic, hybrid, and real datasets that proved its usability in real-time applications such as virtual assistants, smart devices, and other security or surveillance systems.

Finally, this Chapter summarizes this Thesis by reviewing the contributions and presenting the practical use of the developed models. Many of the limitations found while carrying out this research are discussed. Some suggestions about future work involve the extension of these models towards more complicated settings and integrating multimodal data to get a better scene understanding. This Chapter emphasizes the value of the research in further developing the field of sound-based scene understanding and serving as a prologue for the following discussion.

7.2 Contributions to Sound Event Recognition Field

The contributions of this Thesis span both theoretical advancements and practical applications in sound-based scene understanding using AI techniques:

- **Efficient ASC models.** The use of Chebyshev moments provided an innovative approach to feature extraction, allowing the development of models that achieve good performance while maintaining low computational demands, making them suitable for deployment in mobile and edge computing environments.

- **Tackling the domain shift of ASC.** A new semi-supervised approach has been devised by introducing an iterative FT procedure. This contribution improves the robustness of sound-based scene understanding systems, making them more adaptable to diverse and unseen environments.

- **Enhanced Unsupervised Anomaly Detection.** Anomaly detection was advanced through attention-based models that effectively focused on relevant patterns, improving the detection of rare and subtle anomalies in complex acoustic environments. An initial effort was made to explain data-driven UASD models.

- **Robust and efficient SED Models for Noisy Environments.** By optimizing feature extraction using ASPP, the developed SED models achieved greater accuracy in detecting sound events in noisy environments, especially in public safety monitoring scenarios. The introduction of the SEDDOB dataset also serves as a valuable resource for future research on the detection of sound events in public transport settings.

- **Advanced Speaker Distance Estimation.** It has been demonstrated that estimating the distance between the speaker and an omnidirectional microphone is possible by exploiting the phase of the STFT. In this direction, a robust approach to speaker distance estimation was developed, using a combination of CRNNs and attention mechanisms. In addition, the model demonstrated weak generalization across different datasets and acoustic conditions, proving that further study is required in this direction from the research community.

7.3 Practical Applications

The models and methodologies developed in this Thesis have practical implications across a range of industries and applications as follows:

- **Mobile and IoT Devices.** The low-complexity models for ASC and SED are well-suited for integration into resource-constrained devices, such as smartphones, wearable devices, and edge computing platforms. Their real-time processing capabilities make them ideal for applications like personal assistant technologies, autonomous vehicles, and smart city infrastructure.

- **Public Safety and Surveillance.** The UASD and SED models developed in this Thesis have significant applications in public safety and security systems. By detecting anomalies or safety-critical events like gunshots or altercations in noisy environments, these models can enhance real-time surveillance and monitoring in public spaces, transportation hubs, and large events.

- **Voice-based Authentication systems.** Together with the identification of anomalous sound events, it is fundamental to identify whether a speech is real or not. Speech can be used to authorize a user to access restricted or protected data. By introducing an attention mechanism, thus improving the explainability, voice-based authentication systems can achieve higher accuracy and reliability, ensuring secure and seamless user verification in a variety of conditions.
- **Smart Home Systems.** Estimating the distance of the speaker has applications in smart home environments, where understanding the distance of a speaker can improve interaction quality in voice-controlled systems and virtual assistants. Its robustness in handling reverberation and noise makes it suitable for various home and office settings.
- **Industrial Monitoring.** The unsupervised anomaly detection model is particularly relevant for industrial applications, where real-time monitoring of machinery and equipment is crucial. By detecting subtle anomalies without requiring labeled data, the model enables the early identification of potential malfunctions, reducing maintenance costs and operational downtime.

7.4 Limitations and Future Perspectives

Although this Thesis highlighted many contributions, there are still several open problems to be solved in the context of scene understanding. Although the models performed well on the tested datasets, generalizing them to more diverse real-world environments presents a challenge. For example, the ASC model may require further fine-tuning or domain adaptation techniques [153] when applied to new acoustic environments not represented in the datasets used. Future work could focus on gathering more diverse real-world datasets to further improve model generalization. These datasets could cover various room configurations, different microphone placements, and a broader range of acoustic environments.

The speaker distance estimation model exhibited increasing errors beyond 6 meters, indicating the need for further refinement when estimating distances over long ranges. Additional features, such as information on the geometry of the room, could help mitigate this limitation. As already stated in Chapter 6, given such limitations, one probable line of improvement comes in the form of generating better and finer labels. By providing the model with *strong labels* that add speech activity and speaker distance estimation to the dataset, it could learn the room acoustics better. Other advantages of this augmentation are that the model can exploit additional contextual clues and fine-tune its predictions, hence giving better performance in the estimation of speaker distances with greater precision and also the dynamics of speech activity. In this direction, extending the speaker distance estimation framework to track moving speakers in real time would be a valuable addition. Incorporating dynamic tracking algorithms or motion data could enable continuous updates to speaker position estimates, enhancing applications in smart environments and surveillance.

Future work could focus on improving the temporal precision of the SED model without significantly increasing computational demand. This could involve optimizing feature extraction techniques or designing hybrid models that balance spatial and temporal resolution.

In conclusion, while this Thesis primarily explored CNNs and RNNs, future research could explore transformer-based architectures, which have shown promise in capturing long-range dependencies in audio sequences. Transformers [22] could enhance the anomaly detection and speaker distance estimation tasks by better modeling temporal dynamics. In addition, integrating audio with other sensory modalities, such as video or text, could improve scene understanding in complex environments. Multimodal models could provide a more holistic view, especially in situations where audio alone is insufficient to interpret a scene accurately.

Acknowledgements

Appendix A

Other Academic Contributions

A.1 Editorial Contributions

I am a Managing Editor for *Signal Processing: Image Communications* Elsevier journal from September 2024.

I have been a reviewer for several journals and conferences.

- **Journals.** IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, Expert Systems with Application, Signal Processing: Image Communications, IEEE Access.

- **Conferences.** IEEE International Conference on Multimedia & Expo (ICME), IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP), IEEE International Workshop on Multimedia Signal Processing (MMSP), Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE), IEEE International Workshop on Information Forensics and Security (WIFS), International Symposium on Image and Signal Processing and Analysis (ISPA).

In 2023 I was Local Arrangement Co-Chair for ISPA and a Session Chair (*Visual Data Acquisition and Computation* Session) for IEEE ICME 2024.

A.2 Teaching and Student Supervision

During my Ph.D., I delivered several guest lectures on specific topics in various courses, providing expertise in information security and audio processing. The courses were as follows:

- *Multimedia Laboratory* from B.Sc. in Electronic Engineering, Roma Tre University;

- *Ethical Hacking* from M.Sc. in Communication and Information Technology Engineering, Roma Tre University;
- *Telecommunication Systems* from B.Sc. in Electronic Engineering, Roma Tre University;
- *Internet & Multimedia* from B.Sc. in Electronic Engineering, Roma Tre University.

Moreover, I co-supervised the following undergraduate students:

- Marco Mirabella (Roma Tre University, M.Sc. in Communication and Information Technology Engineering) with the thesis *Binary Anomaly Detection in Polyphonic Audio Signals*, 2022.
- Martino Buongiorno (Roma Tre University, M.Sc. in Communication and Information Technology Engineering) with the thesis *Classificazione di segnali audio mediante l'utilizzo di algoritmi di Machine Learning*, 2022.
- Nicolò Scialpi (University of Padua, M.Sc. in ICT for Internet and Multimedia) with the thesis *Traffic anomaly detection based on 2D representation with computer vision and machine learning techniques*, 2023.
- Mirko Mannari (University of Padua, B.Sc. in Information Engineering) with the thesis *Rilevamento di anomalie del traffico di rete basato su tecniche di elaborazione delle immagini*, 2023.
- Sofia Vitale (Roma Tre University, M.Sc. in Communication and Information Technology Engineering) with the thesis *Unsupervised Anomaly Detection on Audio Signals*, 2024.
- Asma Mirkhan (University of Padua, M.Sc. in ICT for Internet and Multimedia) with the thesis *Enhancing Point Cloud Quality Assessment with Grouped Convolutions: A Streamlined Approach Inspired by COPP-Net*, 2024.

A.3 Dissemination Activities

To translate knowledge and university research into the work world, several "Third Mission" activities has been carried out during the Ph.D. period:

- *Notte della ricerca & Maker Faire 2022*: an automatic emotion recognition system ("I know that feel, bro!") based on the analysis of body language through Artificial Intelligence was presented. In addition, part of Chapter 5 regarding the SED model was presented.

- *Visiting Tampere University*: visited Tampere University from January to April 2023 under the supervision of Prof. Virtanen to work on the estimation of the speaker distance from single-channel audio recordings;

- *Maker Faire 2023*: "VisionArt VR: What are you looking at?" a Virtual Reality application for assessing the quality of immersive multimedia where the user is free to move.

- *Notte della ricerca 2023 & Roma Tre Open Night 2023*: an automatic emotion recognition system ("I know that feel, bro!") based on the analysis of body language through AI was presented.

- *Maker Faire 2024*: "VisionArt VR: What are you looking at?" a Virtual Reality application for assessing the quality of immersive multimedia where the user is free to move.

Appendix B

List of Publications by the Author

The following articles[1] have been published during the Ph.D. period:

Journal Articles

- **Michael Neri***, F. Battisti, A. Neri, and M. Carli. “Sound Event Detection for Human Safety and Security in Noisy Environments”, in: *IEEE Access*, 2022.
- K. Lamichhane, **Michael Neri**, P. Pradip, F. Battisti, and M. Carli, “No-Reference Light Field Image Quality Assessment Exploiting Saliency”, in: *IEEE Transactions on Broadcasting*, 2023.
- **Michael Neri***, A. Politis, D. Krause, M. Carli, and T. Virtanen. “Speaker Distance Estimation in Enclosures from Single-Channel Audio”, in: *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.
- **Michael Neri***, “Anomaly Detection and Classification of Audio Signals with Artificial Intelligence Techniques”, in: *Science Talks*, 2024.
- **Michael Neri*** and M. Carli. “Low-complexity Unsupervised Audio Anomaly Detection exploiting Separable Convolutions and Angular Loss”, in: *IEEE Sensors Letters*, 2024.
- M. Bernabei, S. Colabianchi, M. Carli, F. Costantino, A. Ferrarotti, **Michael Neri**, S. Stabile, “Enhancing occupational safety and health training: a guideline for virtual reality integration”, in: *IEEE Access*, 2024.

[1] An asterisk is present when a conference paper has been presented by me or a journal article has been published with me as a corresponding author.

- **Michael Neri***, and F. Battisti, “Low-Complexity Patch-based No-Reference Point Cloud Quality Metric exploiting Weighted Structure and Texture Features”, in: *IEEE Transactions on Broadcasting*, 2025.

Conference papers

- L. Pallotta, **Michael Neri**, M. Buongiorno, A. Neri, and G. Giunta, “A Machine Learning-Based Approach for Audio Signals Classification using Chebychev Moments and Mel-Coefficients”, in: *International Conference on Frontiers of Signal Processing (ICFSP)*, 2022.

- S. Baldoni, F. Battisti, M. Brizzi, **Michael Neri***, and Neri. A., “A Semantic Segmentation-based Approach for Train Positioning”, in: *ITM/PTTI Institute Of Navigation (ION)*, 2022.

- **Michael Neri***, A. Ferrarotti, L. de Luisa, A. Salimbeni, and M. Carli, “ParalMGC: Multiple Audio Representations for Synthetic Human Speech Attribution”, in: *10th European Workshop on Visual Information Processing (EUVIP)*, 2022.

- **Michael Neri*** and F. Battisti, “3D Object Detection on Synthetic Point Clouds for Railway Applications”, in: *10th European Workshop on Visual Information Processing (EUVIP)*, 2022.

- **Michael Neri***, L. Pallotta, and M. Carli, “Low-Complexity Environmental Sound Classification using Cadence Frequency Diagram and Chebychev Moments”, in: *International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2023.

- **Michael Neri*** and M. Carli, “Artificial Intelligence Techniques for Quality Assessments of Immersive Multimedia”, in: *ACM International Conference on Interactive Media Experiences (IMX)*, 2023.

- **Michael Neri***, A. Politis, D. Krause, M. Carli, and T. Virtanen, “Single-Channel Speaker Distance Estimation in Reverberant Environments”, in: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023.

- **Michael Neri*** and M. Carli, “Semi-Supervised Acoustic Scene Classification under Domain Shift using Attention-based Separable Convolutions and Angular Loss”, in: *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2024.

- R. Bentivenga, M. Bernabei, M. Carli, S. Colabianchi, F. Costantino, A. Ferrarotti, **Michael Neri**, E. Pietrafesa, E. Sorrentino, S. Stabile, "Advancing Occupational Safety and Health training: a Safety-II integration of the ADDIE model for virtual reality", in: *Methodologies and Intelligent Systems for Technology Enhanced Learning, 14th International Conference*, 2024.

- R. Bentivenga, M. Bernabei, M. Carli, S. Colabianchi, F. Costantino, A. Ferrarotti, **Michael Neri**, E. Pietrafesa, E. Sorrentino, S. Stabile, "Transforming Training With New Enabling Technologies: A Proposal To Verify The Efficacy Of Virtual Reality Tools In The Occupational Health And Safety Sector", in: *8th World Conference on Smart Trends in systems, Security, and Sustainability (Worlds4)*, 2024.

Appendix C

Acronyms

AI Artificial Intelligence

AE Autoencoder

ASC Acoustic Scene Classification

ASD Anomalous Sound Detection

ASPP Atrous Spatial Pyramid Pooling

AuSPP Audio Spatial Pyramid Pooling

AUC Area Under the Curve

AVM Automatic Vehicle Monitoring

BSMD-STD Binaural Signal Magnitude Difference Standard Deviation

CAS Chinese Acoustic Scene

CFD Cadence Frequency Diagram

CNN Convolutional Neural Network

COTS Commercial-off-the-shelf

CRNN Convolutional Recurrent Neural Network

DCASE Detection and Classification of Acoustic Scenes and Events

DCT Discrete Cosine Transform

DFT Discrete Fourier Transform

DNN Deep Neural Network

DoA Direction of Arrival

DoAE Direction of Arrival Estimation

DRR Direct-to-Reverberant energy Ratio

ELU Exponential Linear Unit

ER Error Rate

ERB Equivalent Rectangular Bandwidth

ERM Empirical Risk Minimization

FCN Fully Convolutional Network

FFT Fast Fourier Transform

FIR Finite Impulse Response

FPR False Positive Rate

FT Fine-Tuning

GAN Generative Adversarial Network

GAP Global Average Pooling

GDPR General Data Protection Regulation

GMM Gaussian Mixture Model

GRU Gated Recurrent Unit

GTCC GammaTone Cepstral Coefficient

HAS Human Auditory System

IID Interchannel Intensity Difference

ILD Interchannel Level Difference

IM Inlier Modeling

IR Information Retrieval

ITD Interchannel Time Difference

KNN K-Nearest Neighbours

LDA Linear Discriminative Analysis

LPC Linear Predictive Coding

LSTM Long Short-Term Memory

LTI Linear Time-Invariant

MAE Mean Absolute Error

MAP Maximum A Posteriori Probability

MEMS Micro-ElectroMechanical Systems

MFCC Mel-Frequency Cepstrum Coefficients

ML Machine Learning

MLP MultiLayer Perceptron

MSE Mean Squared Error

MSC Magnitude Squared Coherence

PDE Partial Differential Equation

pAUC partial Area Under the Curve

RF Random Forest

RIR Room Impulse Response

RNN Recurrent Neural Network

ReLU Rectified Linear Unit

ROC Receiving Operating Curve

SAM Spectrogram-aware Attention Module

SDE Source Distance Estimation

SED Sound Event Detection

SEDDOB Sound Event Detection Dataset On Bus

SELD Sound Event Localization and Detection

SER Sound Event Recognition

SNR Signal-to-Noise Ratio

SOTA State of the Art

STFT Short-Time Fourier Transform

SVM Support Vector Machine

TPR True Positive Rate

UAS Urban Acoustic Scene

UASD Unsupervised Anomalous Sound Detection

VC Vapnik-Chervonenkis

References

[1] H. Kuttruff, *Room acoustics*, Crc Press, 2016.

[2] V. Valimaki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty Years of Artificial Reverberation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1421–1448, 2012.

[3] H. Wallach, E. B. Newman, and M. R. Rosenzweig, "A precedence effect in sound localization," *The Journal of the Acoustical Society of America*, vol. 21, no. 4_Supplement, pp. 468–468, 1949.

[4] M. Barron, "The subjective effects of first reflections in concert halls—the need for lateral reflections," *Journal of sound and vibration*, vol. 15, no. 4, pp. 475–494, 1971.

[5] D. Griesinger, "The psychoacoustics of apparent source width, spaciousness and envelopment in performance spaces," *Acta Acustica united with Acustica*, vol. 83, no. 4, pp. 721–731, 1997.

[6] E. Zwicker, "Subdivision of the audible frequency range into critical bands (Frequenzgruppen)," *The Journal of the Acoustical Society of America*, vol. 33, no. 2, pp. 248–248, 1961.

[7] X. Valero and F. Alias, "Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1684–1689, 2012.

[8] V. N. Vapnik and A. Ya. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability & Its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

[9] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," in *ACM International Conference on Multimedia*, 2014.

[10] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *ACM International Conference on Multimedia*, 2015.

[11] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018.

[12] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017.

[13] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An Open Dataset of Human-Labeled Sound Events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022.

[14] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath, "Deep Learning for Audio Signal Processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[15] I. Goodfellow, *Deep learning*, vol. 196, MIT press, 2016.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*. pmlr, 2015.

[17] C. Djork-Arné, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *International Conference on Learning Representations (ICLR)*, 2016.

[18] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[19] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[20] S. Adavanne, A. Politis, and T. Virtanen, "Localization, Detection and Tracking of Multiple Moving Sound Sources with a Convolutional

Recurrent Neural Network," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[21] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2019.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] Y. Gong, Y. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Interspeech*, 2021.

[24] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *IEEE/CVFConference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] S. Dixon, "On the Computer Recognition of Solo Piano Music," in *Arts & Cultural Management Conference*, 2000.

[26] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for Polyphonic Sound Event Detection," *Applied Sciences*, vol. 6, no. 6, 2016.

[27] C. J. Van Rijsbergen, *Information Retrieval*, Butterworth-Heinemann, Newton, MA, USA, 1979.

[28] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic Scene Classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[29] Y. Tan, H. Ai, S. Li, and M. D. Plumbley, "Acoustic Scene Classification Across Cities and Devices via Feature Disentanglement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1286–1297, 2024.

[30] Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," *arXiv preprint arXiv:2005.14623*, 2020.

[31] M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[32] J. Salamon, C. Jacoby, and J. P. Bello, "A Dataset and Taxonomy for Urban Sound Research," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[33] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015.

[34] S. P. Priyal and P. K. Bora, "A Study on Static Hand Gesture Recognition using Moments," in *International Conference on Signal Processing and Communications (SPCOM)*, 2010.

[35] L. Pallotta, M. Cauli, C. Clemente, F. Fioranelli, G. Giunta, and A. Farina, "Classification of micro-Doppler Radar Hand-Gesture Signatures by means of Chebyshev Moments," in *IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2021.

[36] R. Mukundan, S. H. Ong, and P. A. Lee, "Image Analysis by Tchebichef Moments," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1357–1364, 2001.

[37] R. Diaz and E. Pariguan, "On Hypergeometric Functions and Pochhammer k-Symbol," *arXiv preprint math/0405596*, 2004.

[38] A. Ghaleb, L. Vignaud, and J. M. Nicolas, "Micro-Doppler Analysis of Wheels and Pedestrians in ISAR Imaging," *IET Signal Processing*, vol. 2, no. 3, 2008.

[39] S. Björklund, T. Johansson, and H. Petersson, "Evaluation of a micro-Doppler Classification Method on mm-Wave Data," in *IEEE Radar Conference*. IEEE, 2012.

[40] L. Pallotta, M. Neri, M. Buongiorno, A. Neri, and G. Giunta, "A Machine Learning-Based Approach for Audio Signals Classification using Chebychev Moments and Mel-Coefficients," in *7th International Conference on Frontiers of Signal Processing (ICFSP)*, 2022.

[41] C. Clemente, L. Pallotta, A. De Maio, J. J. Soraghan, and A. Farina, "A Novel Algorithm for Radar Classification based on Doppler Characteristics Exploiting Orthogonal Pseudo-Zernike Polynomials," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 417–430, 2015.

[42] P. Lopez-Meyer, J. A. del Hoyo Ontiveros, H. Lu, and G. Stemmer, "Efficient End-to-End Audio Embeddings Generation for Audio Classification on Target Applications," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[43] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[44] M. Neri, F. Battisti, A. Neri, and M. Carli, "Sound Event Detection for Human Safety and Security in Noisy Environments," *IEEE Access*, vol. 10, pp. 134230–134240, 2022.

[45] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A Survey on Vision Transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023.

[46] A. I. Mezza, E. A. P. Habets, M. Müller, and A. Sarti, "Unsupervised Domain Adaptation for Acoustic Scene Classification Using Band-Wise Statistics Matching," in *EUSIPCO*, 2021.

[47] D. Yang, H. Wang, and Y. Zou, "Unsupervised multi-target domain adaptation for acoustic scene classification," *arXiv preprint arXiv:2105.10340*, 2021.

[48] Y. Liu, J. Guan, Q. Zhu, and W. Wang, "Anomalous Sound Detection Using Spectral-Temporal Information Fusion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[49] H. Chen, L. Ran, X. Sun, and C. Cai, "SW-WAVENET: Learning Representation from Spectrogram and Wavegram Using Wavenet for Anomalous Sound Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[50] S. Choi and J. Choi, "Noisy-Arcmix: Additive Noisy Angular Margin Loss Combined With Mixup For Anomalous Sound Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[51] M. Neri, A. Politis, D. A. Krause, M. Carli, and T. Virtanen, "Speaker Distance Estimation in Enclosures From Single-Channel Audio," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2242–2254, 2024.

[52] J. Bai, M. Wang, H. Liu, H. Yin, Y. Jia, S. Huang, Y. Du, D. Zhang, D. Shi, W. Gan, M. D. Plumbley, S. Rahardja, B. Xiang, and J. Chen, "Description on IEEE ICME 2024 Grand Challenge: Semi-supervised Acoustic Scene Classification under Domain Shift," *arXiv:2402.02694*, 2024.

[53] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[54] S. Chen, Y. Liu, X. Gao, and Z. Han, "MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices," in *CCBR*, 2018.

[55] M. Neri, L. Pallotta, and M. Carli, "Low-complexity environmental sound classification using cadence frequency diagram and chebychev moments," in *2023 International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2023, pp. 1–6.

[56] M. Neri and M. Carli, "Semi-Supervised Acoustic Scene Classification under Domain Shift using Attention-based Separable Convolutions and Angular Loss," in *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2024.

[57] M. Neri, "Anomaly Detection and Classification of Audio Signals with Artificial Intelligence Techniques," *Science Talks*, 2024.

[58] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection for Domain Generalization Task," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2022.

[59] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[60] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and Discussion on DCASE 2022 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2022.

[61] T. Dissanayake, T. Fernando, S. Denman, S. Sridharan, H. Ghaemmaghami, and C. Fookes, "A Robust Interpretable Deep Learning Classifier for Heart Anomaly Detection Without Segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 6, pp. 2162–2171, 2021.

[62] M. Neri, F. Battisti, A. Neri, and M. Carli, "Sound Event Detection for Human Safety and Security in Noisy Environments," *IEEE Access*, vol. 10, pp. 134230–134240, 2022.

[63] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *IEEE AVSS*, 2007.

[64] M. Neri, A. Ferrarotti, L. D. Luisa, A. Salimbeni, and M. Carli, "Paralmgc: Multiple audio representations for synthetic human speech attribution," in *2022 10th European Workshop on Visual Information Processing (EUVIP)*, 2022.

[65] K. Wilkinghoff and F. Kurth, "Why Do Angular Margin Losses Work Well for Semi-Supervised Anomalous Sound Detection?," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 608–622, 2024.

[66] J. Wu, F. Yang, and W. Hu, "Unsupervised anomalous sound detection for industrial monitoring based on ArcFace classifier and gaussian mixture model," *Applied Acoustics*, vol. 203, pp. 109188, 2023.

[67] J. Guan, F. Xiao, Y. Liu, Q. Zhu, and W. Wang, "Anomalous Sound Detection Using Audio Representation with Machine ID Based Contrastive Learning Pretraining," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[68] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised Detection of Anomalous Sound Based on Deep Learning and the Neyman–Pearson Lemma," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 212–224, 2019.

[69] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous Sound Detection Based on Interpolation Deep Neural Network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[70] k. Dohi, T. Endo, H. Purohit, R. Tanabe, and Y. Kawaguchi, "Flow-Based Self-Supervised Density Estimation for Anomalous Sound Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[71] G. Bovenzi, G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé, "Network anomaly detection methods in IoT environments via deep learning: A Fair comparison of performance and robustness," *Computers & Security*, vol. 128, pp. 103167, 2023.

[72] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020.

[73] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and Discussion on DCASE 2021 Challenge Task 2: Unsupervised Anomalous Detection for Machine Condition Monitoring Under Domain Shifted Conditions," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021, pp. 186–190.

[74] J. Guan, F. Xiao, Y. Liu, Q. Zhu, and W. Wang, "Anomalous Sound Detection Using Audio Representation with Machine ID Based Contrastive Learning Pretraining," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[75] M. Prabhu, K. M, *Window functions and their applications in signal processing*, Taylor & Francis, 2014.

[76] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *International Conference on Learning Representations*, 2018.

[77] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and Discussion on DCASE2020 Challenge Task2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring," in *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020.

[78] T. K. Chan and Cheng Siong Chin, "A Comprehensive Review of Polyphonic Sound Event Detection," *IEEE Access*, vol. 8, pp. 103339–103373, 2020.

[79] J. F. Gemmeke, "AudioSet: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017.

[80] L.-C. Chen, G. Papandreou, T. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[81] D.J. Symes, "Automatic vehicle monitoring: A tool for vehicle fleet operations," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 2, pp. 235–237, 1980.

[82] N. Kanopoulos, N. Vasanthavada, and R.L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[83] X. Wang, "Laplacian Operator-Based Edge Detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 886–890, 2007.

[84] C. Djork-Arne, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *International Conference on Learning Representations*, 2016.

[85] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, 2015.

[86] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[87] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, 2017.

[88] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks," in *International Conference on Learning Representations*, 2015.

[89] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Computer Vision and Pattern Recognition*, 2016.

[90] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Computer Vision and Pattern Recognition*, 2017.

[91] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Computer Vision and Pattern Recognition*, 2018.

[92] M. Yiwere and E. J. Rhee, "Distance Estimation and Localization of Sound Sources in Reverberant Conditions using Deep Neural Networks," in *International Journal of Applied Engineering Research*, 2017.

[93] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, vol. 12, 2016.

[94] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," in *ICLR*, 2018.

[95] Z. Almutairi and H. Elgibreen, "A review of modern audio deepfake detection methods: challenges and future directions," *Algorithms*, vol. 15, no. 5, pp. 155, 2022.

[96] M. Wölfel and J. W. McDonough, *Distant speech recognition*, Wiley, 2009.

[97] M. Bekrani, A. W. H. Khong, and M. Lotfizad, "A Linear Neural Network-Based Approach to Stereophonic Acoustic Echo Cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1743–1753, 2011.

[98] E. Berglund and J. Sitte, "Sound source localisation through active audition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 653–658.

[99] T. Rodemann, "A study on distance estimation in binaural sound localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[100] D. A. Krause, G. García-Barrios, A. Politis, and A. Mesaros, "Binaural sound source distance estimation and localization for a moving listener," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 996–1011, 2024.

[101] A. Brendel and W. Kellermann, "Distance estimation of acoustic sources using the coherent-to-diffuse power ratio based on distributed training," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2018.

[102] Y.C. Lu and M. Cooke, "Binaural Estimation of Sound Source Distance via the Direct-to-Reverberant Energy Ratio for Static and Moving Sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1793–1805, 2010.

[103] S. Vesa, "Sound Source Distance Learning Based on Binaural Signals," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2007.

[104] S. Vesa, "Binaural Sound Source Distance Learning in Rooms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1498–1507, 2009.

[105] E. Georganti, T. May, S. van de Par, and J. Mourjopoulos, "Sound Source Distance Estimation in Rooms based on Statistical Properties of Binaural Signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1727–1741, 2013.

[106] M. Yiwere and E. J. Rhee, "Sound source distance estimation using deep learning: An image classification approach," *Sensors*, vol. 20, no. 1, pp. 172, 2019.

[107] A. Sobhdel, R. Razavi-Far, and S. Shahrivari, "Few-Shot Sound Source Distance Estimation Using Relation Networks," *arXiv:2109.10561*, 2021.

[108] J. H. DiBiase, *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*, Brown University Providence, RI, 2000.

[109] D. A. Krause, A. Politis, and A. Mesaros, "Joint direction and proximity classification of overlapping sound events from binaural audio," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2021.

[110] K. Patterson, K. Wilson, S. Wisdom, and J. R. Hershey, "Distance-Based Sound Separation," in *Interspeech*, 2022.

[111] E. Georganti, T. May, S. van de Par, A. Harma, and J. Mourjopoulos, "Speaker distance detection using a single microphone," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1949–1961, 2011.

[112] R. Venkatesan and A.B. Ganesh, "Analysis of monaural and binaural statistical properties for the estimation of distance of a target speaker," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 3626–3651, 2020.

[113] B. T. Balamurali, K. E. Lin, S. Lui, J. M. Chen, and D. Herremans, "Toward Robust Audio Spoofing Detection: A Detailed Comparison of Traditional and Learned Features," *IEEE Access*, vol. 7, pp. 84229–84241, 2019.

[114] A. K. Singh and P. Singh, "Detection of AI-Synthesized Speech Using Cepstral & Bispectral Statistics," in *IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2021, pp. 412–417.

[115] R.L.M.A.P.C. Wijethunga, D.M.K. Matheesha, A. A. Noman, K.H.V.T.A. De Silva, M. Tissera, and L. Rupasinghe, "Deepfake Audio Detection: A Deep Learning Based Solution for Group Conversations," in *2nd International Conference on Advancements in Computing (ICAC)*, 2020, vol. 1.

[116] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of Audio Deepfake Detection," in *Odyssey*, 2020.

[117] H. Dhamyal, A. Ali, I. A. Qazi, and A. A. Raza, "Using Self Attention DNNs to Discover Phonemic Features for Audio Deep Fake Detection," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 1178–1184.

[118] C: Borrelli, P. Bestagini, F. Antonacci, A. Sarti, and S. Tubaro, "Synthetic speech detection through short-term and long-term prediction traces," *EURASIP Journal on Information Security*, vol. 2021, pp. 1–14, 2021.

[119] J. M. Martín-Doñas and A. Álvarez, "The Vicomtech Audio Deepfake Detection System Based on Wav2vec2 for the 2022 ADD Challenge,"

in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 9241–9245.

[120] S.B. Dhonde, A.A. Chaudhari, and M.P. Gajare, "Performance evaluation of Mel and bark scale based features for text-independent speaker identification," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 3734–3738, 2019.

[121] S. Ewert, M. Muller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 1869–1872.

[122] R. N. D'souza, P. Huang, and F. Yeh, "Structural analysis and optimization of convolutional neural networks with a small sample size," *Scientific reports*, vol. 10, no. 1, pp. 834, 2020.

[123] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, pp. 611–629, 2018.

[124] G. Maguolo, M. Paci, L. Nanni, and L. Bonan, "Audiogmenter: a MATLAB toolbox for audio data augmentation," *Applied Computing and Informatics*, 2021.

[125] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010.

[126] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch, and K. A. Lee, "ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2507–2522, 2023.

[127] J. K. Nielsen, "Loudspeaker and listening position estimation using smart speakers," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[128] J. Gontmacher, A. Yarhi, P. Havkin, D. Michri, and E. Fisher, "Dsp-based audio processing for controlling a mobile robot using a spherical microphone array," in *IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, 2012.

[129] D. Gabriel, R. Kojima, K. Hoshiba, K. Itoyama, K. Nishida, and K. Nakadai, "2D sound source position estimation using microphone arrays and its application to a VR-based bird song analysis system," *Advanced Robotics*, vol. 33, no. 7-8, pp. 403–414, 2019.

[130] J. Hwang, S. Seon, and C. Park, "Position Estimation of Sound Source Using Three Optical Mach-Zehnder Acoustic Sensor Array," *Curr. Opt. Photon.*, vol. 1, no. 6, pp. 573–578, 2017.

[131] L. Ghamdan, M. A. I. Shoman, R. Abd Elwahab, and N. A. E. Ghamry, "Position estimation of binaural sound source in reverberant environments," *Egyptian Informatics Journal*, vol. 18, no. 2, pp. 87–93, 2017.

[132] P. N. Samarasinghe, T. D. Abhayapala, M.A. Polettfi, and T. Betlehem, "On room impulse response between arbitrary points: An efficient parameterization," in *6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, 2014.

[133] K. Zhagyparova, R. Zhagypar, A. Zollanvari, and M. T. Akhtar, "Supervised Learning-based Sound Source Distance Estimation Using Multivariate Features," in *IEEE Region 10 Symposium (TENSYMP)*, 2021.

[134] E. Georganti, T. May, S. van de Par, and J. Mourjopoulos, "Sound Source Distance Estimation in Rooms based on Statistical Properties of Binaural Signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1727–1741, 2013.

[135] M. Neri, A. Politis, D. Krause, M. Carli, and T. Virtanen, "Speaker Distance Estimation from Single Channel Audio in Reverberant Environments," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023.

[136] Daniel Krause, Archontis Politis, and Konrad Kowalczyk, "Feature overview for joint modeling of sound event detection and localization using a microphone array," in *28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021.

[137] Zhong-Qiu Wang, Peidong Wang, and DeLiang Wang, "Multi-microphone complex spectral mapping for utterance-wise and continuous speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2001–2014, 2021.

[138] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light Gated Recurrent Units for Speech Recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.

[139] G. García-Barrios, D. A. Krause, A. Politis, A. Mesaros, J. M. Gutiérrez-Arriola, and R. Fraile, "Binaural source localization using deep learning and head rotation information," in *30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 36–40.

[140] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "TIMIT Acoustic-Phonetic Continuous Speech Corpus," *Linguistic Data Consortium*, 1993.

[141] A. Politis, *Microphone array processing for parametric spatial audio techniques*, Doctoral thesis, School of Electrical Engineering, Aalto University, 2016.

[142] "Sound absorption coefficient chart: JCW acoustic supplies," https://www.acoustic-supplies.com/absorption-coefficient-chart, Accessed: 2023-06-17.

[143] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, "WHAM!: Extending Speech Separation to Noisy Environments," in *Interspeech*, 2019.

[144] F. Jacobsen and T. Roisin, "The coherence of reverberant sound fields," *The Journal of the Acoustical Society of America*, vol. 108, no. 1, pp. 204–210, 2000.

[145] R. Stewart and M. Sandler, "Database of omnidirectional and b-format room impulse responses," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 165–168.

[146] N. Bertin, E. Camberlein, R. Lebarbenchon, E. Vincent, S. Sivasankaran, I. Illina, and F. Bimbot, "VoiceHome-2, an extended corpus for multichannel speech processing in real homes," *Speech Communication*, vol. 106, pp. 68–78, 2019.

[147] A. Politis, K. Shimada, P. Sudarsanam, S. Adavanne, D. Krause, Y. Koyama, N. Takahashi, S. Takahashi, Y. Mitsufuji, and T. Virtanen, "Starss22: A dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Nancy, France, November 2022.

[148] A. Politis, K. Shimada, P. Sudarsanam, A. Hakala, S. Takahashi, D. A. Krause, N. Takahashi, S. Adavanne, Y. Koyama, K. Uchida, Y. Mitsufuji, and T. Virtanen, "Starss23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," *arXiv preprint arXiv:2306.09126*, 2023.

[149] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465–494, 2011.

[150] A. Kumar and B. Raj, "Audio event and scene recognition: A unified approach using strongly and weakly labeled data," in *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[151] I. Martín-Morató and A. Mesaros, "Strong Labeling of Sound Events Using Crowdsourced Weak Labels and Annotator Competence Estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 902–914, 2023.

[152] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[153] S. Singh, H. L. Bear, and E. Benetos, "Prototypical Networks for Domain Adaptation in Acoustic Scene Classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 346–350.