UNIVERSITÀ DEGLI STUDI DI CATANIA

Math and Computer Science Department

PhD in Computer Science Cycle XXXVI

Valerio Francesco Puglisi

Audio Analysis via Deep Learning for Forensics and Investigation Purposes

Ph.D. Thesis

Student:Valerio Francesco Puglisi

Tutor: Prof. Sebastiano Battiato

Company Tutor : Ing. Oliver Giudice

Anno Accademico 2022 - 2023

Contents

1 Abstract 12

1 Introduction **13**
1 Publications 16

2 Introduction to Sound **17**
1 The Sound 17
2 Sound Pressure Level: SPL 18
3 Wavelength, Frequency, and Spectrum 19
3.1 Wavelength 19
3.2 Frequency 19
3.3 Pure Tone Waveform 20
3.4 Periodic Waveform 21
3.5 Non-periodic Waveform 22
4 Wave Propagation and Spherical Spreading 24
4.1 Sound propagation and temperature 25
4.2 Reflections and Reverberation 27
4.3 Microphone Directionality 28
5 Human Hearing Characteristics 30
5.1 Anatomy and Physiology of the Ear 30
5.2 Psychoacoustics 33
5.3 Frequency Weighting in SPL Measurements 35
5.4 Speech Intelligibility 35
6 Signal Processing 36
7 Digital Audio 38
8 Perceptual Audio Coding 38

3 Digital Audio Forensics **40**
1 Historical cases 41
1.1 McKeever Case 41
1.2 McMillan Case 42
1.3 FBI Procedures 42
1.4 The Watergate Tapes 43
1.5 Reevaluation of the Assassination of President Kennedy 43
1.6 Talker Identification and “Voiceprints” 44
2 Audio Examiner role 45
3 Audio Examination procedure 45
3.1 Fundamental tools 47
3.2 Initial Aural Evaluation 53
3.3 Critical Listening 53
3.4 Waveform Analysis 53

3.5 Spectral Analysis 54

4 Background & State of the Art 56
1 A brief history of Deep Learning 56
2 Deep Learning Models 57
2.1 FFNNs: Feed Forward Neural Networks 57
2.2 CNNs: Convolutional Neural Networks 57
2.3 RNNs: Recursive Neural Networks 58
2.4 CRNNs: Convolutional Recursive Neural Networks 58
2.5 Residual Neural Networks 58
2.6 Encoder-Decoder Neural Networks 58
2.7 Attention-based NN 59
3 Speech Recognition 59
3.1 Deep feedforward and recurrent neural networks 60
3.2 End-to-end automatic speech recognition 60
4 Speaker Identification & Verification 62
5 Speech Enhancement 62
6 Speech Separation 63
7 Emotion Recognition 63
8 Voice Activity Detection 64
9 Sound Source Localization 65
9.1 Acoustic Environment 66
9.2 Source Type 67
9.3 Number of sources 67
9.4 Moving Sources 69
9.5 Microphones 69
9.6 Sound Source Localization Design 70
9.7 Sound Source Localization Methods 74
10 Performance Metrics 89
10.1 WER: Word Error Rate 89
10.2 CER: Character Error Rate 89
10.3 DER: Diarization Error Rate 90
10.4 PESQ 91
10.5 SDR: Source-to-Distortion Ratio 92
10.6 SI-SDR 92
10.7 SI-SDRi: Scale-Invariant Source-to-Distortion Ratio 92
10.8 SAR: Source-to-Artifact Ratio 93
10.9 SIR: Source-to-Interference Ratio 93
10.10 SNR: Signal to Noise Ratio 93
10.11 SI-SNR: Scale Invariant- Signal to Noise Ratio 93
10.12 Accuracy 94
10.13 F-score 95
10.14 Mean Absolute Error (MAE) 95
11 Datasets 96
11.1 LibriSpeech 96
11.2 TIMIT 96
11.3 CommonVoice 96
11.4 AISHELL 97
11.5 VoiceBank 97
11.6 WSJ: Wall Street Journal 97

11.7 WHAM! . . . 97
11.8 WHAMR! . . . 98
11.9 LibriMix: An Open-Source Dataset for Generalizable Speech Separation . . . 98
11.10 IEMOCAP . . . 98

5 Deep Audio Analyzer 100
1 Architecture . . . 101
2 Audio Features Visualization Module . . . 101
2.1 Preprocessing Audio Features . . . 102
3 Deep Learning Audio Inference Module . . . 107
4 Pipeline Creation and Saving . . . 107
5 Pipeline Execution and Download Report . . . 109
6 Experiments and Results . . . 109
6.1 New pipeline generation . . . 109
6.2 Experiments . . . 111
6.3 Results . . . 113

6 Sound Source Localization 120
1 PyRoomAcoustics: Audio Virtual Environment Configurations for Simulations . . . 121
2 Single Microphone Sound Source Localization . . . 122
2.1 Data Visualization . . . 123
3 Multiple Microphone Sound Source Localization . . . 137
3.1 Room . . . 137
3.2 Microphone Type . . . 139
3.3 Type of sources . . . 139
3.4 Microphones Configurations . . . 139
3.5 Methods Applied . . . 143
3.6 Results . . . 156
3.7 Single Speaker . . . 156
4 On going works and Future Works . . . 156

7 Conclusions 158

A Deep Learning Models used in Deep Audio Analyzer 160
1 Automatic Speech Recognition . . . 160
1.1 Wav2Vec & Wav2Vec2 . . . 160
1.2 CRDNN . . . 162
1.3 Conformer for KsponSpeech (with Transformer LM) . . . 163
1.4 Transformer for AISHELL (Mandarin Chinese) . . . 164
1.5 Transformer for AISHELL + wav2vec2 (Mandarin Chinese) . . . 164
2 Emotion Recognition . . . 164
2.1 Emotion Recognition with wav2vec2 base on IEMOCAP . . . 164
2.2 ECAPA-TDNN . . . 165
3 Speech Enhancement . . . 165
3.1 MetricGAN+: An Improved Version of MetricGAN for Speech Enhancement . . . 165
3.2 SepFormer: Attention is All You Need in Speech Separation . . . 166
4 Voice Activity Detection . . . 167

4.1 Voice Activity Detection with a (small) CRDNN model trained on Libriparty . . . 167
5 Speaker Verification . . . 167
5.1 Speaker Verification with ECAPA-TDNN embeddings on Voxceleb . . . 167
5.2 Speaker Verification with xvector embeddings on Voxceleb . . . 168
6 Language Identification . . . 168

B Projects outside the research field **170**
1 S.A.T.U.R.N . . . 170
1.1 Input . . . 170
1.2 Output . . . 170
1.3 Activities . . . 171

List of Figures

2.1 Air particle motion in a sound wave: longitudinal (forward and backward) parallel to the direction of propagation [164] 17
2.2 The product of frequency (cycles per second) and wavelength (meters per cycle) is the speed of sound (meters per second)[164] 20
2.3 Single cycle of a 1 kHz sinusoidal wave, also known as a pure tone, consisting of sound energy at a single frequency[164] 21
2.4 The frequency spectrum magnitude of a 1 kHz pure tone. The spectrum of a pure sinusoid contains energy only at the single frequency of that tone[164] 21
2.5 Example periodic waveform with fundamental frequency 1 kHz[164] . 22
2.6 The frequency spectrum magnitude of a 1 kHz pure tone. The spectrum of a pure sinusoid contains energy only at the single frequency of that tone[164] 23
2.7 A quasiperiodic section of approximately 5 cycles of a recorded speech signal[164] 23
2.8 Fourier transform magnitude of the quasiperiodic speech waveform of Fig. 2.7. Vertical grid depicts approximate harmonic spacing of 93.75 Hz[164] 24
2.9 Speed of sound in air as a function of air temperature[164] 25
2.10 Refraction occurs when the air near the ground is colder, leading to a downward curvature of the sound wavefront. Conversely, if warmer air is near the ground while colder air resides aloft, the result is an upward curvature of the wavefront.[164] 26
2.11 The sound directly emanating from a source and the initial reflections received by a microphone.[164] 27
2.12 Alteration in sound pressure level as the microphone is progressively distanced from a continuous source within a room characterized by reverberation. In scenarios devoid of reverberation, the relative sound level adheres to the inverse 1/r trend (a decrease of 6 dB for each doubling of distance) attributable to spherical spreading. However, in the presence of room reverberation, the sound level converges with the background reverberation level as the microphone's separation from the source increases.[164] 28
2.13 Polar diagrams depicting the directional characteristics of three prevalent microphone types: (a) omnidirectional, (b) bidirectional, and (c) unidirectional. These diagrams illustrate the variation in sound pickup relative to the angle concerning the microphone's pointing direction.[164] 29
2.14 Human Hearing System in a simplified way[164] 31
2.15 Equal-loudness contours for human hearing based on International Organization for Standardization standard 226:2003 (ISO 2003)[164] . 34

2.16 Equal-loudness contours for a human hearing based on International Organization for Standardization standard 226:2003 (ISO 2003)[164] . 36
2.17 Equal-loudness contours for a human hearing based on International Organization for Standardization standard 226:2003 (ISO 2003)[164] . 37

3.1 Digital audio display of a time span sufficiently short to show individual samples, with “connect-the-dots” lines between the sample points . . . 48
3.2 Waveform display of a time span that is too long to depict every individual sample: the display shows the signal envelope 49
3.3 The concept of the Short-Time Fourier Transform (STFT) involves the segmentation of the audio signal into overlapping blocks or frames. Subsequently, the Fast Fourier Transform (FFT) is applied to calculate the short-time spectral magnitude of each individual block [164]. . . . 50
3.4 The provided image illustrates a combined time domain and spectrographic display of a stereo (2-channel) audio recording featuring a rock-and-roll instrumental combo composed of electric guitar, bass, and drums. The total duration of the recording is 10 seconds, with consistent time scales across all four displayed panels. The frequency range in the lower two panels, represented on the vertical axis, spans from 0 to 20 kHz, utilizing a logarithmic scale. The upper two panels, characterized by a light green hue, portray the time domain envelope, which denotes signal amplitude against time, for both the left channel (top row) and right channel (second row). Meanwhile, the lower two panels, marked with an orange hue, depict the spectrograms of the left and right channels, respectively. In these spectrogram panels, the vertical axis represents frequencies from 0 to 20 kHz, while the horizontal axis represents time. Brightness of colors in the spectrograms corresponds to spectral energy: darker-colored pixels represent less energy at the respective time and frequency, while brighter-colored pixels indicate higher energy at the corresponding time and frequency. Notably, there is a recurring pattern of vertical reddish bars in the spectrogram, attributed to drum hits, and horizontal yellow stripes at lower frequencies, originating from harmonics of the electric guitar and bass lines [164]. 51
3.5 The provided images display two spectrograms of the same speech utterance, delivered by a male speaker, thereby illustrating the fundamental trade-off between time and frequency resolution. In the upper frame, longer time block lengths are employed, resulting in improved resolution in frequency, allowing for a more detailed representation of harmonic partials. However, this enhancement in frequency resolution comes at the expense of clarity in rendering the sound’s attack and release characteristics, which appear somewhat blurred in the spectrogram. Conversely, the lower frame utilizes shorter time block lengths, providing superior resolution in time, which highlights the "edges" occurring when the signal undergoes changes. However, this advantage in time resolution comes at the cost of reduced frequency detail. The overall duration of the audio segment depicted in both frames is 2.5 seconds, with the frequency range spanning from 0 to 10 kHz, utilizing a linear scale for reference [164]. 52

3.6 Subtle trade-offs involving time-frequency resolution are observed. The uppermost two rows feature spectrograms of the left and right channels in a stereo audio recording, displaying a marginal refinement in frequency resolution. Conversely, the lowermost two rows portray spectrograms of the identical stereo recording, showcasing a marginal enhancement in time resolution. The recording spans a total duration of 14 seconds and encompasses frequencies ranging from 0 to 4 kHz, with a linear scale representation [164]. 55

4.1 Speaker Identification and Speaker Verification Processes 62
4.2 Sound Source Localization System anatomy 66
4.3 Ambisonics Microphones rapresentation 70
4.4 Microphones Configurations 71
4.5 Sound Source Localization System Design 72

5.1 **Angular Front End:** The User Interface of Deep Audio Analyzer is divided into pages and components in order to categorize all the functions separately. All the modules of Deep Audio Analyzer are developed on different pages. **Flask Backend:** Deep Learning Audio Analyzer employs a simple software stack (i.e., Python → PyTorch → SpeechBrain → HuggingFace → Flask → Angular) to avoid dealing with too many levels of abstraction. It is developed on top of SpeechBrain and HuggingFace directly, with external APIs that can retrieve the newest model uploaded from the SpeechBrain community and other Companies. 101
5.2 Architecture of Audio Feature Visualization Module. 102
5.3 Audio Feature Visualization Module. 102
5.4 Pipeline Creation and Dynamic Audio Analysis Flowchart 108
5.5 Architecture of Audio Feature Visualization Module. 108
5.6 Select Task and Model to create a step of audio analysis pipeline. . . 108
5.7 Pipeline Creation and Dynamic Audio Analysis Flowchart 109
5.8 Save Pipeline created during the analysis. 110
5.9 Pipeline Execution Audio Analysis Flowchart 110
5.10 Pipeline Multi-speaker Multi-Language ASR : Speech Separation + Language ID + ASR 111
5.11 Pipeline Automatic Speech Recognition in noisy environment 111
5.12 Automatic Speech Recognition with Wav2Vec 2.0 + CTC model trained on Librispeech and tested on Augmented Noisy Librispeech . 114
5.13 Audio Enhancement with MetricGAN+ and Automatic Speech Recognition with Wav2Vec 2.0 + CTC model trained on Librispeech and tested on Augmented Noisy Librispeech 115

6.1 Empty Room Museum Size 122
6.2 The simulated measurements take into account 8 angles on the same plane: 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°. As for the distances, a range from 1 to 10 meters is considered with a step of 1 meter. Therefore, for each audio sample of a Speaker ID, 80 measurements are taken = 10 distances × 8 angles. 123
6.3 Multiple Speech Chunks processed on Raw Waveform of Speaker ID 19 at different distances from microphone 124

6.4 Multiple Speech Chunks processed with Spectrogram of Speaker ID 19 at different distances from microphone . . . 125
6.5 Multiple Speech Chunks processed with MFCC of Speaker ID 19 at different distances from microphone . . . 125
6.6 Multiple Speech Chunks processed with Wav2Vec2 of Speaker ID 19 at different distances from microphone . . . 125
6.7 Multiple Speech Chunks processed of Multiple Speaker at different distances from microphone . . . 126
6.8 Multiple Speech Chunks processed with Spectrogram of Multiple Speaker at different distances from microphone . . . 126
6.9 Multiple Speech Chunks processed with MFCC of Speaker ID 19 at different distances from microphone . . . 127
6.10 Single2 Speech Chunks processed with Wav2Vec2 of Multiple Speaker at different distances from microphone . . . 127
6.11 Data Visualization with T-SNE applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 128
6.12 Data Visualization with T-SNE applied on Spectrograms of a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 128
6.13 Data Visualization with T-SNE applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 129
6.14 Data Visualization with T-SNE applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 129
6.15 Data Visualization with PCA applied on Raw Waveform of Multiple Speech Chunks of a Single Speaker at different distances from microphone 130
6.16 Data Visualization with PCA applied on Spectrogram of Multiple Speech Chunks of a Single Speaker at different distances from microphone 130
6.17 Data Visualization with PCA applied on MFCC of Multiple Speech Chunks of a Single Speaker at different distances from microphone . . 131
6.18 Data Visualization with PCA applied on Wav2Vec features extracted from Multiple Speech Chunks of a Single Speaker at different distances from microphone . . . 132
6.19 Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 133
6.20 Data Visualization with PCA applied on Spectrograms of a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 133
6.21 Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 134
6.22 Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 134
6.23 Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 135

6.24 Data Visualization with PCA applied on Spectrograms of a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 135
6.25 Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 136
6.26 Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone . . . 136
6.27 An Office Room simulated with PyRoom Acoustic with 3 Stations with chair and desk . . . 137
6.28 Empty Room simulated with PyRoom Acoustic . . . 138
6.29 Cardioid Microphone for simulations . . . 139
6.30 2D perspective of microphones configuration . . . 140
6.31 3D perspective of microphones configuration with area covered by cardioid microphones . . . 140
6.32 2D perspective of Triaural microphones configuration . . . 141
6.33 3D perspective of Triaural microphones configuration with area covered by cardioid microphones . . . 141
6.34 2D perspective of microphones configuration . . . 141
6.35 3D perspective of microphones configuration with area covered by cardioid microphones . . . 141
6.36 2D perspective of microphones configuration . . . 142
6.37 3D perspective of microphones configuration with area covered by cardioid microphones . . . 142
6.38 **MUSIC with 1 speaker** values of azimuth for configuration experiments in degrees: a) Binaural configuration: Real: 321.08, Recovered: 303, Error: 18.07 b) Triaural configuration: Real: -26.56, Recovered: 270, Error: 63.43 c) Tetra-aural configuration: Real: -38.92, Recovered: 318, Error: 3.07 d) Circular Array configuration: Real: 45, Recovered azimuth: 41, Error: 4 . . . 144
6.39 **MUSIC with 2 speakers:** values of azimuth for configuration experiments in degrees: a) Binaural configuration: Real azimuth: [270, 45], Recovered: [30, 330], Error: [15, 60] b) Triaural configuration: Real: [45, 270], Recovered: [181, 218], Error: [136, 52] c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0] d) Circular Array configuration: Real: [45, 270], Recovered: [43, 274], Error: [2, 4] . . . 145
6.40 **NormMUSIC with 1 speaker:** azimuth for experiments (in degrees): a) Binaural configuration: Real: 321.08, Recovered: 286, Error: 35.07 b) Triaural configuration: Real: -26.56, Recovered: 90, Error: 116.56 c) Tetra-aural configuration: Real: -38.92, Recovered: 284, Error: 37.07 d) Circular Array configuration: Real: 45, Recovered: 39, Error: 6. . . . 146
6.41 **NormMUSIC with 2 speakers:** azimuth for experiments (in degrees): a) Binaural configuration: Real: [45, 270], Recovered: [157, 203.], Error: [112, 67] b) Triaural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0] c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0] d) Circular Array configuration: Real: [45, 270], Recovered: [42, 271], Error:[3,1] 147

6.42 **TOPS with 1 speaker:** azimuth for experiments (in degrees): a) Binaural configuration: Real: 321.08, Recovered: 241, Error: 80.07 b) Triaural configuration: Real: -26.56, Recovered: 235, Error: 98.43 c) Tetra-aural configuration: Real: -38.92, Recovered: 84, Error: 122.92 d) Circular Array configuration: Real: 45, Recovered: 38, Error:7 . . 148
6.43 **TOPS with 2 speakers** azimuth for experiments (in degrees): a) Binaural configuration: Error during execution -> index -1 is out of bounds for axis 0 with size 0 b) Triaural configuration: Real: [45, 270], Recovered: [90, 147], Error:[45, 123] c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error:[45. 0.] d) Circular Array configuration: Real: [45, 270], Recovered: [28, 268], Error:[17, 2] . . . 149
6.44 **WAVES** values of azimuth for configuration experiments in degrees: a) Binaural configuration: Real: 321.08, Recovered: 281, Error: 40.07 b) Triaural configuration: Real: -26.56, Recovered: 339, Error: 5.56 c) Tetra-aural configuration: Real: -38.92, Recovered: 348, Error: 26.92 d) Circular Array configuration: Real: 45, Recovered: 49, Error:4 . . 151
6.45 **WAVES** speakers' azimuth for experiments (in degrees): a) Binaural configuration: Error b) Triaural configuration: Real : [45, 270], Recovered: [196, 357], Error: [151, 87] c) Tetra-aural configuration: Error -> Singular Matrix d) Circular Array configuration: Real : [45, 270.], Recovered: [41, 164], Error: [4, 106] 151
6.46 **SRP-PHAT** speaker azimuth for experiments (in degrees): a) Binaural configuration: Real: 321.08, Recovered: 272, Error: 49.07 b) Triaural configuration: Real: -26.56, Recovered: 90, Error: 116.56 c) Tetra-aural configuration: Real: -38.92, Recovered: 306, Error: 15.07 d) Circular Array configuration: Real: 45, Recovered: 39, Error: 6 . . 152
6.47 **SRP-PHAT** speakers' azimuth for experiments (in degrees): a) Binaural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0] b) Triaural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0] c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0] d) Circular Array configuration: Real: [45, 270] Recovered: [88, 271], Error: [43, 1] 153
6.48 GCC-PHAT: Values of [x,y] in cartesian axes per configuration in meters: a) Binaural configuration: Groundtruth: [0.6 1.25], GCC estimate: [1.36 0.] b) Triaural configuration: Groundtruth: [2. 1.], GCC estimate: [1.39 1.36] c) Tetra-aural configuration: Groundtruth: [0.5 2.], GCC estimate: [1.71 1.17], GCC Error: [-1.21 0.82] 154
6.49 NGCC-PHAT: position of [x,y] in cartesian axes and for different configuration in meters: a) Binaural configuration: Groundtruth: [0.6 1.25], NGCC estimate: [1.29 0] b) Triaural configuration: Groundtruth position: [2. 1.], NGCC estimate: [1.44461964 1.43318727] c) Tetra-aural configuration: Groundtruth: [0.5 2.], GCC estimate: [1.31010009 1.41599011], GCC Error: [-0.81010009 0.58400989] 155

List of Tables

4.1 Best Speech Separation Performance on the WSJ0-3mix dataset, showed in Attention Is All You Need In Speech Separation(2021)[249] 64

5.1 Deep Learning Models: ASR: Automatic Speech Recognition, ER: Emotion Recognition, LI: Language Identification, SE: Speech Enhancement, SS: Speech Separation, SV: Speaker Verification, VAD: Voice Activity Detection 116
5.2 Evaluation of Automatic Speech Recognition Models on different Test Datasets 117
5.3 Evaluation of Speech Separation Models on different Test Datasets . . 118
5.4 Evaluation of Wav2Vec 2.0 + CTC Automatic Speech Recognition Model on Noisy Librispeech 118
5.5 Evaluation of Wav2Vec 2.0 + CTC Automatic Speech Recognition Model on Noisy Librispeech 119

6.1 **Single Speaker**: DOA Estimation Error Results: Speaker azimuth for experiments in degrees 156
6.2 **Multiple Speakers (2 Speaker)**: DOA Estimation Error Results: Speaker azimuth for experiments in degrees 156

1 Abstract

This doctoral dissertation explores the fusion of deep learning and state-of-the-art models, to advance the fields of audio forensics, audio intelligibility, and enhancement. In an era where audio recordings hold critical significance across multiple domains, the ability to authenticate and enhance them is paramount.Deep learning models, such as convolutional and recurrent neural networks, are harnessed to detect tampered audio recordings, enhancing the authentication process. Additionally, cutting-edge Transformers, renowned for their sequence-to-sequence capabilities, are employed to tackle the challenges of audio intelligibility and enhancement. These models can effectively denoise and clarify audio recordings, improving their overall quality. Practical tools and methodologies are developed to address real-world scenarios, accounting for noise, compression artifacts, and variations in recording devices. The research contributes significantly to the reliability of audio evidence, benefiting fields like law enforcement, legal proceedings, and digital media forensics. In summary, this doctoral research represents a substantial advancement in the realm of audio forensics, intelligibility, and enhancement. By combining deep learning state-of-the-art models, it offers comprehensive solutions to the authentication, clarity, and enhancement of audio recordings in the information-driven era.

Chapter 1

Introduction

This PhD has a research theme proposed that aims to develop novel algorithms for audio analysis with the specific purposes of forensic investigations.

Audio Forensics, a subcategory of forensic acoustics, has taken on an increasingly prominent role in the field of law enforcement, criminal investigation, and justice over the past few decades. This specialized branch of forensic science, dedicated to the capture, analysis, and interpretation of audio recordings within the context of legal investigations, has enabled the provision of decisive evidence in an ever-increasing number of legal cases.

Audio forensic investigations focus on three core aspects: authenticity, enhancement, and interpretation. Ensuring the authenticity of audio recordings is crucial as investigator deductions rely on the recording conditions. Examiners must validate the chain of custody, detect intentional tampering, and prevent accidental modifications. Audio enhancement is commonly requested to address non-ideal acoustic environments, emphasizing features of interest for court presentations. Interpretation involves reconstructing timelines, transcribing dialogues, and identifying unknown sounds, considering other evidence and testimonies.

Recently, applications of novel deep learning solutions to forensics investigations have experienced unprecedented growth in interest and obtained results, with many researchers developing innovative algorithms and models to solve these complex problems.

The proposed research activity aims to operate on the state of the art to identify and improve the techniques to identify people, recognize emotion, automatic speech recognition, etc.. starting from audio. The technological opportunity of the proposed research topic is offered by the recent achievements of Deep Learning which offers an optimal starting point for solving the problems highlighted.

A study of the state of the art is necessary to identify the best solutions available for solving the main tasks such as speaker verification, emotion recognition, automatic speech recognition, etc .. of people by exploiting the union of the two disciplines of Deep Learning Audio Models and Digital Forensics.

The research activity will be focused on the creation of audio analysis processes for the tasks of identifying people using audio and extracting information from it,

which will flow, at first, in the acquisition of different databases relating to monitoring.

Another important problem that is considered from this PhD is reproducing published experiments and results remain a significant challenge due to the needed programming skills required. This is a problem because often is not possible to verify other scientists' experiments.

This challenge is further compounded by the lack of (or an extremely limited) standardization in the way experiments are conducted. This issue results in a significant amount of time being spent by researchers trying to get other researchers'code to work, which leads to a significant waste of resources. Open-source toolkits have largely driven the development of speech-processing technologies.

With the emergence of general-purpose deep learning libraries, more flexible speech recognition frameworks have emerged and hubs where scientists load trained models for others to download.

The main goal (as described in the PhD project) is to create a tool that integrates these new technologies and enables users to analyze audio data through the visualization audio features, running several different models of Deep Learning, evaluate the performance of pre-trained models and create new audio analysis workflows by combining deep neural network models. These features will be implemented by combining the usability needed for the user and the experiments' reproducibility, with the hope of encouraging scientists to standardize and share deep learning models or audio pipeline processes with the research community.

For these reasons, we will develop a platform called Deep Audio Analyzer. With this platform examiners and researchers can perform these features without developing any code.
The tool also provides dedicated modules to test state-of-the-art models on customized data and combine models to create a new deep learning audio processing pipeline, combing for tasks such as Automatic Speech Recognition, Speech Enhancement, Speaker Separation, Speaker Verification and Voice Activity Detection, Sound Source Localization.

The present study will contribute to examinates what are the main Deep Neural Networks that can help the Audio Forensic field in a better way than the traditional methods. This research aims to provide a tool that covers Enhancement, Interpretation and Localization, which are part of the three main goals of Audio Forensics.

To explore Sound Source Localization several studies will be conducted to identify the microphones' position for forensic purposes.

To achieve the aims of the project, the following activities are envisaged:

1. Study of the State of the Art.

2. Definition of system requirements in a specific application context.

    - Preprocessing Audio Via Classical methods in literature present in librosa library to have the capability to have a visual analysis of the features extracted

- Applying State-of-the-art Models via online search and/or uploading private pre-trained models (Upload of private models via dictionary is done, model search)
- Combine Deep Neural Network models to create customized audio analysis pipelines to extract the interested information.
- Create a pipeline for a single task that uses all models for that task to compare all the SOTA models for a task in the interested audio files.
- Save, Share and Reuse pipelines in different files or datasets.

3. Collection of data to be used for the design and testing of the algorithms.
4. Definition of the measures useful to evaluate the algorithms in the considered context.
5. State-of-the-art testing considering the system requirements and the application context.
6. Design, development and evaluation of innovative algorithms in the reference context.
7. Development of a demonstrator.
8. Dissemination of results through the publication of scientific articles, the exhibition of the work at international scientific events and the production of patents useful for technology transfer.

The project includes a period of study abroad at the research group of Dr. Salvatore Livatino, University of Hertfordshire, UK and a period of study at the company iCTLab s.r.l. (https://www.ictlab.srl/) spinoff of the The University of Catania, specialized in digital investigations over the years has developed know-how on the current issue.

The remainder of the PhD thesis is organized as follows. In Chapters 2 and 3, we explore introduction topics comprising the Sound Principles, Digital Audio Forensics and Deep learning models used in the audio context. Chapter 4 reports the research studies done during the PhD project that comprehend the state of the art of tasks that can be used for audio forensic analysis. Chapter 65 delves into the implementation and development of new technology from previous studies and works done for Sound Source Localization. Chapter 7 is the conclusion of the PhD Thesis.

1 Publications

This PhD thesis is an extract from the following papers published and submitted during these three years of studies.

- (Conference: in proceedings) Velerio Francesco Puglisi, Oliver Giudice, Sebastiano Battiato (2023). **Investigation on Sound Source Localization: Past and Recent approaches**

- (Journal: Submitted) Velerio Francesco Puglisi, Oliver Giudice, Sebastiano Battiato (2023).**Deep Audio Analyzer: A Novel Framework to improve on Audio Forensics issues** AAFS American Academy of Forensic Sciences : Journal Of Forensic Sciences

- Velerio Francesco Puglisi, Oliver Giudice, Sebastiano Battiato (2023).**Deep Audio Analyzer: A Framework to Industrialize the Research on Audio Forensics** AAFS 2024: 76th Annual AAFS American Academy of Forensic Sciences 76^{th} Annual Scientific Conference.

- Velerio Francesco Puglisi, Oliver Giudice, Sebastiano Battiato (2023).**Deep Audio Analyzer: A Framework to Industrialize the Research on Audio Forensics** The 2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering - IEEE MetroXRAINE 2023 October 25-27

- Avanzato, Roberta, Francesco Beritelli, and Valerio Francesco Puglisi. **"Dairy Cow Behavior Recognition Using Computer Vision Techniques and CNN Networks."** 2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS). IEEE, 2022.

- Michele Russo, Valerio Francesco Puglisi, Roberta Avanzato, Francesco Beritelli. **A CNN-based Audio Sensor for Rainfall Estimation: Implementation on Embedded Board** . The 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 22-25 September, 2021, Cracow, Poland

Chapter 2

Introduction to Sound

1 The Sound

Sound in the gaseous medium emanates from vibrational perturbations. In contradistinction to wind, where air particles engage in continuous motion over considerable distances, the oscillation of a surface elicits localized reciprocating excursions among particles. While the vibratory surface undergoes outward motion, neighbouring air particles experience condensation due to impelling forces. Inversely, during the alternate phase of vibration, wherein the surface retracts, air particles near it undergo rarefaction or expansion. The oscillatory pattern of compression and rarefaction contiguous to the vibrating interface precipitates analogous cycles of push and pull among adjacent air particles, cascading the effect onward through successive air layers. The emergent outcome is a disseminating wave characterized by alternating domains of elevated and diminished pressure, as depicted in Figure 2.1.

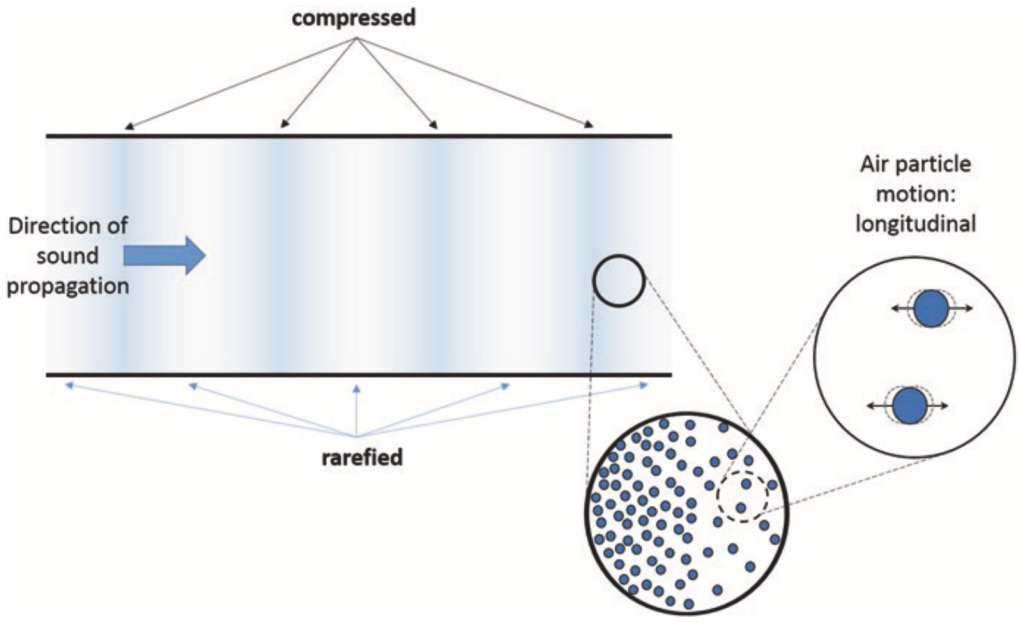


Figure 2.1: Air particle motion in a sound wave: longitudinal (forward and backward) parallel to the direction of propagation [164]

The sound wave, epitomizing a longitudinal perturbation, entails oscillatory motions wherein air particles oscillate to and fro from their equilibrium configurations in alignment with the propagating wavefront's trajectory. This longitudinal modality

proves intricate to delineate visually. Conventional visual representation manifests as a bi-dimensional graph encapsulating the temporal evolution of acoustic pressure, albeit inadvertently fueling misapprehensions of vertical oscillations as the wave propagates. An enhanced comprehension perceives particle oscillations as directed "in and out," correlating with commensurate fluctuations in acoustic pressure during wave transmission [125].

Pertinent to acknowledge is that acoustic pressure deviations remain minute when compared to ambient atmospheric pressure. Earth's gravitational force sustains an approximately 100 km expanse of our atmospheric strata in proximity to the planet's surface, engendering nominal sea-level air pressure, denoted as $1atm = 101.325kPa \approx 1 \times 10^5 Pa$. By contrast, characteristic pressure fluctuations engendered by sonic vibrations are infinitesimal, often confined within the millipascal range ($10^{-3}Pa$). In actuality, the faintest perceivable auditory manifestations manifest pressure amplitudes approximating $20\mu Pa(2 \times 10^{-5}Pa)$, tantamount to a mere fraction of 1 part in 5 billion when compared to nominal atmospheric pressure.

Contrastingly, exceedingly intense sonic phenomena, such as those encountered in rock-and-roll concerts or proximate to industrial machinery, may exhibit amplitudes surpassing $1Pa$. Notably, despite their ear-piercing resonance, these events constitute merely 1 part in 50,000 relative to nominal atmospheric pressure.

2 Sound Pressure Level: SPL

The numerical range of audible acoustic pressures, spanning from $2 \times 10^{-5}Pa$ to $1Pa$, entails figures of considerable magnitude, rendering them less practical for concise notetaking and printing. As a convention, the audible spectrum of sound pressures is commonly expressed logarithmically. In this representation, the faintest perceptible auditory signal assumes a level of zero, while the most resonant ambient sound is denoted by just a few digits. This scientific portrayal is accomplished through the utilization of the $bel[B]$, being the logarithm to the base 10 of the power or intensity ratio, often expressed in $watts[W]$ or intensity $[W/m^2]$:

$$B = log_{10}\left(\frac{power_1}{power_0}\right)[W] = log_{10}\left(\frac{intensity_1}{intensity_0}\right)[W/m^2] \tag{2.1}$$

For the conversion of the bel representation to sound, a transition from acoustic pressure $[pascal]$ to acoustic intensity $[watts/m^2]$ is necessary. This transition rests on the relationship wherein acoustic intensity is proportionate to the square of acoustic pressure. Consequently, the bel can be formulated in terms of pressure:

$$bel = log_{10}\left(\frac{power_1^2}{power_0^2}\right)[pascal] = log_{10}\left(\frac{power_1}{power_0}\right)[watts/m^2] \tag{2.2}$$

The customary mode of expression involves the decibel [dB], denoting a precision that is a tenth of a bel. Given the equivalence of 10 dB to 1 bel, measurements in decibels amount to 10 times the measurement represented in bel:

$$decibel[dB] = 20log_{10}\left(\frac{power_1}{power_0}\right) \tag{2.3}$$

The sound pressure level (SPL) in decibels employs the stipulation that $pressure_0$ equals $20\mu Pa(20\mu Pa = 0.00002Pa)$, while $pressure_1$ signifies the effective pressure

(root-mean-square or RMS) as gauged by a microphone.

The selection of 20 μPa as the reference pressure is apt, as it approximately aligns with the threshold of human auditory perception. Specifically, an acoustic signal possessing an effective pressure of 20 μPa corresponds to zero dB SPL. A signal rated at 100 dB SPL is profoundly resonant, nearly attaining the threshold of pain for the auditory system. Ergo, the spectrum of viable sound levels within the realm of human audibility spans from 0 to 100 dB SPL. It is imperative that measurements using a sound level meter consistently bear the dB label and reference to sound pressure, as exemplified by "60 dB SPL re 20 μPa [100].

Given the nonuniform sensitivity of the human ear across the frequency spectrum, sound pressure level assessments frequently entail the use of weighting filters that approximate the frequency-dependent characteristics of auditory sensitivity.

The sound wave, signifying alternating high and low-pressure disturbances, propagates through the air at a rate termed the speed of sound. This speed is contingent upon the interplay between acoustic pressure and the ensuing vibratory motion (particle velocity) of air particles. At 20 °C (room temperature), the speed of sound in air approximates $343 m/s$. In contrast, the speed of light reaches $3 \times 10^8 m/s$, rendering it nearly a million times faster than sound.

For the sake of practical estimations in sound propagation, several heuristic approximations exist. Notably, the American approximation posits that sound traverses about 1 foot per millisecond and requires approximately 5 seconds to cover a mile. Meanwhile, the metric heuristic suggests a rate of about 35 cm per millisecond, translating to roughly 3 seconds for sound to traverse a kilometre.

3 Wavelength, Frequency, and Spectrum

In the context of oscillating sound sources, such as a loudspeaker cone executing to-and-fro motions or a guitar string undergoing cyclic vibrations, the resultant sound manifests as alternating cycles of elevated and diminished pressure.

3.1 Wavelength

The temporal extent required for one complete oscillation cycle is defined as the vibration's period. For instance, in the case of a vibrating string, the period signifies the duration taken for the string to traverse from one extremity to the opposite extremity and back to its initial position, accomplishing a single oscillation cycle. During the span of a single oscillation (i.e., one period), the ensuing sound pressure perturbation advances through the air at the speed of sound, traversing a specific distance termed the wavelength, expressed in [meters/cycle]. Conceptually, the wavelength denotes the distance traversed by the sound wave within the temporal expanse of one oscillation cycle.

3.2 Frequency

Sound oscillations find their commonplace expression in terms of an oscillation rate: the count of oscillation cycles occurring within a second [cycles/second]. This oscillation rate is conventionally denoted as the frequency of the oscillation and is quantified in hertz (abbreviated as Hz), denoting cycles per second.

In instances where oscillation transpires at a lower frequency, each cycle's period expands, consequently fostering greater inter-cycle wave travel. In effect, lower frequency begets longer wavelengths. Conversely, when oscillation attains heightened frequency, the temporal margin for pressure disturbance propagation between cycles diminishes, resulting in a shorter wavelength at a higher frequency.

Mathematically, the interrelation between frequency [f, cycles per second or Hz] and wavelength [λ, meters] is expressed as

$$c = f\lambda \Rightarrow f = \frac{c}{\lambda} \Rightarrow \lambda = \frac{c}{f} \tag{2.4}$$

where c represents the speed of sound [meters/second]. Thus, higher-frequency sound phenomena correspond to shorter wavelengths, whereas lower-frequency manifestations align with longer wavelengths (Fig. 2.2).

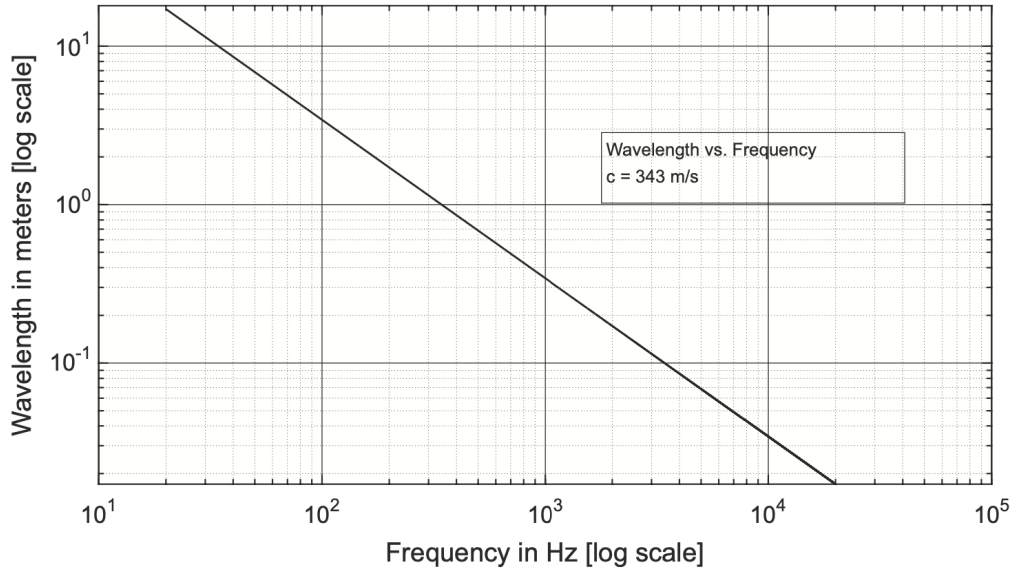


Figure 2.2: The product of frequency (cycles per second) and wavelength (meters per cycle) is the speed of sound (meters per second)[164]

3.3 Pure Tone Waveform

The most elementary manifestation of enduring sound encapsulates energy characterized by a solitary frequency, constituting what is termed a "pure tone". Graphical representation of the waveform for such single-frequency sound finds illustration in Figure 2.3. This waveform, under mathematical purview, is identified as a sine wave or a sinusoid.

The vertical axis of the graphical depiction in Figure 2.3 is attributed to a pertinent parameter, such as pressure, voltage, or displacement, whereas the horizontal axis delineates time. The visualization portrays a solitary cycle of the sine wave, taking a span of 1 ms (1/1000th of a second) in this particular instance. Given that one complete cycle corresponds to a *period* $T = 1ms$, the *frequency* intrinsic to this waveform (1/T) stands at $1kHz$, encompassing $1000[cycles/second]$.

As a result of a pure tone's (sinusoidal waveform) energy being concentrated solely at its repetition rate frequency, the spectrum plot of a sinusoid takes on the appearance of a solitary "frequency line." Theoretical depiction of the spectrum

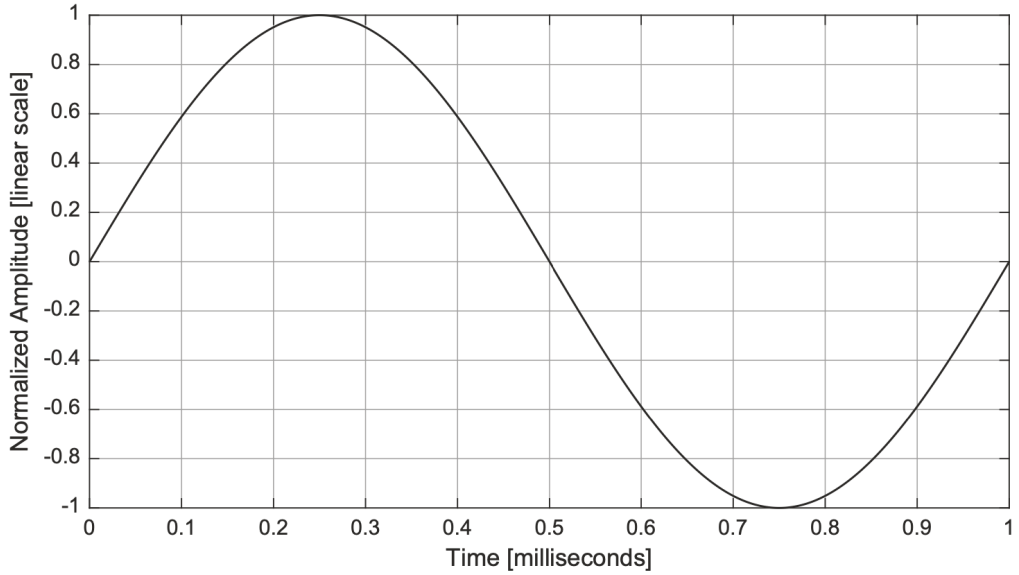


Figure 2.3: Single cycle of a 1 kHz sinusoidal wave, also known as a pure tone, consisting of sound energy at a single frequency[164]

about a 1 kHz sinusoid aligns with the illustration presented in Figure 2.4 where the spectrum harbours energy solely at the frequency of 1 kHz, with no energy presence across any other frequency range.

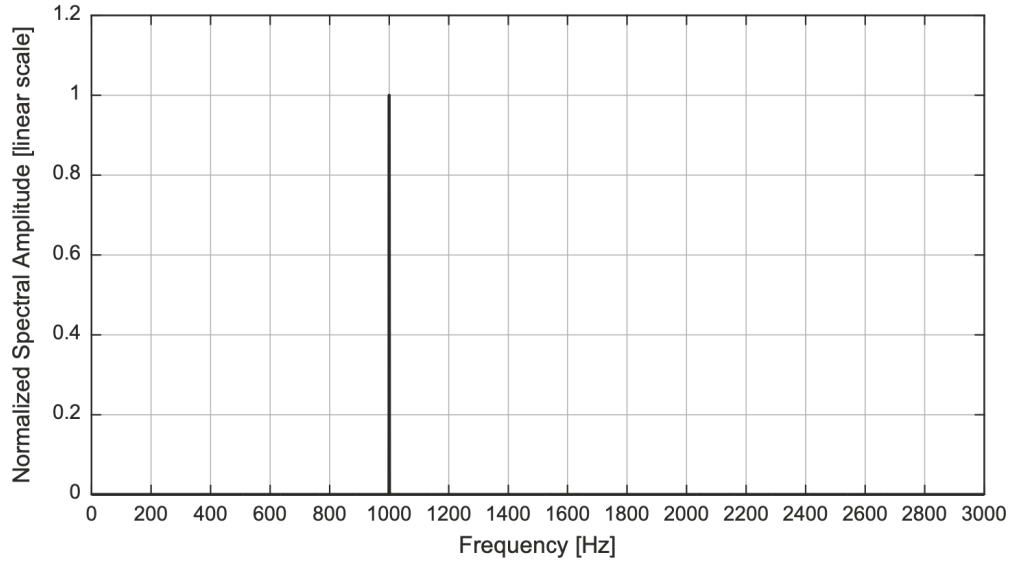


Figure 2.4: The frequency spectrum magnitude of a 1 kHz pure tone. The spectrum of a pure sinusoid contains energy only at the single frequency of that tone[164]

3.4 Periodic Waveform

Persistent auditory patterns characterized by recurrent waveforms, as exemplified in the vocalization of the vowel sound "ahhh" at a consistent pitch, exhibit a spectrum of heightened intricacy when contrasted with the straightforward, single-frequency composition of a pure tone [100]. Such periodic waveforms give rise to a spectrum featuring harmonics, a characteristic wherein energy is exclusively concentrated at

frequencies that constitute integer multiples of a fundamental frequency, denoted as F_0. This configuration is illustrated in Figure 2.5.

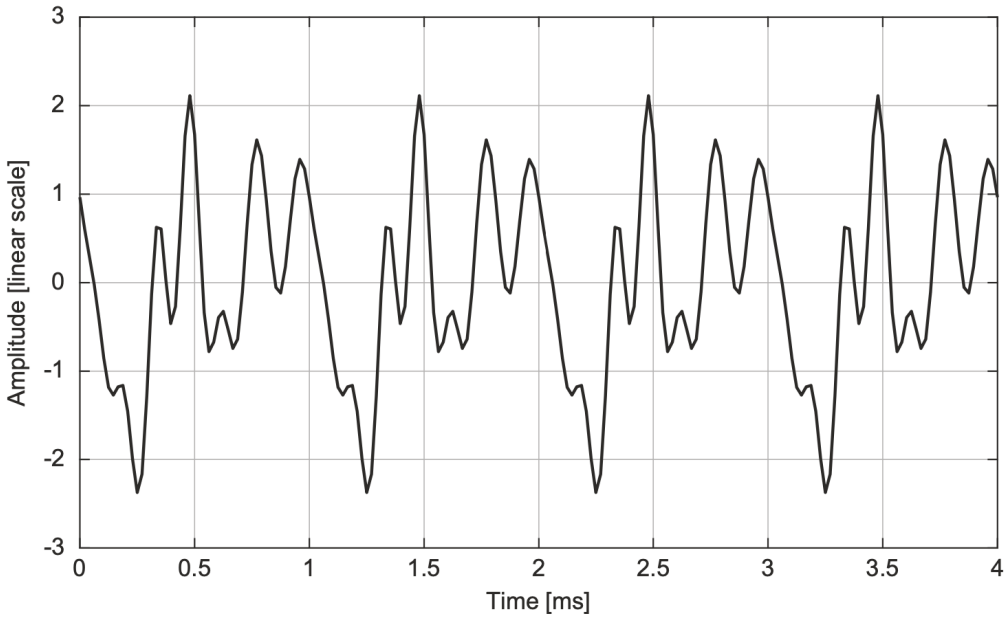


Figure 2.5: Example periodic waveform with fundamental frequency 1 kHz[164]

The spectral amplitude distribution corresponding to the periodic waveform depicted in Figure 2.5 is visualized in Figure 2.6. It's noteworthy that the constituent spectral components align exclusively with harmonics, which are integer multiples of the fundamental frequency of 1 kHz (1 kHz, 2 kHz, 3 kHz, 4 kHz, and so forth).

For a waveform that embodies pure periodicity, akin to the waveform presented in Figure 2.5, the computational determination of harmonic amplitudes as depicted in Figure 2.6 can be executed through a mathematical methodology recognized as the Fourier series.

3.5 Non-periodic Waveform

Conversely, the evaluation of the spectrum for a non-periodic waveform can be achieved by employing the Fourier transform, another mathematical technique. The Fourier transform serves as a pivotal tool for comprehending the spectral attributes of a diverse array of signals pertinent to the realm of audio forensicsn[100].

To elucidate further, consider an illustrative excerpt of a male speech recording, displayed in Figure 2.7. Notably, the waveform assumes a quasi-periodic semblance; however, each cycle diverges from being a precise replica of its counterparts. The Fourier transform magnitude of this waveform is delineated in Figure 2.8.

Unlike the pristine harmonic sequence of monofrequency spikes characterizing the spectrum of the theoretically boundless waveform in Figure 2.5 and the corresponding spectrum in Figure 2.6, the Fourier analysis of a finite-length quasi-periodic speech waveform manifests broadened spectral lines and inharmonicity. This divergence emanates from cycle-to-cycle variation inherent to the speech waveform and the finite temporal domain for signal observation within the Fourier transform [8].

In scenarios involving the coexistence of multiple sound sources, the discerned spectrum (Fourier transform magnitude) encompasses an additive amalgamation

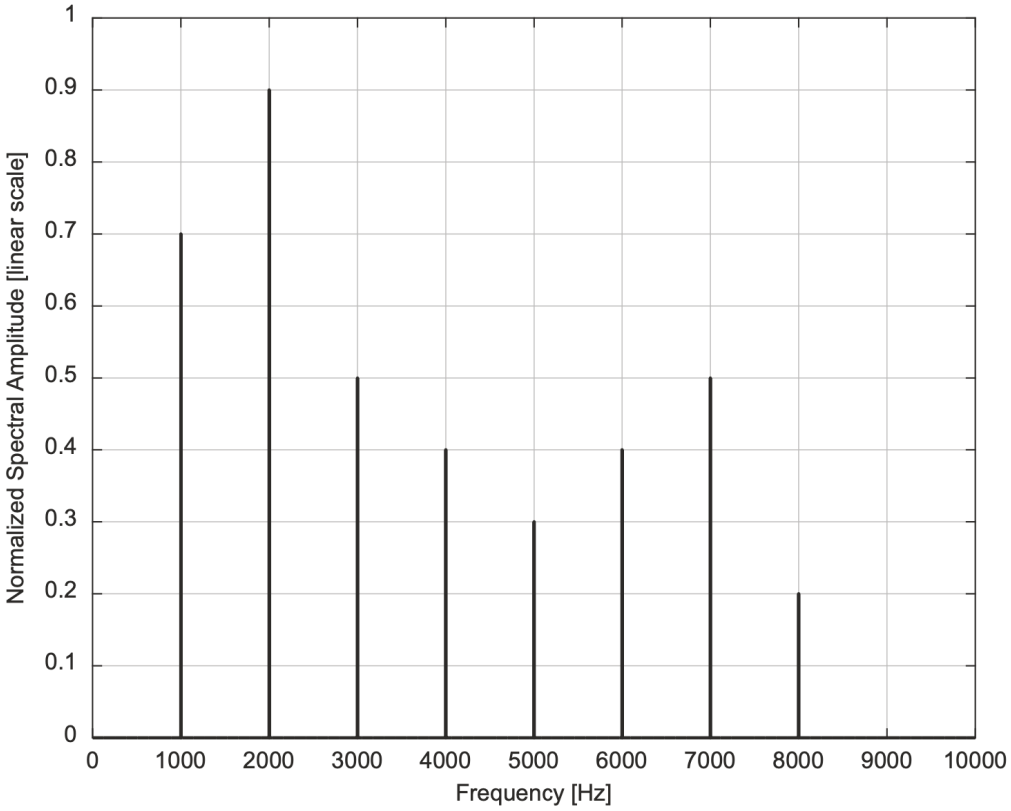


Figure 2.6: The frequency spectrum magnitude of a 1 kHz pure tone. The spectrum of a pure sinusoid contains energy only at the single frequency of that tone[164]

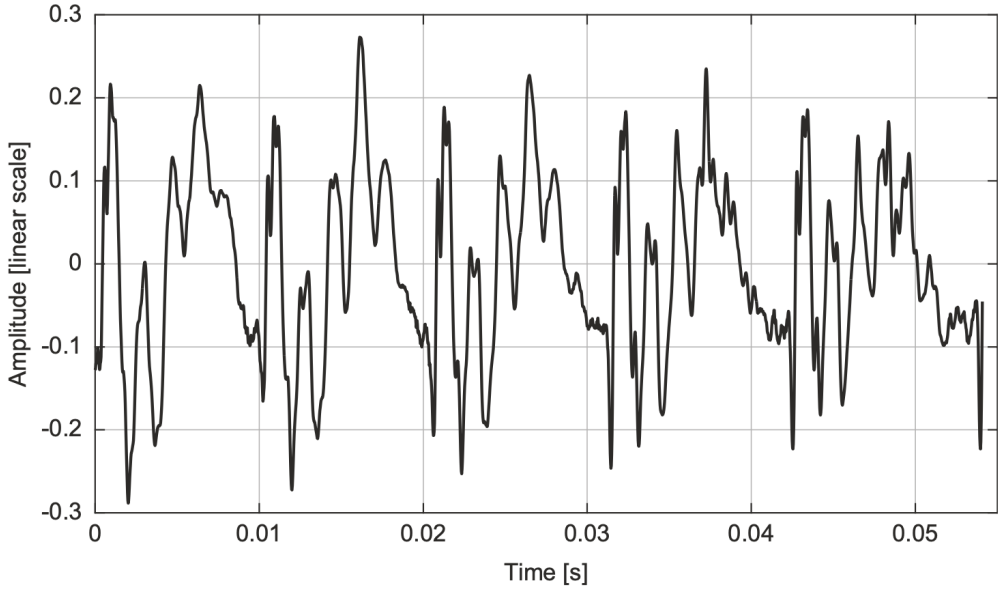


Figure 2.7: A quasiperiodic section of approximately 5 cycles of a recorded speech signal[164]

of spectral constituents originating from diverse sources. This intermixture arises deliberately when musical instruments converge as an ensemble. To expound, a musical signal with a fundamental frequency of 100 Hz will engender harmonic energy within its spectrum at 200, 300, 400, 500, 600, 700, 800, 900 Hz, and so on. Should a simultaneous musical signal bearing a fundamental frequency of 150 Hz be present,

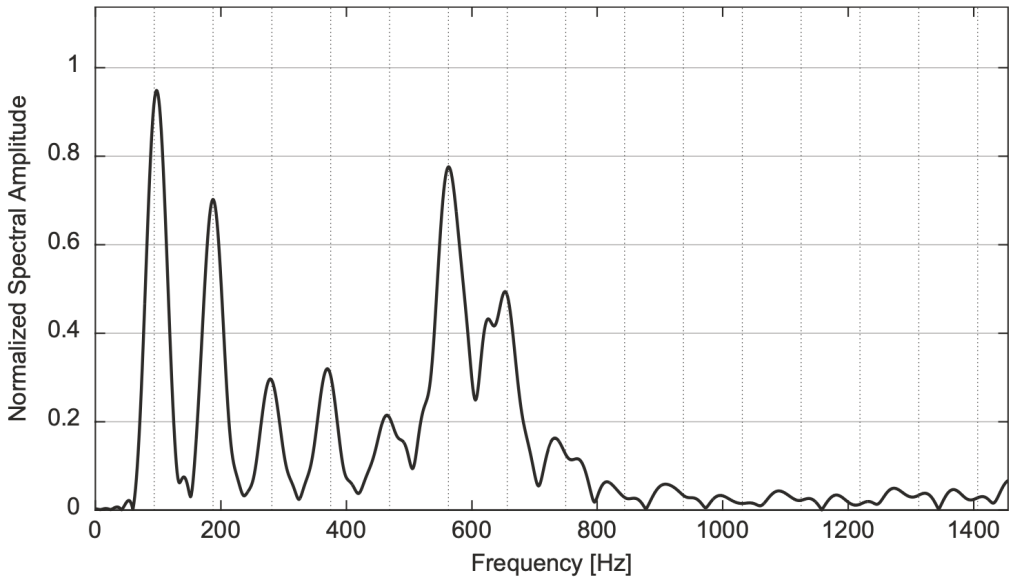


Figure 2.8: Fourier transform magnitude of the quasiperiodic speech waveform of Fig. 2.7. Vertical grid depicts approximate harmonic spacing of 93.75 Hz[164]

its spectral energy will be manifest at 150, 300, 450, 600, 750, 900 Hz, and so forth.

This entails that alternate frequency components of the 150 Hz tone will align harmoniously with the harmonics of the 100 Hz tone (300, 600, 900 Hz). Within the realm of music theory, the harmonic relationship between a 100 Hz tone and a 150 Hz tone is denoted as a "perfect fifth," an appellation rooted in a broader context beyond the scope of this discourse. Suffice it to acknowledge that harmonic frequency relationships wield significant influence across a multitude of global musical traditions.

4 Wave Propagation and Spherical Spreading

Sound waves in the air exhibit a propagation pattern where they disperse uniformly in all directions from their source. When the source dimensions are much smaller than the sound wavelength, a balanced distribution of sound pressure waves emerges in a spherical manner, constituting what is termed spherical wave propagation.

This phenomenon entails the dispersion of sound energy over the progressively expanding surface of the encompassing sphere, leading to a reduction in sound power in a given direction as the distance increases. In the absence of sound reflections, the surface area of the sphere theoretically grows proportionally to the square of the radius (surface area = $4\pi r^2$), consequently causing the wave intensity (measured in watts per unit area) to decline following a $\frac{1}{r^2}$ relationship.

Concomitantly, the acoustic intensity, which is proportionate to the square of the acoustic pressure, results in the sound pressure amplitude diminishing with a 1/r pattern. It's important to note that this analysis disregards any potential influence of sound reflections.

The consequence in practice is the commonly observed phenomenon that as the distance of observation from a sound source increases, its perceived loudness diminishes (Kinsler et al., 2000). This attenuation follows a 1/r relationship, leading to a reduction of 6 decibels in Sound Pressure Level (SPL) for each doubling of the distance. Mathematically, this can be expressed as

$$20\log_{10}(\frac{1}{r}\times\frac{P}{P_{ref}}) = 20\log_{10}(\frac{P}{P_{ref}}) - 20\log_{10} r \tag{2.5}$$

where a change in distance from 1 to 2 (doubling of distance) yields a decrease of approximately -6.02 dB, nonetheless, practical scenarios typically involve the presence of boundary surfaces, such as walls, the ground, and other physical obstructions, causing deviations from the simple spherical wave prediction due to the interaction between the direct sound path and the reflections from these surfaces.

4.1 Sound propagation and temperature

It is evident that the velocity of sound propagation within the air is contingent upon the air's temperature. Warmer air facilitates faster sound transmission, whereas colder air engenders slower sound propagation. Consequently, for a given frequency, the wavelength of sound waves elongates in warmer air, where velocity is elevated, and contracts in colder air, where velocity is diminished (Fig. 2.9).

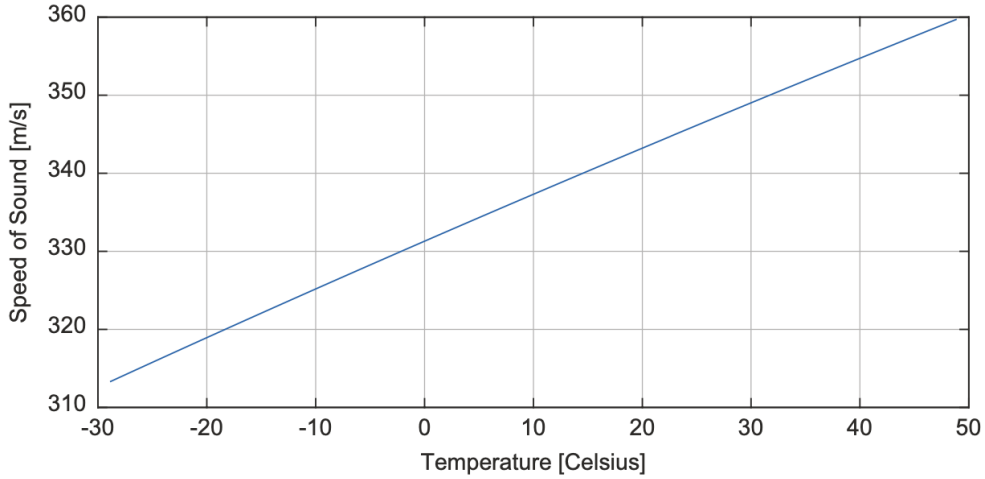


Figure 2.9: Speed of sound in air as a function of air temperature[164]

In the majority of instances, the temperature-dependent nature of sound velocity remains inconspicuous in practical scenarios, except when varying air layers possess distinct temperatures [125]. Disparate temperatures among air layers lead to differential sound velocities. For instance, during a chilly winter morning, a temperature gradient often prevails in the air, characterized by colder air proximate to the ground and warmer air at higher altitudes. Consequently, sound waves experience deceleration in the proximity of the ground and acceleration within the warmer upper air layers. This situation results in horizontal sound wave fronts undergoing downward refraction, wherein the segment traversing the colder near-surface air advances more sluggishly compared to the portion advancing through the warmer air above. This discrepancy induces a downward bending of the sound wavefront.

Conversely, during the early evening subsequent to a sun-warmed day, the temperature in the vicinity of the sun-exposed ground surface surpasses the temperature aloft.

Consequently, the sound wavefront will undergo upward refraction, induced by the higher velocity of the wavefront within the warm air layer proximate to the ground, in contrast to the segment of the sound wave advancing through the cooler, slower air at greater heights.

The outcome of this phenomenon (as depicted in Fig. 2.10) manifests as follows: during colder mornings, a distant sound might exhibit enhanced audibility owing to the refractive focusing effect. Conversely, when the air in proximity to the surface is heated, a distant sound's audibility might be diminished due to the upward curvature of the wavefront.

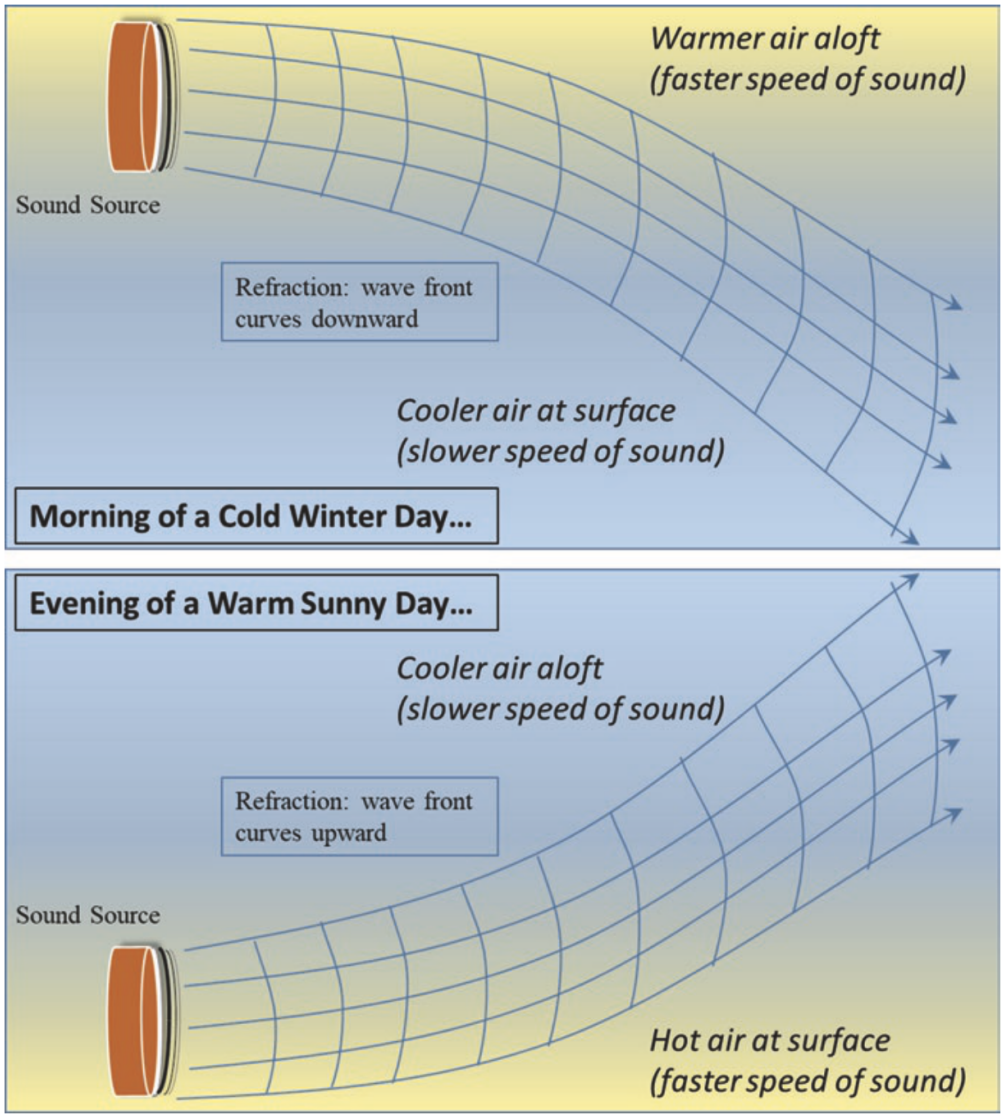


Figure 2.10: Refraction occurs when the air near the ground is colder, leading to a downward curvature of the sound wavefront. Conversely, if warmer air is near the ground while colder air resides aloft, the result is an upward curvature of the wavefront.[164]

In conjunction with the factors of spherical spreading and potential refraction, the process of sound propagation could entail energy losses attributed to air humidity and temperature [99]. These intricacies within the realm of sound physics hold

significance in the context of audio forensic analysis, particularly concerning outdoor sounds observed from substantial distances away from the source.

4.2 Reflections and Reverberation

A microphone registers the immediate acoustic pressure, encompassing sound waves that directly propagate from a sound source to the microphone's position, as well as waves that reach the microphone subsequent to bouncing off surfaces like the ground and walls, in addition to reverberant sound stemming from multiple surface reflections. As a result, an acoustic recording encompasses insights into both the sound source and the acoustic characteristics of the surrounding physical environment during recording. In cases where distinctive background sounds such as mechanical noises, music, or alarm signals are present, these secondary sounds are also captured by the microphone in addition to the more prominent foreground sounds.

The temporal discrepancy between the direct sound and its reflections relies on the variance in the travel path lengths between the sound source and the microphone. When there exists a line-of-sight path connecting the source and the microphone, the shortest path corresponds to the direct route (refer to Fig. 2.11).

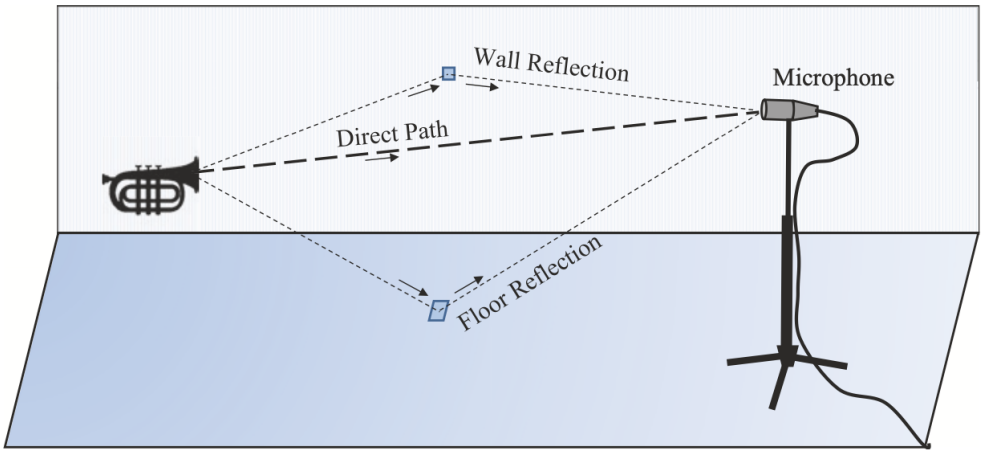


Figure 2.11: The sound directly emanating from a source and the initial reflections received by a microphone.[164]

Significant reflections arise when the reflecting surface is relatively distant from both the source and the microphone. Increased distance of reflection implies a notable temporal lag in the arrival of reflected sound in comparison to the direct sound, resulting in an audible echo. Conversely, if the source and microphone are in close proximity to the reflecting surface, such as being near the ground in an open area, the delay between the direct sound and its reflection is minimal, often imperceptible. Despite being undetectable to a human listener, such reflections can still be discerned within an audio recording, as will be expounded upon later in this text.

The knowledge of sound speed or its estimation renders the time gap between the arrival of the direct sound and the reflected sound a "measuring stick" for discerning the difference in path lengths. This type of information can aid in deducing event geometry in certain investigations, particularly in forensic contexts.

It should be noted that sound propagation within an enclosed space, such as a room, leads to the microphone capturing a combination of direct and reflected/reverberant sounds. In a spacious environment with a continuous sound source, like a lecture hall speaker or a musical ensemble, the energy of reverberant sound is generally distributed evenly. Since these reflections emerge from diverse directions, the sound field is characterized as diffuse.

In recordings where the microphone is close to the sound source, the dominant element is typically the direct sound from the source, overshadowing the reverberation. Conversely, when the microphone is moved farther away from the source, the background reverberation level remains relatively constant, but the sound pressure amplitude of the direct sound diminishes due to the 1/r effect associated with spherical propagation. As a consequence, the recording's equilibrium between the direct sound and room reverberation shifts from direct sound dominance to reverberation dominance as the source-to-microphone distance increases. This transition is shown in Figure 2.12.

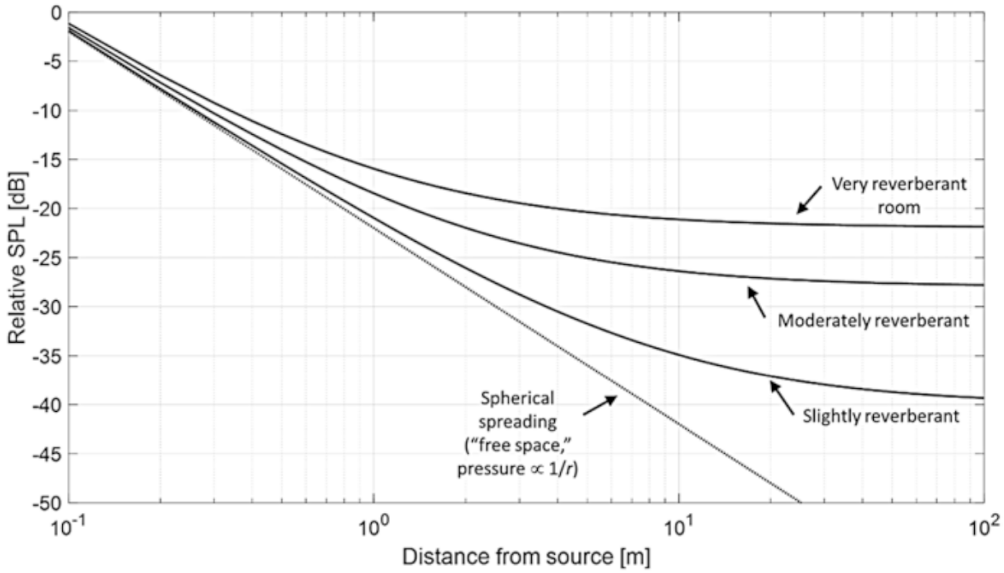


Figure 2.12: Alteration in sound pressure level as the microphone is progressively distanced from a continuous source within a room characterized by reverberation. In scenarios devoid of reverberation, the relative sound level adheres to the inverse 1/r trend (a decrease of 6 dB for each doubling of distance) attributable to spherical spreading. However, in the presence of room reverberation, the sound level converges with the background reverberation level as the microphone's separation from the source increases.[164]

4.3 Microphone Directionality

The directional attributes of a microphone contribute significantly to the recorded signal. Microphones generally respond to the acoustic pressure exerted on their diaphragms; however, microphone engineers may intentionally engineer devices to favour certain incoming wave directions or to suppress responses from other directions, aiming to mitigate unwanted background noises.

Three prevalent directional characteristics for microphones are omnidirectional, bidirectional, and unidirectional. These characteristics are often depicted through polar diagrams illustrating the relative sound capture across different directions the microphone is oriented towards (as exemplified in Fig. 2.13).

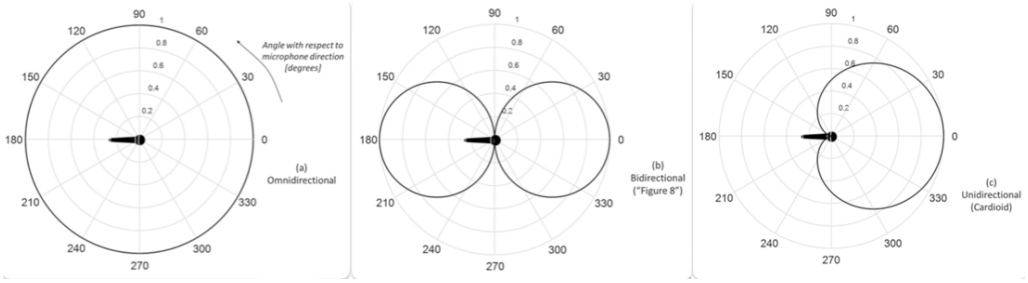


Figure 2.13: Polar diagrams depicting the directional characteristics of three prevalent microphone types: (a) omnidirectional, (b) bidirectional, and (c) unidirectional. These diagrams illustrate the variation in sound pickup relative to the angle concerning the microphone's pointing direction.[164]

The term "omni" signifies all, reflecting the omnidirectional microphone's capacity to capture sounds from all angles, resulting in a circular directional pattern. A bidirectional microphone predominantly captures sound from two directions: the forward direction (0°) and the opposite direction (180°), while attenuating sensitivity to the sides (90° and 270°). The directional pattern of the bidirectional microphone thus exhibits unity in the forward and backward directions and zero on the sides.

Conversely, a unidirectional microphone captures sound from one direction, which coincides with the microphone's pointing direction (0°). The design of the unidirectional microphone aims to minimize sensitivity to sounds originating from the opposite direction (180°).

The bidirectional microphone is sometimes referred to as a "figure eight" microphone due to its directional pattern resembling the numeral 8. The unidirectional microphone is commonly known as a cardioid microphone, as its directional pattern somewhat resembles a mathematical cardioid.

When a directional microphone is oriented towards a sound source (0°), the recorded signal's level tends to be higher compared to instances where the sound source is situated at an angle to the microphone, rendering the microphone less sensitive. Consequently, two distinct recorded sounds with varying levels in a forensic recording might originate from disparate sources or could potentially stem from the same source, assuming there was microphone or source movement resulting in a directional change.

Similarly, if a directional microphone is directed towards a sound source within a reverberant room, the recording equilibrium will emphasize the direct sound from the source in contrast to the reverberant room sound. This disparity emerges as the microphone's directional selectivity attenuates the reverberant sound arriving from off-axis directions, thereby favouring the direct sound originating from the source.

5 Human Hearing Characteristics

Audio forensic investigations often revolve around issues of audibility: determining whether a specific sound would have been perceivable by a human listener given the contextual circumstances. For instance, inquiries might arise regarding the audibility of an alarm signal at a particular distance or in the presence of known interfering noises. Addressing such queries necessitates an understanding of the capabilities and limitations of the human auditory system. The following two subsections offer a concise and simplified overview of (1) the anatomy and physiology of the auditory system, encompassing the ear and its neural connections to the brain, and (2) the subjective dimensions of hearing (psychoacoustics), encompassing the ability to identify a signal of interest amidst competing sounds and noise.

5.1 Anatomy and Physiology of the Ear

The auditory system is closely associated with the ear, which functions to convert sound energy into neural signals that undergo processing within specialized cerebral structures. The ear's fundamental anatomical divisions consist of the outer ear, middle ear, and inner ear ([125, 204]).

The outer ear constitutes the externally visible segment of the auditory system. The pinna, known as the external ear flap, encircles the entrance to the external auditory canal. Certain animals, like deer or cats, possess mobile pinnae that can be intentionally oriented in a specific direction. In humans, however, the pinnae are not typically movable in a practical manner, except through the rotation of the entire head. The term "concha" pertains to the central recess within the pinna, which connects to the external opening of the auditory canal.

The ear canal, slightly curved along its central axis, boasts a diameter of around 0.8 cm and a length of 2.5 cm. Positioned to the exterior of the head, the ear canal is exposed to external air, aligning its average pressure with the ambient air pressure. The innermost end of the auditory canal is hermetically sealed by the airtight and waterproof tympanic membrane (eardrum). This canal plays a role in safeguarding the eardrum and the delicate structures of the middle and inner ear, while simultaneously facilitating direct acoustic transmission of external sound.

The architecture and positioning of the external ear, coupled with the ear's orientation with respect to the head and upper body, result in acoustic diffraction that is contingent on the sound source's direction (both azimuth and elevation) as well as the sound's wavelength. Localizing a sound source within the azimuthal (left-right) plane predominantly hinges on binaural hearing—the ear's sensory mechanisms independently encode sounds before higher-order neural processing takes place.

When engaging with sounds in open spaces or when using circumaural (covering the ear) or supra-aural (on-ear) headphones, the auditory route from the concha to the eardrum comes into play. However, when employing insert headphones (earbuds), the acoustical pathway involving the concha of the external ear is not utilized.

The middle ear, positioned between the eardrum and the inner ear, encompasses three minuscule ossicles (tiny bones) and a tympanic cavity nestled within the

temporal bone of the human skull. These ossicles comprise the malleus (hammer), which connects to the eardrum's inner surface, the incus (anvil), and the stapes (stirrup), linked to the oval window of the inner ear (refer to Fig. 2.14).

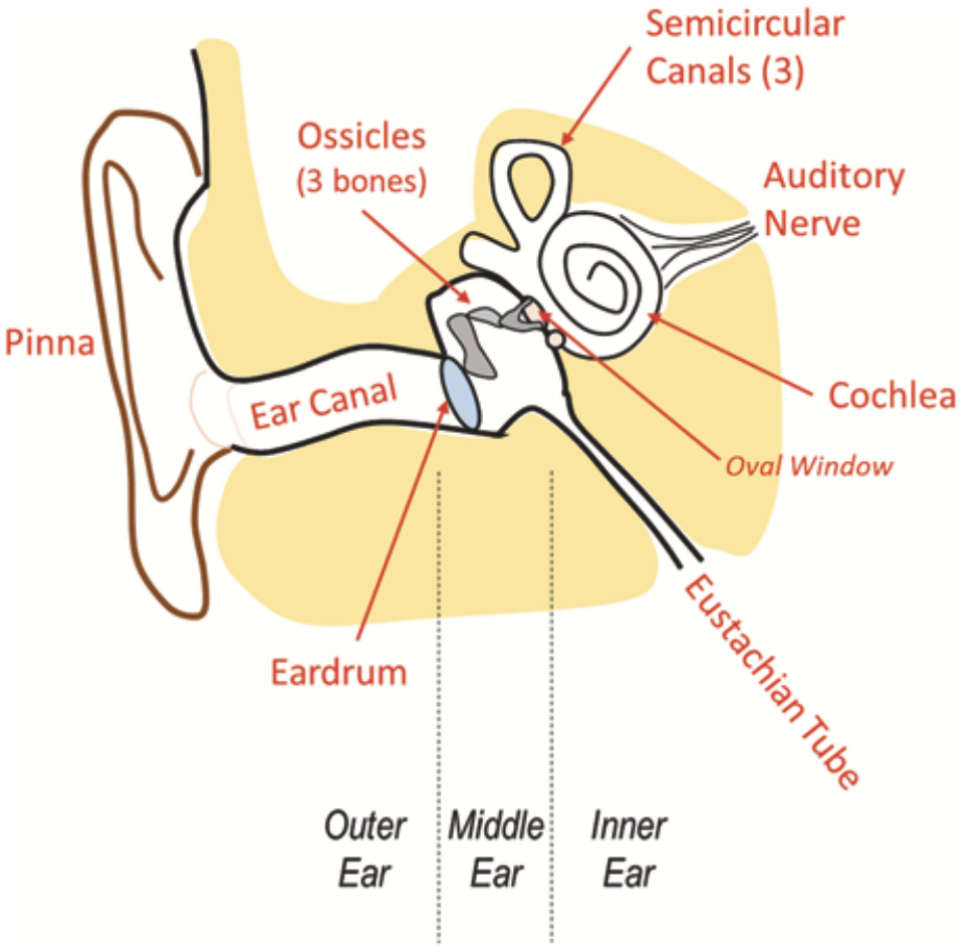


Figure 2.14: Human Hearing System in a simplified way[164]

The stapes, the tiniest and lightest bone within the human body, is situated within the inner ear's ossicular assembly. These ossicles are interconnected within the middle ear cavity by ligaments and two diminutive muscles—the tensor tympani and the stapedius. The tensor tympani is linked to the malleus (hammer), while the stapedius, measuring approximately 1 mm in length and representing the smallest skeletal muscle in the human body, is connected to the stapes. These auditory muscles contribute to a physiological response termed the acoustic reflex, which will be elucidated in the subsequent section.

The Eustachian tube, depicted in Figure 2.14, constitutes a conduit connecting the middle ear cavity with the rear section of the nasopharynx, which serves as the juncture between the throat and the nasal passage. Ordinarily, the Eustachian tube (for each ear) remains closed, briefly opening during swallowing to permit gradual air movement to and from the middle ear through this passage. In cases where a consistent air pressure discrepancy exists between the air confined within the middle ear and the air within the nasopharynx, the tube facilitates air passage, thereby equalizing the pressure on both sides of the eardrum. Instances of rapid ambient pressure alterations, such as swift ascents or descents during air travel, lead to accelerated air-

flow through the Eustachian tubes, generating the familiar sensation of "ear popping."

Typically, the air in the middle ear cavity maintains a pressure level roughly equivalent to the ambient atmospheric pressure. However, temporary blockages of the Eustachian tube, inflammation, or illnesses can cause air pressure variations within the middle ear cavity in relation to the surrounding atmosphere. This leads to an imbalance in pressure across the eardrum, causing its stiffening and influencing the mechanical sensitivity of the ossicular chain.

The inner ear encompasses the cochlea, functioning as the auditory organ, as well as three semicircular canals and associated structures that constitute the vestibular organ responsible for maintaining balance.

While not directly pertinent to audio forensic analysis, the three semicircular canals exhibit sensitivity to angular accelerations in three spatial dimensions, and smaller vestibular structures detect linear accelerations in relation to gravity. These motions are neurologically encoded to form the foundation for an individual's sense of balance, spatial orientation, and the capacity to harmonize physical movement with equilibrium.

The cochlea serves as the principal neurosensory organ within the auditory system. It takes the form of a spiral-shaped bony cavity, encapsulating and safeguarding the delicate biological tissues that respond acutely to sound-induced vibrations. Internally, the cochlea is divided into multiple fluid-filled chambers and minuscule neural structures. The stapes bone from the middle ear connects to the cochlea's oval window. Within the cochlear structure, microscopic hair cells within the organ of Corti are responsible for detecting sound-induced vibrations and translating them into neural signals. Roughly 3500 inner hair cells and about 12,000 outer hair cells are present in the human cochlea. Inner hair cells perform the neural transduction of vibrational stimuli, while outer hair cells are believed to serve as a cochlear amplifier and gain compressor. Neurons of the auditory nerve are linked from the base of each inner hair cell to specific locations in the brainstem.

The subsequent phases of processing along the auditory physiological pathway are briefly outlined in the remainder of this section. Since sound comprises small pressure fluctuations above and below ambient pressure, sound waves surrounding the head result in alternating pressure within the ear canals, which subsequently induces the eardrum to move in and out in response to these pressure variations. Oscillatory sound energy that displaces the eardrum induces motion within the ossicles, including the malleus. Consequently, the middle ear functions as a transducer, converting acoustic energy into mechanical energy via oscillatory transmission of forces and torques throughout the ossicular chain. In essence, the ossicular chain acts as a mechanical lever system, transferring the force from the comparatively large eardrum to the minute aperture of the oval window, thus engaging the fluid-filled cochlea's mechanical components.

A pivotal role of the middle ear is to efficiently transmit energy from air-based sound waves to mechanical displacement within the cochlea's sensory structures. While serving as an impedance transformer in terms of mechanical action, the middle ear notably concentrates sound energy from the larger area of the pinna to the

substantially smaller area of the stapes footplate. In summary, the external and middle ear structures alleviate, to a significant extent, the acoustic impedance disparity between air pressure and particle velocity and the corresponding mechanical displacements within the cochlea [7, 204].

The aforementioned account provides a succinct overview of the auditory pathway for one ear. This pathway exists in both of our ears, with specialized brain regions facilitating the combination of neural information across both ears for binaural processing. The higher-level processing within the auditory system encompasses nerves and structures that exchange information between the two ears, enabling us to estimate the direction and distance of specific sound sources relative to our heads. For those interested, more comprehensive explanations of auditory anatomy and physiology are accessible [84, 204].

5.2 Psychoacoustics

The human auditory system exhibits numerous notable strengths and limitations as a sound detector. Commonly acknowledged, the sense of hearing spans a frequency range from approximately 20 Hz to around 20 kHz under controlled laboratory conditions. However, the ear's sound detection capacity is contingent upon the pressure amplitude at a given frequency, stimulus complexity, and individual-specific factors. While this book does not delve into the broader scope, there are accessible and intriguing resources available that delve into the human auditory system, encouraging interested readers to explore this captivating domain [22, 173].

Audio forensic examiners generally do not extensively delve into the intricacies of human auditory physiology. Yet, on occasions, the perception of sound becomes pertinent in cases involving matters of audibility, intelligibility, speaker identification, and earwitness testimony (Koenig, 1986)[128]. Human psychoacoustics encompasses numerous fascinating facets. For the present context, we will focus on three: frequency sensitivity, frequency masking, and speech detection in noise.

In contrast to sound pressure level, which possesses a precise and objective definition, sound loudness is a perceptual attribute contingent upon the listener's perception. Rigorous tests involving human subjects demonstrate that subjective evaluations of sound loudness rely on both the frequency and amplitude of the sound perceived by the ears. Acousticians employ empirical equal-loudness contour charts like the Fletcher-Munson or Robinson-Dadson graphs to illustrate average sensitivity patterns. During these investigations, a substantial cohort of healthy young individuals were engaged in subjective loudness evaluations. Participants heard a sinewave tone at 1 kHz with a consistent sound pressure level and adjusted the loudness of a tone at a different frequency until it was perceived as equally loud as the reference 1 kHz tone. This process was replicated across a range of sound pressure levels for the 1 kHz reference tone, and the collective outcomes of all participants were averaged.

The resultant average response (refer to Fig. 2.15) demonstrates that generally, healthy young listeners require higher sound pressure levels for frequencies below 1 kHz to achieve a perceived loudness on par with the 1 kHz tone. Notably, the typical healthy ear is less sensitive to low-frequency sounds compared to tones in the 2–4

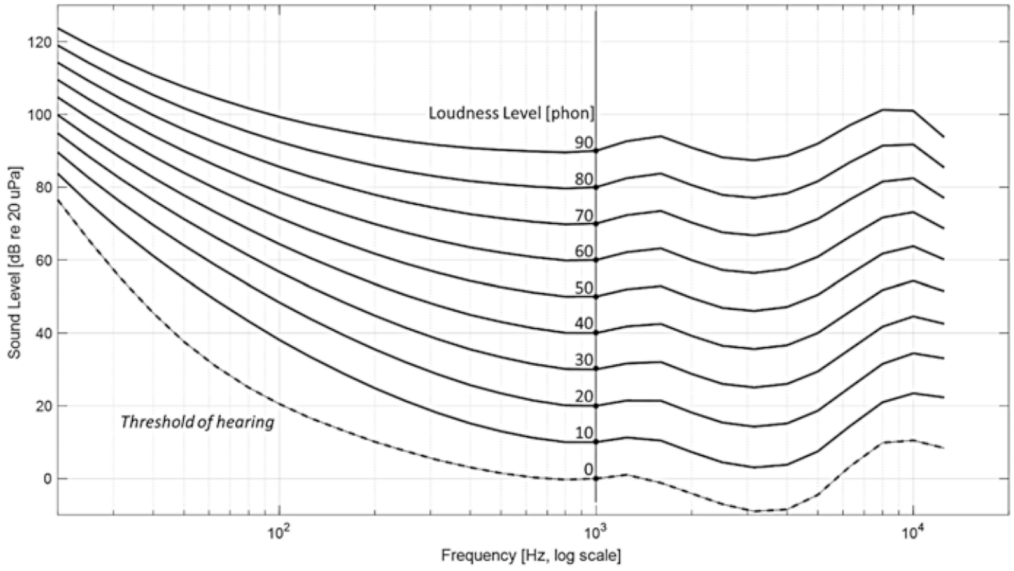


Figure 2.15: Equal-loudness contours for human hearing based on International Organization for Standardization standard 226:2003 (ISO 2003)[164]

kHz range (ISO, 2003). Optimal sensitivity is observed for sound frequencies around 3 kHz, aligning with the wavelength that triggers the auditory canal's tube resonance.

The human auditory system exhibits variations in sensitivity across different frequencies. The average ear displays reduced sensitivity at frequencies exceeding 4 kHz, progressively diminishing to little or no sensation beyond 20 kHz. Another noteworthy insight offered by equal-loudness curves is that sensitivity alterations are not only frequency-dependent but also amplitude-dependent. As the reference loudness at 1 kHz increases, the equal-loudness curves tend to become flatter, implying a more consistent sensitivity across frequencies. Consequently, when comparing louder tones across frequencies, their perceived loudness tends to be more uniform than when comparing quieter tones.

Beyond the frequency-related effects, temporal changes in sensitivity occur when the auditory system is exposed to high-level sounds. This results in the acoustic reflex, a neural response triggered by intense sounds like gunshots. The stapedius muscle contracts, modifying the coupling of the stapes footplate to the cochlea's oval window. This serves to shield the inner ear from potential damage. Despite its protective nature, the reflex's delay precludes immediate defence against abrupt, impulsive sounds. Hearing sensitivity's decline with increasing age is a widely observed phenomenon known as presbycusis. Individual differences, mainly arising from injuries, diseases, or neurological damage to the middle and inner ear structures, can also alter hearing characteristics.

In audio forensics, it is imperative to recognize that the ear operates as a nonlinear and time-varying detector. This consideration applies to both earwitness testimony and examiners using their ears to analyze audio evidence. Regular hearing screenings are recommended for forensic examiners to monitor changes in hearing acuity. Masking refers to the phenomenon where the presence of one sound makes it difficult for the ear and brain to perceive another sound presented simultaneously, especially if they share similar frequency content. While masking can be frustrating

in some situations, it is useful for estimating the human hearing system's ability to detect unwanted background sounds.

The masking effect finds utility in contemporary perceptual audio coding systems like MP3, AAC, and WMA, allowing for reduced bit usage while maintaining audio quality. However, forensic audio examiners must be cautious when interpreting the reconstructed signal's waveform and spectrum, as perceptual encoding might introduce inaudible features that could interfere with objective analysis.

For further elaboration on these topics, refer to Section 2.8.

5.3 Frequency Weighting in SPL Measurements

Due to the nonuniform sensitivity of the human ear across various frequencies, sound pressure level assessments commonly incorporate a filter approximating the ear's sensitivity. This filter, known as a weighting filter, accentuates (or "weights") sound energy within the frequency range where the ear is most responsive, while attenuating this emphasis in less sensitive frequency ranges. The resulting filter is termed a bandpass filter, allowing the transmission of the signal within a specific frequency range or band. The widely adopted A-weighting filter approximates the average equal-loudness curve for a 40 dB reference signal. Standard sound level meters generally feature an A-weighting option, while some may include additional weighting choices like C-weighting and an "unweighted" (flat) frequency selection. When utilizing a weighting filter for sound level measurements, the result should be specified, such as "the meter reading was 45 dBA re 20 μPa," with "dBA" indicating the application of the A-weighting filter (Kinsler et al. 2000)[125] (see Fig. 2.16).

5.4 Speech Intelligibility

Audio forensic analyses frequently involve the interpretation of audio recordings containing human speech. In certain scenarios, the task may encompass assessing the probability that a spoken statement was comprehensible given the conditions provided by a witness or established through other evidence. As a primary channel of communication, human speech has evolved to incorporate substantial redundancy, thereby enabling listeners to grasp a speaker's message even when confronted with concurrent sounds and noise. Linguistic structures contribute context and semantics, facilitating the listener's comprehension of the main idea of a statement without requiring a full understanding of each individual word. Nevertheless, noise has a tendency to disrupt the intelligibility of speech communication (Quatieri 2002)[210].

Noisy speech is often characterized by its signal-to-noise ratio (SNR) expressed in decibels. SNR estimation generally relies on assumptions regarding speech and noise levels. A 0 dB SNR indicates that the signal (speech) level matches the noise level, while a negative dB SNR signifies that the noise surpasses the speech in intensity.

Subjective evaluations of noisy speech intelligibility typically align with the pattern illustrated in Fig. 2.17.

Intelligibility, measured as the percentage of accurate transcription by a listener, remains nearly 100% for SNRs exceeding 10 dB but rapidly diminishes to essentially zero when the SNR deteriorates below -10 dB.

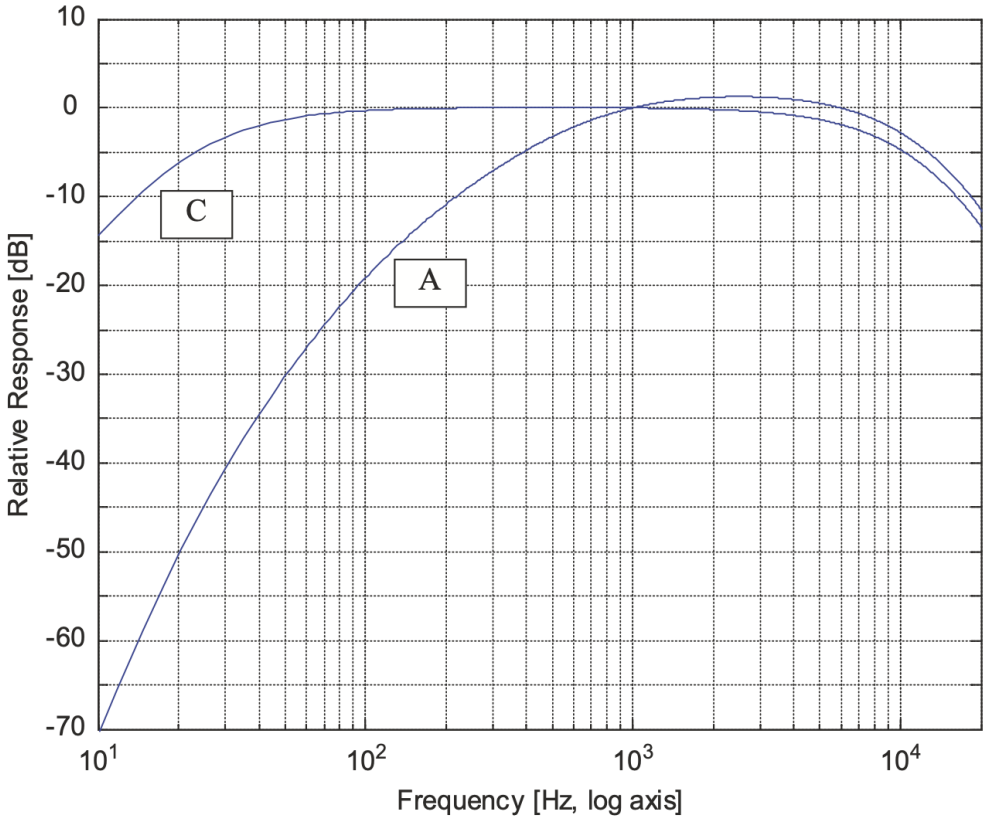


Figure 2.16: Equal-loudness contours for a human hearing based on International Organization for Standardization standard 226:2003 (ISO 2003)[164]

Human speech encompasses substantial signal energy within a bandwidth of approximately 200 Hz to 4 kHz. This bandwidth corresponds to the audio range transmitted by common telephones and mobile radio systems optimized for speech transmission. While widening the audio bandwidth generally enhances perceived speech quality, it doesn't necessarily enhance intelligibility, even if listeners perceive improved quality. This observation is crucial for audio forensic experts addressing the enhancement of noisy speech recordings: post-processing can sometimes lead to reduced speech intelligibility, even if listeners believe the quality has improved.

6 Signal Processing

Similar to the human ear, audio engineering systems capture air pressure fluctuations and transform acoustic energy into mechanical motion and electrical signals. Physicists and engineers use the term 'transduction' to describe this energy conversion process, with microphones and loudspeakers serving as audio transducers.

Microphones feature a diaphragm akin to the eardrum. Instantaneous air pressure on one side of the diaphragm, facing the sound source, differs from fixed air pressure on the other side. This difference generates a force that moves the diaphragm in response to sound pressure cycles. The diaphragm's motion activates a generating element, converting it into an electrical signal. Over time, audio engineers have developed various generating elements for microphones, such as variable resistance, electromagnetic induction, variable capacitance, and piezoelectric materials.

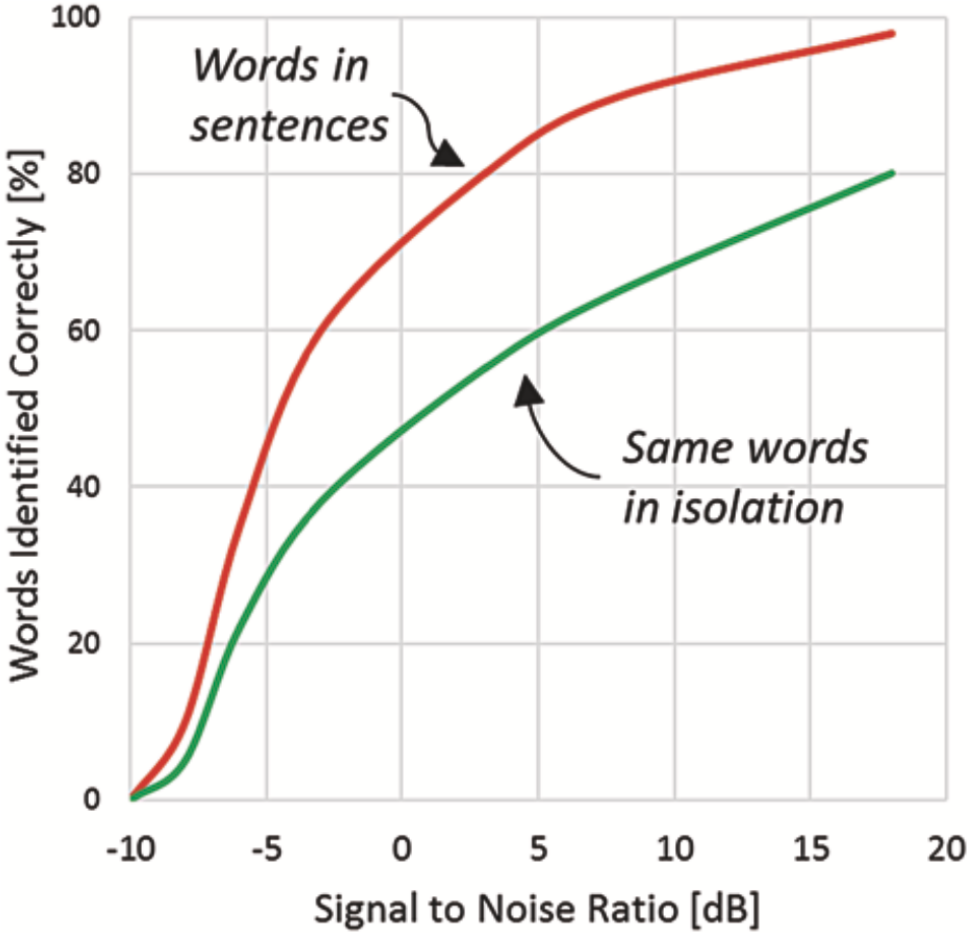


Figure 2.17: Equal-loudness contours for a human hearing based on International Organization for Standardization standard 226:2003 (ISO 2003)[164]

The microphone's electrical signal, produced in response to sound, is an analog signal. The continuous variation in electric output over time corresponds linearly to the continuous variation in the acoustic pressure wave affecting the diaphragm. This analog audio signal can undergo processes like amplification, filtration, recording, reproduction, modulation, broadcasting, and other electrical communication treatments.

Loudspeakers perform inverse transduction, converting analog electrical signals into sound. A typical loudspeaker driver includes a motor element generating force proportional to the audio signal, and a diaphragm effectively translating the motor's mechanical motion into acoustic waves. Loudspeakers often include a driver (motor and diaphragm) within a specifically designed resonant enclosure (cabinet) to enhance system linearity and efficiency. The enclosure, driver, and sometimes the amplifier in powered speakers are collectively designed. Modern loudspeakers often employ multiple drivers of varying sizes to optimize sound reproduction across the extensive range of wavelengths audible to humans (from wavelengths exceeding 17 m for 20 Hz to under 2 cm for 20 kHz)."

7 Digital Audio

In modern audio systems, digital signal processing and storage are prevalent, although microphones produce analog signals. The conversion to digital involves an analog-to-digital converter (ADC) performing two key processes. The first is time sampling, where the analog audio signal's instantaneous value is rapidly measured multiple times per second, creating time samples. The rate of time sampling is the sampling rate, measured in samples per second (Hz). The second process is quantization, where each waveform sample is represented by an integer value. Precision is determined by the number of digital bits used per sample; for instance, 16-bit quantization provides 65,536 values.

Consider the standard audio CD as an example. It has stereo channels, each sampled at a 44.1 kHz rate with 16-bit resolution per sample. Unlike analog signals, digital signals can be stored in memory, transmitted digitally, and safeguarded with error-correcting codes. Digital recordings allow perfect copying. However, it's vital to ensure digitization offers sufficient audio bandwidth through a fast sampling rate and ample amplitude precision through an adequate number of quantization bits. Mathematical theory dictates that the sampling rate should exceed twice the analog signal's bandwidth (Nyquist rate), accommodating the audible 20 kHz bandwidth. Quantization precision is determined by the required signal-to-quantization noise ratio (SQNR), which can vary.

SQNR stands for "Signal-to-Quantization Noise Ratio." It is a metric used to quantify the quality of a digitized signal, such as audio or other continuous signals, after they have been converted from analog to digital form through quantization. The SQNR measures the ratio between the strength of the desired signal (the original analog signal) and the unwanted noise introduced during the quantization process. A higher SQNR value indicates that the quantization noise is relatively lower in amplitude compared to the original signal, implying better signal quality and fidelity in the digitized representation. It is typically measured in decibels (dB) and is an important factor in determining the bit depth (number of quantization bits) needed for accurate digitization while minimizing perceptible noise in the signal. Telephone-quality speech might use 8 or 12-bit quantization (45–75 dB SQNR), while high-fidelity music often demands at least 16-bit quantization (>90 dB SQNR).

The complementary process, the digital-to-analog converter (DAC), reconstructs the analog signal from its digital form. This reconstruction usually occurs just before the power amplifier that drives speakers or headphones used for listening.

8 Perceptual Audio Coding

The conventional process of audio sampling, quantization, and reconstruction described earlier is effective but results in a high bitrate, which may be impractical for compact transmission and storage systems. Since the late 1980s, digital audio signal processing systems have harnessed the characteristics of human auditory perception to achieve excellent perceptual quality at significantly lower bitrates than traditional systems.

Perceptual audio coding techniques, like MP3 (MPEG-1, Layer 3), Dolby Digital,

and MPEG Advanced Audio Coding (AAC), capitalize on the masking effect in human psychoacoustics. These methods utilize lower bitrates while effectively masking the quantization noise during intervals with strong signal components.

While the reconstructed audio maintains good quality for human listeners, it's crucial to recognize that perceptual audio coding involves lossy compression. Unlike conventional digital audio systems, where discrepancies between the original and reconstructed signals are bounded by quantization levels, discrepancies in perceptually encoded signals can be considerably larger, even if they are inaudible to humans.

The field of audio forensics is increasingly dealing with recording systems that generate perceptually encoded audio.
Caution must be exercised when conducting waveform analysis on content encoded with lossy methods. Another concern arises when decoding a lossy-encoded signal and then subjecting it to subsequent lossy re-encoding.

Even if the same encoding/decoding algorithm is used, the repeated cycle of lossy compression, reconstruction, and compression again leads to the accumulation of audible artifacts and distortion.

Generally, it is advised not to equalize or re-encode perceptually encoded audio, as these processes alter the spectral characteristics exploited by perceptual encoding algorithms.

Chapter 3

Digital Audio Forensics

The first instances of audio files being scrutinized for forensic purposes can be traced back to the 1950s, following the advent of live recording systems not confined to the recording studio. By the early 1960s, the U.S. Federal Bureau of Investigation began cultivating expertise in audio forensics, specifically focusing on refining the clarity of speech, improving the quality, and validating the authenticity of recorded files[164, 300]. While the investigative benefits of tape recordings were evident, their legal admissibility raised questions. The surreptitious acquisition of recordings raised concerns about violating the accused's rights against self-incrimination. Moreover, issues like uncertainty in identifying voices and other details due to poor recording quality, the possibility of fakes, or alterations in the recordings further complicated the matter. These practical and legal challenges became crucial considerations in the field of audio forensics.

Digital audio forensic analysis consists of the acquisition, analysis and evaluation of audio recordings admissible to a court of law as evidence or for forensic investigations. Digital multimedia forensic analysis is commonly used to determine the authenticity and verify the integrity of the evidence submitted to court involving civil or criminal proceedings. The main objective of the audio forensic analysis process is to achieve one or more of the following tasks:

- **Authenticity**: The element of authenticity holds critical importance in these investigations, given that the significant deductions drawn by the investigator from the audio recording largely rely on the conditions in which the recording was carried out. If the recording has been intentionally or accidentally altered before the investigation, it could throw the entire examination into disrepute. Moreover, if there's an intentional or accidental error about the location or time of the recording, the examination could be rendered moot. Consequently, audio forensic examiners are required to validate the evidence's chain of custody, take measures to detect intentional tampering, and ensure safeguards against accidental modification.

- **Enhancement**: In cases involving audio forensic evidence, requests for audio enhancement are commonplace. Owing to non-ideal acoustic environments, many audio recordings of forensic relevance may face issues like poor microphone positioning, strong or fluctuating background noise, unclear enunciation by speakers, and weak signal strength. Under such circumstances, the audio data of investigative interest must be processed to emphasize the features of

interest. Enhancements become especially crucial when audio forensic evidence is to be presented in court, as most judges and jury members are not accustomed to listening and interpreting noisy audio, and lack the time to replay the material at different volumes. Given that courtroom audio presentations often don't happen under ideal conditions, the degree of enhancement must be chosen wisely.

- **Interpretation**: The interpretation of audio evidence can entail a range of forensic questions, from reconstructing event timelines to transcribing dialogues, and identifying unknown sounds. Questions addressed by audio forensic examination are usually founded on an investigator's hypothesis about a crime's circumstances, or in relation to other physical evidence and witness testimonies.

Despite potential shortcomings, such as the general difficulty in pinpointing the sound source's direction and orientation relative to the recording microphone (particularly if a single, monophonic recording is available), and the limited dynamic range of the recording, audio recordings provide several advantages for an investigation compared to film, video, and eyewitness observations. These include gathering information from all directions, rather than just within a specific field of view, and providing an objective, sequential timeline of events, as opposed to a witness's subjective recall.

1 Historical cases

1.1 McKeever Case

The McKeever case (United States v. McKeever, 1958)[1] is a notable legal case involving forensic audio in US federal courts. Defendants Thomas McKeever and Lawrence Morrison were indicted for extortion under federal anti-racketeering laws. McKeever secretly recorded conversations with the Ball Company's representatives to impeach a prosecution witness, George Ball. The court refused to admit the recording as evidence without establishing its authenticity, leading to the formulation of the Seven Tenets of Audio Authenticity. According to the court's statement (United States District Court 1958):

1. The recording device had the capability to capture the conversation now being presented as evidence.
2. The person operating the device possessed the necessary competence to operate it effectively.
3. The recording is genuine and accurate.
4. No alterations, additions, or deletions have been made to the recording.
5. The recording has been preserved in a manner that can be verified by the court.
6. The individuals speaking in the recording are clearly identified.
7. The conversation in question was conducted voluntarily and in good faith, without any form of coercion.

8. Within the forensic audio community, these seven criteria are informally known as the "Seven Tenets of Audio Authenticity."

These tenets include confirming the recording device's capability, operator competence, authenticity, absence of alterations, preservation manner, speaker identification, and voluntary, uncoerced conversation. The court recognized the increasing use of sound recordings as evidence and stressed the need for safeguards against fraud or abuse in this scientific development. Notably, the case highlighted the importance of technical aspects in tape recording and ensuring the reliability of audio evidence, which remains relevant today. It also emphasized the court's role in handling electronic evidence to benefit litigants while upholding rigorous standards for its admission.

1.2 McMillan Case

In the McMillan case of 1974, a federal narcotics conviction faced an appeal due to the use of audio recordings in the trial. Federal informant Beverly Johnson's telephone conversations were recorded by agents during her involvement in heroin trafficking. Some of these conversations involved suspect Harold McMillan in arranging heroin purchases. In the trial, the prosecutor played excerpts of the recordings and had an agent read the written transcripts to the jury. The defense objected, citing a lack of established authenticity and legal foundation. Similar to the McKeever case, the appeal's court upheld the fundamental principles of audio forensic admissibility and addressed concerns about authenticity and talker identification. The case highlights the importance of adhering to the Seven Tenets of Audio Authenticity and ensuring proper authentication and identification procedures for audio evidence in legal proceedings.

1.3 FBI Procedures

In the early 1960s, the US Federal Bureau of Investigation (FBI) initiated audio forensic analyses and enhancements. Building upon the McKeever tenets, the FBI devised a 12-step procedure for processing audio recordings [129]. These steps include

1. Evidence marking.
2. Physical inspection.
3. Recorded track position and configuration.
4. Azimuth alignment determination.
5. Playback speed analysis.
6. Proper playback setup.
7. Overall aural review.
8. Overall FFT review.
9. Setup of enhancement devices.
10. Copying process.

11. Work notes.

12. Reporting complete the procedure.

Steps 3 to 6 focused on challenges related to analog magnetic tape recordings, which were the prevalent recording medium during that era.

1.4 The Watergate Tapes

In 1972, the discovery of duct tape on a basement access door at The Watergate Hotel in Washington, D.C., by security guard Frank Wills led to the apprehension of five burglars in the Democratic National Committee offices. This seemingly isolated event set off a chain of events that exposed a larger conspiracy involving the Nixon reelection campaign and White House officials.

Amid suspicions of a cover-up, White House aide Alexander Butterfield's testimony before the Senate Committee revealed the existence of audiotape recordings of conversations between President Nixon and his advisors. These recordings were made secretly using audiotaping systems installed in various locations, including the Oval Office and the Cabinet Room. [149]

A crucial tape recording from June 20, 1972, which possibly contained discussions about the Watergate cover-up between Nixon and Haldeman, came under scrutiny. Investigators found an 18 1/2-minute gap in the recording, raising suspicions of deliberate erasure to destroy incriminating evidence. Chief Judge John J. Sirica determined that forensic study was necessary to assess the potentially altered tape.

A special Advisory Panel on White House Tapes, comprising technical experts, was formed to conduct a systematic analysis. They examined the physical and mechanical aspects of the tape, looking for signs of alteration or damage. Critical listening and signal processing techniques were employed for intelligibility enhancement.

The panel's May 1974 report concluded that magnetic erasures caused the 18 1/2-minute gap and identified overlapping erasures from a different tape recorder than the original. The "smoking gun" recording, released by the US Supreme Court, revealed President Nixon's attempt to obstruct justice, leading to his resignation on August 8, 1974, to avoid impeachment and removal from office. The Watergate Tapes became a crucial milestone in audio forensic analysis, demonstrating the importance of authenticity assessment in legal proceedings.

1.5 Reevaluation of the Assassination of President Kennedy

The assassination of President John F. Kennedy on November 22, 1963, in Dallas, Texas, has been a subject of extensive interest, scrutiny, and speculation. The Warren Commission's official finding was that Lee Harvey Oswald fired three shots from a sixth-floor window of the Texas School Book Depository. However, inconsistent earwitness testimonies about the number of shots and their direction complicated the investigation.

The Zapruder film, though silent, provided critical evidence regarding the timing of gunshots and the injuries sustained by the President and Governor Connally. Investigators attempted to use audio recordings from Dallas Police radios to gather sound from Dealey Plaza during the assassination. The audio from two radio channels was recorded using the dictabelt and audograph machines. A theory emerged that an "open microphone" recording from a malfunctioning police motorcycle could contain the sound of gunshots.

In 1978, the House Select Committee on Assassinations engaged experts from Bolt, Beranek, and Newman (BBN) to analyze the dictabelt recording. BBN concluded that there were three gunshots from the Book Depository and a likely fourth shot from the "grassy knoll" area, suggesting a second gunman and conspiracy. This finding was corroborated by an independent analysis performed by Mark R. Weiss and Earnest Aschkenasy.

However, subsequent investigations raised questions about the acoustic evidence, the location of the open microphone, and the methodology used to assess the findings. The US Justice Department requested a review by the National Academy of Sciences, which challenged the earlier conclusions. Moreover, private citizen Steve Barber identified cross-talk in the dictabelt recording, raising further doubts about the grassy knoll theory.

Despite numerous scientific examinations and rebuttals, debates about the Dallas dictabelt evidence persist to this day. The case remains an enduring subject of investigation and analysis.

1.6 Talker Identification and “Voiceprints”

The term "voiceprint" was first used in Bell Telephone Laboratories publications in 1944. Lawrence Kersta of Bell Labs published a paper in 1962 proposing that speech spectrograms, based on individual dimensions of the talker’s oral, pharyngeal, and nasal cavities, could be used for voice identification, akin to fingerprinting. Initial testing showed promising results [260].

During the 1960s and 1970s, the aural-spectrographic method emerged in audio forensics for comparing an unknown talker’s spectrogram with known talkers. Examiners used critical listening and visual comparison of spectrograms to make determinations. They would report five possible opinions: positive identification, probable identification, no decision, probable elimination, or positive elimination.

However, concerns about the reliability and dependability of this technique emerged. Studies challenged assumptions of spectrographic uniqueness and time-invariance of speech, raising doubts about false identification or elimination [29, 30, 31].

In response to ongoing controversies, the FBI requested a special panel from the National Research Council to study the scientific principles and reliability of aural-spectrographic voice identification in 1976. The panel highlighted technical uncertainties and recommended approaching forensic applications with caution. They

suggested clearly explaining the method's limitations in testimony before judges or juries [32].

Recent discussions of the aural-spectrographic method still echo these concerns, emphasizing the need for caution when utilizing voice identification evidence.

2 Audio Examiner role

The audio forensic examiner plays a crucial role in court proceedings by providing an objective and scientifically grounded understanding of the nature and reliability of the audio evidence. Their primary responsibility is to educate the court, ensuring that all involved parties have a clear understanding of the audio evidence presented. The examiner refrains from taking any side in the legal process and remains impartial throughout the examination.

During their testimony, the audio forensic examiner addresses three main aspects of the audio evidence: the facts, the methods used for analysis, and the interpretation of the findings. This comprehensive approach ensures that the court receives a well-rounded perspective on the audio evidence's credibility and relevance.

After the audio forensic examination is requested by a law enforcement organization or an attorney, the examiner may encounter various scenarios. They may need to determine the availability of the original audio recording or work with a duplicate if the original is unavailable. The examiner may assess the circumstances under which the recording was made to understand its context better.

An essential part of the examiner's role is evaluating the quality of the audio recording, categorizing it as either good, marginal, or poor. Additionally, they may need to address any disputes or concerns about the authenticity of the recording that may have arisen during the investigation.

Sometimes, prior audio forensic examinations have already been conducted, and the examiner must identify the reasons for requesting further analysis. Lastly, the examiner focuses on the specific audio forensic questions raised by the parties involved, which helps direct their analysis and ensure that the court receives the most pertinent information.

3 Audio Examination procedure

In the realm of forensic analysis, the challenge lies in maintaining impartiality during the interpretation process. Within audio forensic examination, bias can often stem from external non-audio information linked to a case, suspects, circumstances, and investigator's suspicions. Such information, originating outside of the audio evidence, might include the arrest history of a suspect, physical evidence details, preferred conclusions, or potentially incriminating comments by involved individuals. While these details might be relevant to the court or jury, they can also prejudice the audio forensic examination. This extraneous information has the potential to consciously or unconsciously influence the examiner's work.

As previously mentioned, the role of the forensic examiner is to provide the court with an objective understanding of the audio evidence's nature and reliability from a scientific perspective. The examiner is neither an advocate for a specific side in the legal process nor a participant in the adversarial proceedings. Instead, they are

an expert who testifies solely about the presented audio evidence. The examiner's testimony covers the facts, methodologies, and interpretations related to the audio evidence, leaving law enforcement and attorneys to weave together different evidence pieces to support their case theories.

The process of audio forensic examinations usually starts with a request from law enforcement or an attorney. The requester's familiarity with audio forensics varies, making a checklist valuable. This checklist should address factors such as the availability and quality of the original audio recording, circumstances of recording, disputes regarding authenticity, prior examinations, and specific questions requiring audio forensic analysis. It's important that analysis requests align with the examiner's expertise, and engagements exceeding their knowledge level should be declined.

Comprehensive notes and documentation for forensic engagements are essential. These records should be detailed enough to recollect requests and processes over extended periods. Documentation should be comprehensive enough that another examiner could understand the procedures and conclusions without ambiguity.

Beginning with the original recorded media and, if possible, the original recording system, while creating verified digital copies, is highly recommended before engaging in enhancement or interpretation. The original recording system can provide valuable settings, data, timestamps, and other relevant information. For devices with special cables, connectors, and power supplies, these details should also be communicated.

Certain recording devices possess volatile memory, losing recorded content in case of power loss. It's crucial to safeguard this memory from potential power disruptions. The examiner should advise the sender to use "write protection" and other overwrite prevention mechanisms.

Formal laboratory protocols dictate the standard procedure for audio forensic examinations, specifying the required evidence and accompanying information. This includes the original recording or an exact digital duplicate, equipment details, maintenance records, recording methods and circumstances, previous reports, transcripts, investigator notes, and more (Scientific Working Group on Digital Evidence 2008).

Upon receiving the audio forensic evidence or equipment, the audio forensics examiner must adhere to laboratory standard protocols (Audio Engineering Society 1996). These protocols encompass several key practices:

- **Ensuring the chain of custody**: Document the date and circumstances of evidence receipt, and maintain secure handling throughout the review process to prevent potential damage or loss.
- **Observing data carrier details and metadatae any signs of damage like cracks, marks, scratches, etc.**: Employ photographs and written notes to comprehensively document all submitted materials, including packaging specifics, model and serial numbers, formats, and more. Not
- **Initial labeling nondestructively** : Abide by laboratory guidelines for uniquely marking evidence for future identification. Some labs use case numbers and date marks, while others rely on the examiner's initials and date. Special

caution is necessary when marking items like CD/DVD media to prevent harm. If marking isn't feasible, store the data carrier in a suitable sealable container and mark the container.

- **Utilize a verified digital copy, reserving the original only for necessary cases**: When dealing with analog evidence, generate a high-quality digital copy from the analog original. This may involve locating appropriate playback equipment, aligning it with the tape, and ensuring the tape's integrity to prevent damage during playback. Seeking assistance from an analog specialist is recommended in such scenarios.
- **For digital audio evidence, create direct digital "bitstream" copies that are verified:** Ensuring that the copying process preserves the original content is crucial. Many digital forensics labs employ hardware write blocker devices between the storage device and control computer. These blockers intercept any commands that could modify storage contents, thus preserving the integrity of the material.

3.1 Fundamental tools

Contemporary audio forensic examination relies on a set of fundamental tools: a high-quality audio playback system, a waveform display program, and a spectrographic display program. These essential functions are typically executed on standard desktop or laptop computers.

Audio Playback System

The audio playback system employed must possess both quality and versatility that surpass the frequency range and dynamic range of the forensic audio material under scrutiny. In simpler terms, any limitations in audio quality should be attributed to the original recording and not to issues with the playback system.

The computer's integrated audio subsystem, soundcard, or externally connected USB converter must offer support for a wide range of sampling rates and formats. Additionally, it should be equipped with the necessary audio format decoding and reconstruction software modules required to handle the native format of the audio evidence. Typically, reliable manufacturers of professional general-purpose recording studio monitors supply suitable loudspeakers. A reasonable benchmark for the stated frequency response is around 50 Hz to 20 kHz.

For many audio forensic tasks, it is advisable to use headphones. They help mitigate the influence of room reverberation, computer fan noise, and other audible distractions that may be present in the playback environment. When selecting headphones, opt for professional-grade ones with comfortable earpieces that create a complete seal around the ears. Ensure that the playback system includes a separate volume control knob for the headphone system.

While there might be a temptation to increase the sound level when attempting to discern potentially significant sounds in low-quality audio forensic recordings, it is crucial to avoid elevating the volume to a point where it induces decreased sensitivity in the ears (known as the acoustic reflex). Furthermore, listening to the recording

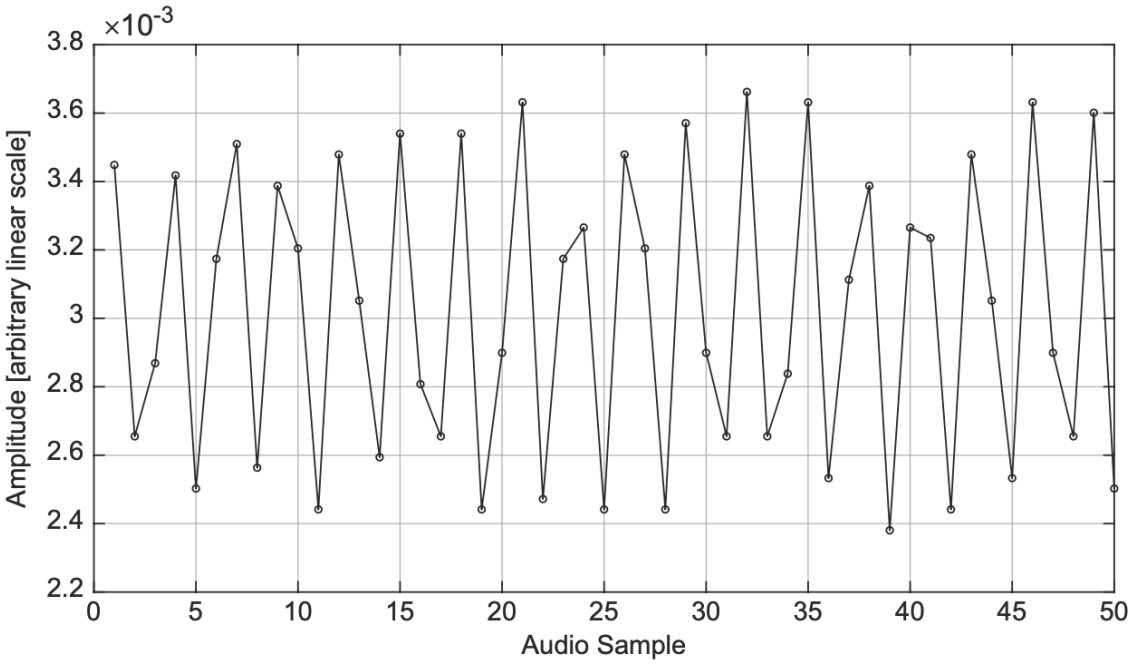


Figure 3.1: Digital audio display of a time span sufficiently short to show individual samples, with "connect-the-dots" lines between the sample points

should be done in a manner that guards against unexpected loud sounds that could be harmful to the ears.

Waveform Visualization

In audio forensic analysis, aural interpretation is fundamental, but visual aids can significantly assist in the process. One crucial visual tool is the graphical waveform display, which represents an audio recording as a graph with time on the horizontal axis (abscissa) and amplitude on the vertical axis (ordinate). These waveform display programs offer the capability to view specific time intervals and provide controls for zooming in or out on both the time and amplitude axes.

Typically, the graphical display represents individual waveform samples as dots when dealing with very short time intervals. Some programs employ a "connect-the-dots" approach, creating lines between these sample points (fig. 3.1). In cases where there are more samples than horizontal pixels on the screen due to longer time intervals, most display programs show the maximum and minimum sample amplitudes within that short time span, effectively outlining the audio signal's envelope (fig. 3.2).

The most valuable waveform display programs offer simultaneous audio playback, allowing users to select playback start and stop positions using cursors or select-and-drag highlighting. This iterative process enables a combined auditory and visual assessment of waveform details.

While these programs often include features like waveform editing, format conversion, and audio effects processing, it's critical to safeguard and preserve the original reference copy of the audio. Special care should be taken to avoid unintentional alterations during the viewing and initial assessment stages.

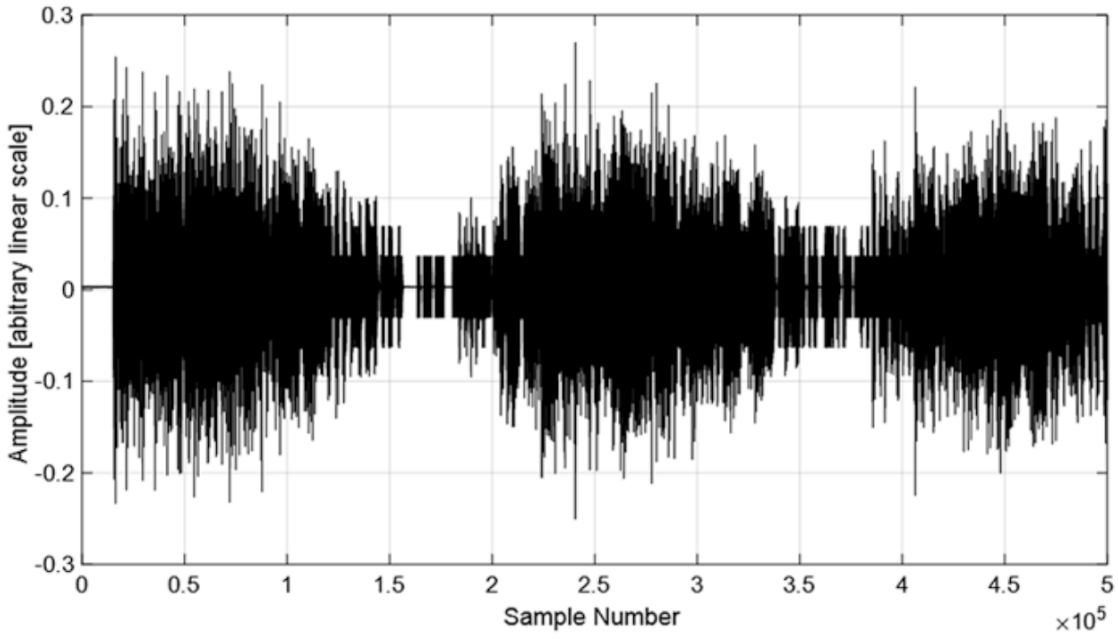


Figure 3.2: Waveform display of a time span that is too long to depict every individual sample: the display shows the signal envelope

A significant concern arises when dealing with encoded audio files, such as MP3. To facilitate viewing and listening, the display program must decode the MP3 file into standard pulse-code modulation (PCM) samples. However, caution must be exercised when editing such files and subsequently saving them as MP3 again, as this process re-encodes the PCM samples into MP3 format, effectively generating a second encoding cycle. It's crucial to recognize that perceptual coders like MP3 are lossy, meaning that each decode/re-encode/decode iteration tends to accumulate audible distortion due to the lossy encoding steps.

As previously emphasized, it is imperative to avoid the practice of decoding, modifying, and then re-saving the edited file in an encoded format. This ensures the preservation of audio quality and minimizes the introduction of additional distortions that may compromise the forensic analysis process.

Spectrographic Visualization

A valuable method for visually representing audio forensic recordings is the spectrogram. The spectrogram is a specialized graph created by computing the magnitude of the short-time Fourier transform (referred to as the spectrum) for consecutive, brief time intervals within the input signal and displaying these sequentially across the screen. This process involves selecting successive short blocks or frames from the audio signal recording, as illustrated in Fig. 3.3.

Similar to the waveform display, the spectrogram offers a graph of audio signal energy, with the horizontal axis representing time. However, unlike the waveform display, the vertical axis of the spectrogram is the signal's frequency scale in hertz. The spectrogram's color or brightness at specific time and frequency coordinates within the graph indicates the relative amount of audio signal energy.

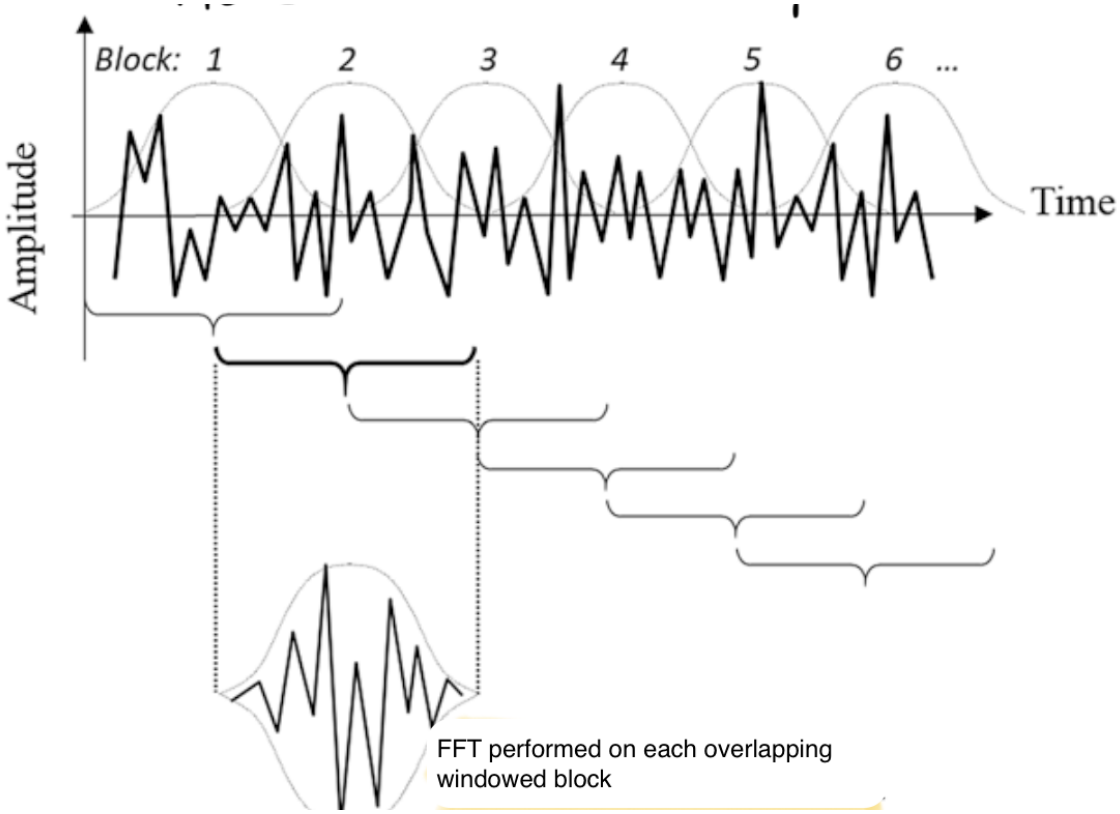


Figure 3.3: The concept of the Short-Time Fourier Transform (STFT) involves the segmentation of the audio signal into overlapping blocks or frames. Subsequently, the Fast Fourier Transform (FFT) is applied to calculate the short-time spectral magnitude of each individual block [164].

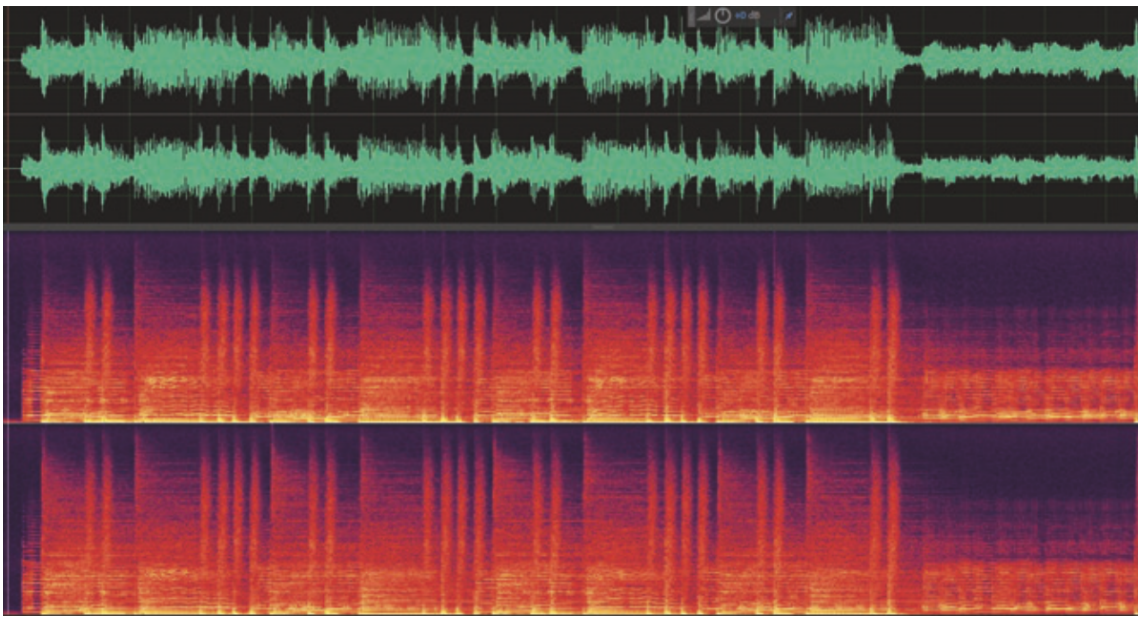


Figure 3.4: The provided image illustrates a combined time domain and spectrographic display of a stereo (2-channel) audio recording featuring a rock-and-roll instrumental combo composed of electric guitar, bass, and drums. The total duration of the recording is 10 seconds, with consistent time scales across all four displayed panels. The frequency range in the lower two panels, represented on the vertical axis, spans from 0 to 20 kHz, utilizing a logarithmic scale. The upper two panels, characterized by a light green hue, portray the time domain envelope, which denotes signal amplitude against time, for both the left channel (top row) and right channel (second row). Meanwhile, the lower two panels, marked with an orange hue, depict the spectrograms of the left and right channels, respectively. In these spectrogram panels, the vertical axis represents frequencies from 0 to 20 kHz, while the horizontal axis represents time. Brightness of colors in the spectrograms corresponds to spectral energy: darker-colored pixels represent less energy at the respective time and frequency, while brighter-colored pixels indicate higher energy at the corresponding time and frequency. Notably, there is a recurring pattern of vertical reddish bars in the spectrogram, attributed to drum hits, and horizontal yellow stripes at lower frequencies, originating from harmonics of the electric guitar and bass lines [164].

For this reason, the spectrogram is often referred to as depicting the signal in the frequency domain, while the waveform display portrays the signal in the time domain, as depicted in Fig. 3.4. The upper section of the display exhibits the time waveform envelope for both stereo channels, while the lower section presents the spectrogram for each channel.

In the spectrographic view, impulsive sounds like clicks or gunshots manifest as vertical lines, signifying energy across a range of frequencies (broad along the vertical axis) but of brief duration (short along the horizontal axis). Conversely, a whistle or a continuous hum tone appears as a horizontal line, indicating that the sound energy is relatively concentrated in its frequency range but persists over a longer duration.

Spectrographic display programs enable users to specify both a time range and a frequency range. Nevertheless, it's crucial to comprehend the fundamental mathe-

matical trade-off between signal resolution in time and frequency. Zooming in on a very brief time segment of a signal inherently sacrifices fine frequency resolution, while zooming out to encompass a longer time duration offers better frequency detail but diminishes temporal precision. In essence, the spectrogram presents a trade-off between the selectivity of display for separating signal components of similar frequencies and the level of detail in timing (Fig. 3.5).

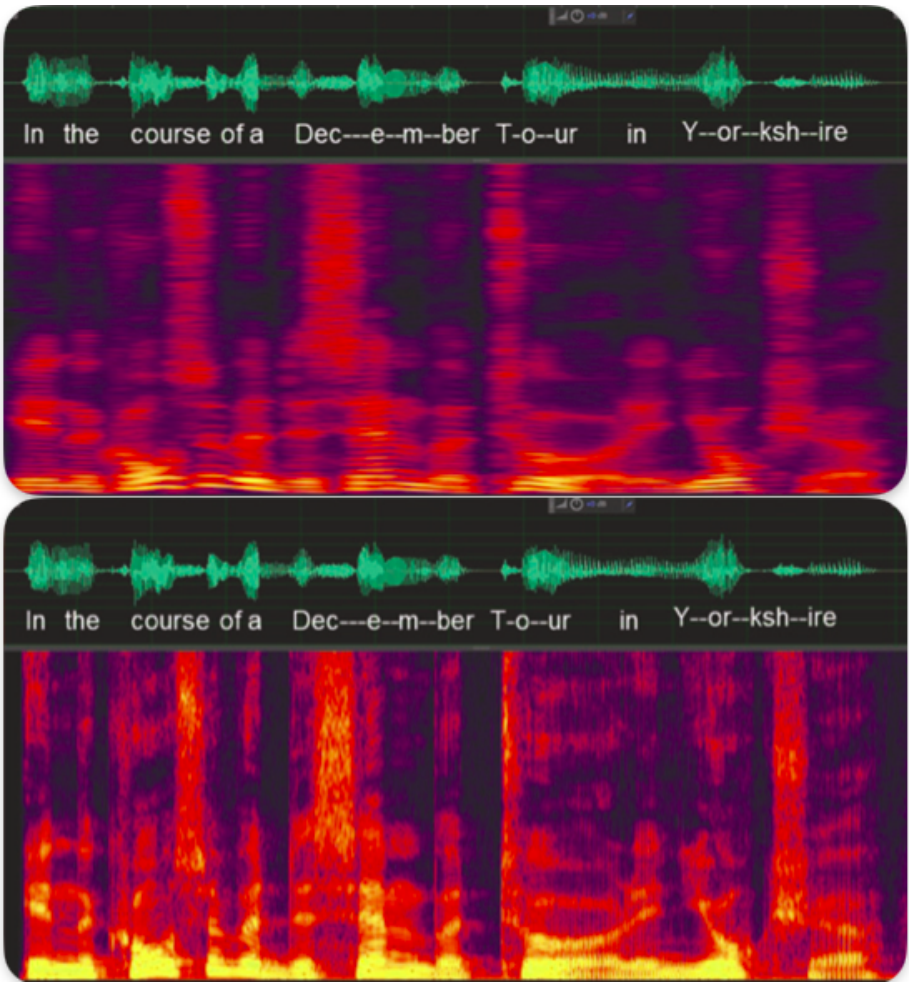


Figure 3.5: The provided images display two spectrograms of the same speech utterance, delivered by a male speaker, thereby illustrating the fundamental trade-off between time and frequency resolution. In the upper frame, longer time block lengths are employed, resulting in improved resolution in frequency, allowing for a more detailed representation of harmonic partials. However, this enhancement in frequency resolution comes at the expense of clarity in rendering the sound's attack and release characteristics, which appear somewhat blurred in the spectrogram. Conversely, the lower frame utilizes shorter time block lengths, providing superior resolution in time, which highlights the "edges" occurring when the signal undergoes changes. However, this advantage in time resolution comes at the cost of reduced frequency detail. The overall duration of the audio segment depicted in both frames is 2.5 seconds, with the frequency range spanning from 0 to 10 kHz, utilizing a linear scale for reference [164].

3.2 Initial Aural Evaluation

The primary step in an audio forensic assessment involves listening to the verified work copy of the audio material. Conduct this initial listening session in a quiet environment, setting the playback volume to a comfortable level. If the playback area is devoid of distractions, using loudspeakers for this purpose is satisfactory. During this initial comprehensive auditory review, it's customary to jot down preliminary notes about the audio material. These notes should encompass initial impressions regarding quality, any noticeable defects, or audible occurrences within the recording.

Many forensic examiners may additionally opt to examine successive spectrograms of the recording, employing appropriate time and frequency ranges. Spectrograms often aid in identifying subtle signal attributes and background sounds in the recording, facilitating further evaluation. Following the preliminary listening and spectrogram analysis, the examiner then addresses the audio forensic queries raised by the inquiring party. The essential set of analysis techniques comprises attentive listening, waveform analysis, and spectral analysis.

3.3 Critical Listening

Known as critical listening, this method involves attentive evaluation of the forensic recording. Critical listening sessions require a quiet environment without distractions, preferably using quality headphones. Playback levels are maintained moderately to prevent auditory fatigue and minimize triggering the acoustic reflex. The critical listening process is iterative, involving repeated playback of significant sections. Many examiners choose to use waveform display software during critical listening, making it easier to add time markers and annotations.

An essential aspect of critical listening is deliberately focusing on foreground sounds, such as speech dialog. Subsequent replays shift the focus to background sounds, including ambient noise, distant conversations, and subtle noises. In specific situations, background sounds can help identify the recording's context, while anomalies in background sounds might indicate editing or alterations.

However, examiners must be cautious when repeatedly listening to a short looped segment to avoid forming a perception based on the loop's rhythm rather than the actual audio evidence.

3.4 Waveform Analysis

The ears excel at detecting and recognizing sounds, but they may struggle with precise time and amplitude measurements. Visual aids, like waveform display programs, offer a graphical representation of the audio signal. This display assists in identifying audible events, time intervals, signal variations, and other signal characteristics.

Forensic examiners commonly employ waveform displays to start with a wide time range, potentially spanning several minutes, to gain an overall understanding of the signal's waveform. The approach then involves progressively zooming in on specific time intervals for closer examination, accompanied by note-taking and preliminary observations. Significant signal elements relevant to the investigation, such as specific utterance times or distinct background sounds, receive particular attention during

this phase.

An effective strategy combines visual identification of signal features with auditory analysis. While zoomed in, the examiner should scrutinize the signal for any anomalies like discontinuities, dropouts, sudden clicks, or waveform irregularities. These irregularities could indicate issues with the recording system or the potential manipulation or deletion of content.

3.5 Spectral Analysis

In addition to the time domain waveform display, examining the spectrogram can aid in identifying significant signal attributes. With practice, one can extract crucial signal characteristics and changes from the spectrogram and then cross-reference with the corresponding audio signal.

Considering the trade-off between time and frequency inherent in the spectrogram, the examiner might alternate between different frequency and time resolution settings. Adjusting the analysis block length offers a better indication of when a sonic event occurred in the spectrographic display. However, increasing the block length enhances frequency resolution but reduces time resolution, potentially obscuring the start and end of sound events.

Another user choice in spectrographic display software is the window function. This involves applying an amplitude weighting that smoothly fades in and out the short-time audio block for each spectrographic segment, preventing abrupt spectral effects from starting or stopping the data block. Various amplitude window functions, like triangular, Bartlett, Hann, Hamming, Kaiser, Blackman-Harris, and others, can be utilized. If no tapering is applied, the implicit window is known as "rectangular."

While the amplitude window addresses abrupt boundaries, it also slightly diminishes spectral resolution. The exact shape of the amplitude window function has nuanced impacts on frequency resolution, prompting experimentation with different window functions and block lengths to visualize relevant spectrographic details in a specific investigation (Fig, 3.6).

Some display programs offer simultaneous presentation of the time waveform, spectrogram, and audio playback, enhancing critical listening and visual assessment of signal characteristics. This feature is highly recommended.

As previously emphasized, thorough and detailed work notes should be maintained during the auditory and visual assessment. Given the potential time lapse between initial evidence observation and subsequent steps like report writing and testimony, recording even seemingly obvious details is crucial for future reference, rather than relying solely on memory.

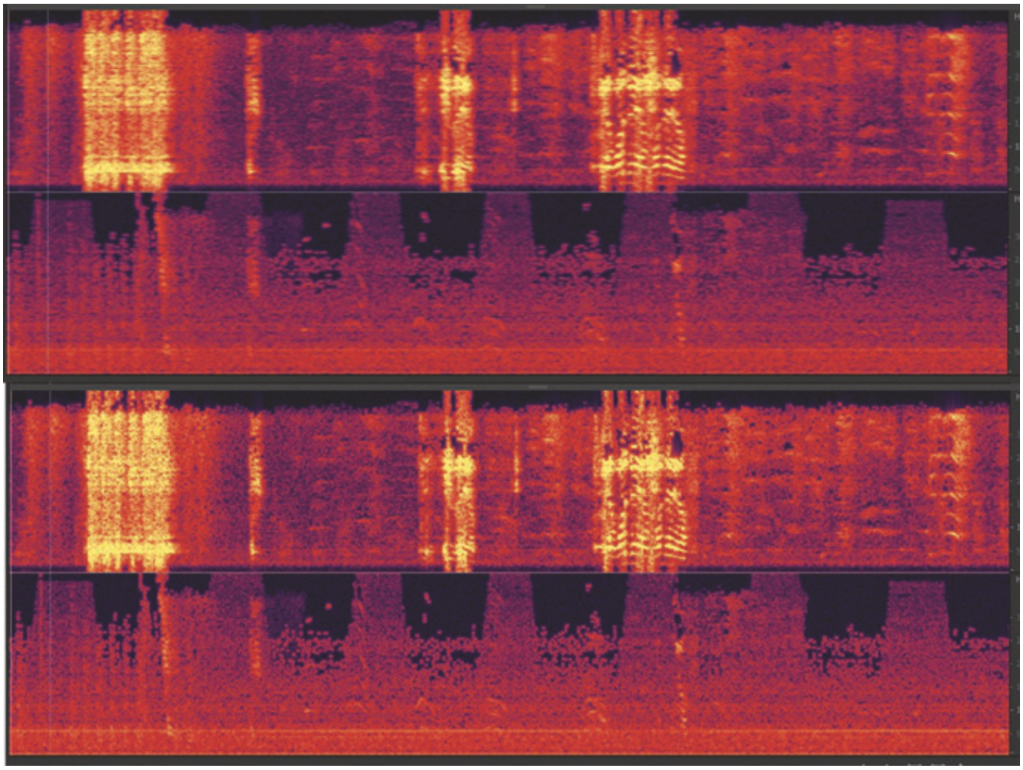


Figure 3.6: Subtle trade-offs involving time-frequency resolution are observed. The uppermost two rows feature spectrograms of the left and right channels in a stereo audio recording, displaying a marginal refinement in frequency resolution. Conversely, the lowermost two rows portray spectrograms of the identical stereo recording, showcasing a marginal enhancement in time resolution. The recording spans a total duration of 14 seconds and encompasses frequencies ranging from 0 to 4 kHz, with a linear scale representation [164].

Chapter 4

Background & State of the Art

1 A brief history of Deep Learning

Deep Learning, a branch of computer science that deals with training deep neural networks to learn from complex data and recognize abstract patterns and representations, has a fascinating history marked by key breakthroughs and technological advancements.

In its early days, before the term "Deep Learning" became popular, the concept of artificial neural networks emerged as a way to mimic the functioning of neurons in the human brain. Researchers explored the idea of using interconnected nodes, or "neurons," to process information and perform tasks. However, due to limited computational power and small datasets, progress in neural networks was slow, and the field faced challenges in training and optimizing these models effectively.

The "AI Winter" of the 1960s and 1970s was a period when artificial intelligence research faced scepticism and funding reductions. Neural networks, being part of the AI landscape, also experienced setbacks during this time. Researchers struggled to demonstrate their practicality and efficiency in real-world applications, which led to a decline in interest in neural networks and machine learning in general.

A major breakthrough came in the 1980s with the rediscovery of the backpropagation algorithm, a method for training neural networks by adjusting the weights of the connections between neurons. This algorithm allowed for more efficient training of neural networks, and it reignited interest in the field. However, even with backpropagation, training deeper networks remained challenging due to the "vanishing gradient" problem. This problem occurs when gradients (sensitivities of the output with respect to the input) diminish rapidly as they are propagated backwards through layers of the network, making it difficult to update the weights of early layers effectively.

The mid-2000s witnessed a turning point for Deep Learning when researchers, notably Geoff Hinton and his team, developed techniques to address the vanishing gradient problem. One such technique was "pre-training," where neural networks were first trained on simpler tasks before being fine-tuned for the main task. Another crucial innovation was "dropout," a regularization technique that randomly disables neurons during training, reducing overfitting and improving generalization performance.

Around the same time, the availability of large datasets, fueled by the rise of the internet and advances in data storage, proved instrumental in training more complex and deeper neural networks. Moreover, the development of powerful graphical processing units (GPUs) and the emergence of cloud computing infrastructures significantly accelerated the training process, making it feasible to train large-scale deep neural networks in reasonable timeframes.

Deep Learning models started showcasing impressive results in various applications, such as image classification, speech recognition, and natural language processing. Convolutional Neural Networks (CNNs) demonstrated remarkable performance in image analysis tasks, while Recurrent Neural Networks (RNNs) showed promise in processing sequential data like language and time series.

This success and the potential for even greater advancements led to a surge of interest and investments in Deep Learning, making it a transformative technology across multiple industries. In healthcare, it aided in medical image analysis and disease diagnosis. In finance, it improved fraud detection and predictive modeling. In automotive, it accelerated the development of autonomous vehicles. In entertainment, it enabled innovative applications in virtual reality and augmented reality.

The ongoing research and innovations in Deep Learning continue to expand its capabilities. Advanced architectures like transformer-based models have revolutionized natural language processing, enabling machines to understand context and semantics with exceptional accuracy. Generative Adversarial Networks (GANs) have opened up new possibilities in generating realistic images, videos, and other creative content. Reinforcement learning techniques have shown tremendous promise in training machines to make decisions and control complex systems.

2 Deep Learning Models

2.1 FFNNs: Feed Forward Neural Networks

The FFNN was the first and simplest type of artificial neural network to be designed. In such a network, data move in one direction from the input layer to the output layer, possibly via a series of hidden layers (Goodfellow et al., 2016 [86]; LeCun et al., 2015 [142]). Non-linear activation functions are usually used after each layer (possibly except for the output layer). While this definition of FFNN is very general and may include architectures such as CNNs (discussed in the next subsection), here we mainly focus on architectures made of fully-connected layers known as Perceptron and Multi-Layer Perceptron (MLP) (Goodfellow et al., 2016 [86]; LeCun et al., 2015 [142]). A Perceptron has no hidden layer, while the notion of MLP is a bit ambiguous: some authors state that an MLP has one hidden layer, while others allow more hidden layers.

2.2 CNNs: Convolutional Neural Networks

CNNs are a popular class of DNNs widely used for pattern recognition due to their property of being translation equivariant (Cohen et al., 2019[51]; Goodfellow et

al., 2016[86]). They have been successfully applied to various tasks, such as image classification (e.g., Krizhevsky et al., 2017)[134], natural language processing (NLP) (e.g., Kim, 2014)[122], or automatic speech recognition (e.g., Waibel et al., 1989)[277]. CNNs have also been used for Sound Source Localization.

2.3 RNNs: Recursive Neural Networks

RNNs are neural networks designed for modeling temporal sequences of data (Goodfellow et al., 2016[86]; LeCun et al., 2015[142]). Particular types of RNNs include long short-term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) and gated recurrent units (GRUs) (Cho et al., 2014)[45]. These two types of RNNs have become very popular thanks to their capability to circumvent the training difficulties that regular RNNs face, in particular, the vanishing and exploding gradient problems (Goodfellow et al., 2016 [86]; LeCun et al., 2015 [142]).

2.4 CRNNs: Convolutional Recursive Neural Networks

CRNNs are neural networks containing one or more convolutional layers and one or more recurrent layers. CRNNs have been regularly exploited for SSL since 2018 because of the respective capabilities of these layers: The convolutional layers have proven to be suitable for extracting relevant features and the recurrent layers are well-designed for integrating the information over time

2.5 Residual Neural Networks

Residual neural networks were originally introduced by He et al. (2016)[101], who highlighted that the design of deep networks can trigger issues like gradient explosion or vanishing gradients due to non-linear activation functions, leading to overall performance degradation. Residual connections were formulated as a solution to this predicament, permitting a feature to circumvent a layer block while concurrently adhering to the standard pathway through that layer block. This strategic arrangement facilitates the direct flow of gradients throughout the network, generally contributing to improved training outcomes.

2.6 Encoder-Decoder Neural Networks

An encoder-decoder network constitutes an architectural framework consisting of two fundamental components: an encoder and a decoder. The encoder, fueled by input features, generates a distinct representation of the input data. Subsequently, the decoder translates this novel data representation, acquired from the encoder, into the desired output data. Architectures adhering to this paradigm have been extensively investigated within the Deep Learning (DL) domain, primarily for their adeptness in yielding concise data representations through unsupervised methods (Goodfellow et al., 2016)[86].

1. **Autoencoder (AE)**: An AE represents a type of encoder-decoder neural network designed to produce an output identical to its input. Frequently, the encoder's final layer output has a smaller dimension compared to the

data's dimension. This specific layer, termed the bottleneck layer, serves as a compressed encoding of the input data. Originally constructed with feed-forward layers, the term "AE" is also contemporarily applied to networks featuring other layer types, such as convolutional or recurrent layers.

2. **Variational Autoencoder (VAE)**: A VAE represents a generative model that originated from the work of Kingma and Welling (2014)[124] and Rezende et al. (2014)[218], gaining significant popularity within the DL community. It can be perceived as a probabilistic adaptation of an AE. In contrast to a conventional AE, a VAE learns not only the data's probability distribution at the decoder's output but also models the probability distribution of the latent vector at the bottleneck layer. This characteristic strongly ties the VAE to the notion of unsupervised representation learning (Bengio et al., 2013)[21]. Consequently, the decoder can generate new data by sampling from these distributions.

3. **U-Net Architecture**: The U-Net architecture, initially introduced by Ronneberger et al. (2015)[225] for biomedical image segmentation, constitutes a distinctive fully-convolutional neural network design. In U-Net, input features undergo successive decomposition into feature maps within encoder layers and then recomposition into symmetrical feature maps within decoder layers, akin to CNNs. Consistent feature map dimensions at corresponding levels in the encoder and decoder permit the direct propagation of information through residual connections, giving rise to the characteristic U-shape configuration.

2.7 Attention-based NN

An attention mechanism is a technique that empowers a neural network to prioritize vectors within a temporal sequence that bears greater relevance to a specific task. Bahdanau et al. (2016)[15] initially introduced attention to enhancing sequence-to-sequence models, particularly Recurrent Neural Networks (RNNs), in the context of machine translation. The core principle involves assigning distinct weights to input sequence vectors when combined to estimate output sequence vectors. The model learns optimal weights reflecting both the interconnections among input sequence vectors (self-attention) and the significance of input vectors in elucidating each output vector (decoder attention). This pioneering work served as a foundation for the widely acclaimed Transformer architecture by Vaswani et al. (2017)[266], which substantially elevated machine translation performance. The Transformer architecture completely replaces RNNs with attention models.

3 Speech Recognition

Speech recognition is an interdisciplinary subfield of computer science and computational linguistics that develops methodologies and technologies that enable the recognition and translation of spoken language into text by computers with the main benefit of searchability. It is also known as automatic speech recognition (ASR), computer speech recognition or speech-to-text (STT). It incorporates knowledge and research in the computer science, linguistics and computer engineering fields. The reverse process is speech synthesis. Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated

vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker-independent" [228] systems.

Systems that use training are called "speaker dependent". Speech recognition applications include voice user interfaces such as voice dialing (e.g. "call home"), call routing (e.g. "I would like to make a collect call"), domotic appliance control, search key words (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), determining speaker characteristics,[180] speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed direct voice input).

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems.

3.1 Deep feedforward and recurrent neural networks

Deep Neural Networks and Denoising Autoencoders [158] are also under investigation. A deep feedforward neural network (DNN) is an artificial neural network with multiple hidden layers of units between the input and output layers.[108] Similar to shallow neural networks, DNNs can model complex non-linear relationships. DNN architectures generate compositional models, where extra layers enable composition of features from lower layers, giving a huge learning capacity and thus the potential of modeling complex patterns of speech data.[67] A success of DNNs in large vocabulary speech recognition occurred in 2010 by industrial researchers, in collaboration with academic researchers, where large output layers of the DNN based on context-dependent HMM states constructed by decision trees were adopted.[299, 58, 64].

See comprehensive reviews of this development and of the state of the art as of October 2014 in the recent Springer book from Microsoft Research.[298] See also the related background of automatic speech recognition and the impact of various machine learning paradigms, notably including deep learning, in recent overview articles.[65] One fundamental principle of deep learning is to do away with hand-crafted feature engineering and to use raw features. This principle was first explored successfully in the architecture of deep autoencoder on the "raw" spectrogram or linear filter-bank features,[66] showing its superiority over the Mel-Cepstral features which contain a few stages of fixed transformation from spectrograms. The true "raw" features of speech, waveforms, have more recently been shown to produce excellent larger-scale speech recognition results.[262]

3.2 End-to-end automatic speech recognition

Since 2014, there has been much research interest in "end-to-end" ASR. Traditional phonetic-based (i.e., all HMM-based model) approaches required separate components and training for the pronunciation, acoustic, and language models.

End-to-end models jointly learn all the components of the speech recognizer. This is valuable since it simplifies the training process and deployment process. For example, a n-gram language model is required for all HMM-based systems, and a typical n-gram language model often takes several gigabytes in memory making them impractical to deploy on mobile devices.[118] Consequently, modern commercial ASR systems from Google and Apple (as of 2017) are deployed on the cloud and require a network connection as opposed to the device locally.

The first attempt at end-to-end ASR was with Connectionist Temporal Classification (CTC)-based systems introduced by Alex Graves of Google DeepMind and Navdeep Jaitly of the University of Toronto in 2014.[88] The model consisted of recurrent neural networks and a CTC layer. Jointly, the RNN-CTC model learns the pronunciation and acoustic model together, however, it is incapable of learning the language due to conditional independence assumptions similar to a HMM. Consequently, CTC models can directly learn to map speech acoustics to English characters, but the models make many common spelling mistakes and must rely on a separate language model to clean up the transcripts.

Later, Baidu expanded on the work with extremely large datasets and demonstrated some commercial success in Chinese Mandarin and English.[9]

In 2016, the University of Oxford presented LipNet,[13] the first end-to-end sentence-level lipreading model, using spatiotemporal convolutions coupled with an RNN-CTC architecture, surpassing human-level performance in a restricted grammar dataset.[13]

A large-scale CNN-RNN-CTC architecture was presented in 2018 by Google DeepMind achieving 6 times better performance than human experts.[242]

An alternative approach to CTC-based models are attention-based models. Attention-based ASR models were introduced simultaneously by Chan et al. of Carnegie Mellon University and Google Brain and Bahdanau et al. of the University of Montreal in 2016.[42, 15]

The model named "Listen, Attend and Spell" (LAS), literally "listens" to the acoustic signal, pays "attention" to different parts of the signal and "spells" out the transcript one character at a time. Unlike CTC-based models, attention-based models do not have conditional-independence assumptions and can learn all the components of a speech recognizer including the pronunciation, acoustic and language model directly. This means, during deployment, there is no need to carry around a language model making it very practical for applications with limited memory.

By the end of 2016, the attention-based models have seen considerable success including outperforming the CTC models (with or without an external language model).[47] Various extensions have been proposed since the original LAS model.

Latent Sequence Decompositions (LSD) was proposed by Carnegie Mellon University, MIT and Google Brain to directly emit sub-word units which are more natural than English characters; University of Oxford and Google DeepMind extended LAS to "Watch, Listen, Attend and Spell" (WLAS) to handle lip reading surpassing

human-level performance.[49]

4 Speaker Identification & Verification

The term voice recognition or speaker identification refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process. In parfticular in speaker identification, an utterance from an unknown speaker is analyzed and compared with speech models of known speakers. The unknown speaker is identified as the one whose model best matches the input utterance.

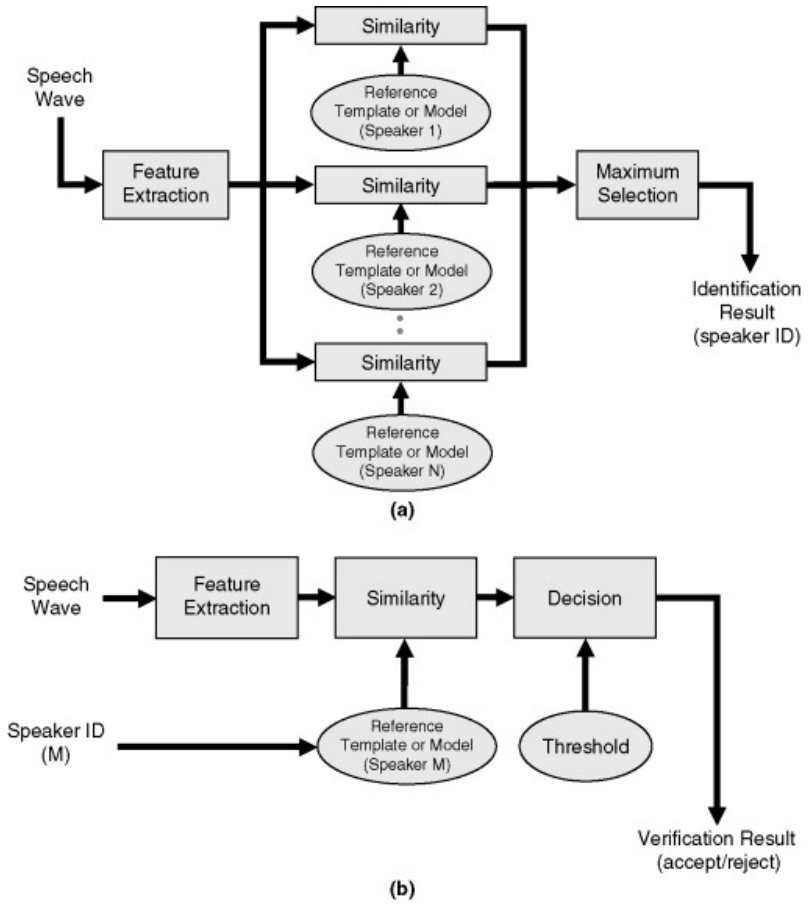


Figure 4.1: Speaker Identification and Speaker Verification Processes

In speaker verification, an identity is claimed by an unknown speaker, whose utterance is compared with a model for the registered speaker (customer) whose identity is being claimed. [82]

5 Speech Enhancement

In the survey [155], researchers have tackled the issue of speech enhancement over the years have been presented. The earliest works done in this domain consist of the various kinds of spectral enhancement methods, statistical based algorithms and subspace enhancement methods. These have performed well under test conditions

but in practical scenarios each comes with its own sets of drawbacks.

Adaptive noise cancellation is another popular domain in this regard. It has made itself an evergreen topic for research by being customizable through the use of machine learning techniques of optimization to tune its coefficients.
Machine learning algorithms are quite vast in nature. It is not possible to cover them all in this PhD Thesis. In [155] the authors discussed a few prominent ones and enlisted the strong points of each.
Advances in the field of Artificial intelligence have yielded fruitful results in speech enhancement. Neural networks have proven to be a strong tool in this regard. After simple NN, came DNN which was stronger in results but showed poor real world generalization upon encountering noise and speech signals that were unseen to it during training phase. Then came the era of the deep learning with CNN, RNN, LSTM and Attention mechanism[249] which has finally proven to be a reliable tool for generalization of real world noise cancellation problems. It can effectively deal with noise signals of all kinds, whether seen or unseen to it during training phase.

6 Speech Separation

Speech separation is the task of separating target speech from background interference. Traditionally, speech separation is studied as a signal processing problem. A more recent approach formulates speech separation as a supervised learning problem, where the discriminative patterns of speech, speakers, and background noise are learned from training data. Over the past decade, many supervised separation algorithms have been put forward. In particular, the recent introduction of deep learning to supervised speech separation has dramatically accelerated progress and boosted separation performance.
On the important survey on speech enhancement made on 2018 [278], the authors provide a comprehensive overview of the research on deep learning based supervised speech separation in the last several years.
Much of the overview is on separation algorithms where we review monaural methods, including speech enhancement (speech-nonspeech separation), speaker separation (multitalker separation), and speech dereverberation, as well as multimicrophone techniques.
In the recent literature, the Deep Learning State Of The Art (SOTA) Models for Speech Separation are based on deep learning and attention methods, like: ConvTasNet (2019)[154], Dual-Path RNN (2020)[153], WaveSplit (2021)[301], Sepformer (2021)[249]

7 Emotion Recognition

Emotion recognition from speech signals is an important but challenging component of Human-Computer Interaction (HCI). In the literature of speech emotion recognition (SER), many techniques have been utilized to extract emotions from signals, including many well-established speech analysis and classification techniques. Deep Learning techniques have been recently proposed as an alternative to traditional techniques in SER [172, 233, 232, 27]. In [120] the researchers present an overview of Deep Learning techniques and discusses some recent literature where these methods are utilized for speech-based emotion recognition. The review covers databases used,

Model	SI-SNRi	SDRi	Num.Param
ConvTasnet	12.7	13.1	5.1M
DualPathRNN	14.7	N.a.	2.6M
VSUNOS	16.9	N.a.	7.5M
WaveSplit	17.3	17.6	29M
WaveSplit + DM	17.8	18.1	29M
Sepformer	17.6	17.9	26M
Sepformer + DM	19.5	19.7	26M

Table 4.1: Best Speech Separation Performance on the WSJ0-3mix dataset, showed in Attention Is All You Need In Speech Separation(2021)[249]

emotions extracted, contributions made toward speech emotion recognition and limitations related to it.
Emotion recognition datasets are relatively small, making the use of the more sophisticated deep learning approaches challenging. In [200], the researchers proposed a transfer learning method for speech emotion recognition where features extracted from pre-trained wav2vec 2.0 models are modeled using simple neural networks.
They proposed to combine the output of several layers from the pre-trained model using trainable weights which are learned jointly with the downstream model. Further, we compare performance using two different wav2vec 2.0 models, with and without finetuning for speech recognition. The proposed approaches [200] was evaluated on two standard emotion databases IEMOCAP and RAVDESS, showing superior performance compared to results in the literature.

8 Voice Activity Detection

Voice activity detection (VAD), also known as speech activity detection or speech detection, is the detection of the presence or absence of human speech, used in speech processing. The main uses of VAD are in speech coding and speech recognition.
It can facilitate speech processing, and can also be used to deactivate some processes during a non-speech section of an audio session: it can avoid unnecessary coding/transmission of silence packets in Voice over Internet Protocol (VoIP) applications, saving on computation and on network bandwidth.
VAD is an important enabling technology for a variety of speech-based applications. Therefore, various VAD algorithms have been developed that provide varying features and compromises between latency, sensitivity, accuracy and computational cost. Some VAD algorithms also provide further analysis, for example whether the speech is voiced, unvoiced or sustained. Voice activity detection is usually independent of language.

Voice Activity Detection (VAD) is an important task in speech processing, and in recent years, deep learning techniques have been applied to improve the performance of VAD systems. The following are some of the most commonly used deep learning models for VAD in the last three years:

Deep Neural Networks (DNN): This is a deep neural network system for the automatic detection of speech in audio signals, otherwise known as Voice Activity

Detection (VAD) [168]

ECAPA-TDNN: This is a deep learning model based on Time Delay Neural Network (TDNN) architecture that uses a Convolutional Neural Network (CNN) with self-attention and a combination of propagation and aggregation techniques to improve the robustness and efficiency of the model [168, 234]

CRDNN (Convolutional Recurrent Deep Neural Network): This is a deep learning model used for Voice Activity Detection (VAD). An example of a CRDNN model for VAD is available on Hugging Face [54, 17]. The model has been trained on Libriparty and can process short and long audio recordings, returning the segments where vocal activity has been detected.

CNN self-attention voice activity detector: This is a deep learning model based on a Convolutional Neural Network (CNN) with self-attention used for Voice Activity Detection [79].

Deep learning models for VAD mainly use recurrent neural network architectures such as BiLSTM and CNN, often in combination with hybrid feature extraction. However, in recent years, new models such as ECAPA-TDNN and CRDNN have been proposed that use propagation and aggregation techniques to improve the robustness and efficiency of the model.

9 Sound Source Localization

The problem of Sound Source (SSL) is an active research area that has gained increasing interest in recent years. The goal of SSL is to identify and locate different sound events including speeches in an acoustic environment using various signal processing and machine learning techniques. SSL has the potential to enable a wide range of applications, from improved speech recognition and speaker verification to enhanced audio-based surveillance and security systems. 3D audio is gaining increasing interest in the machine learning community in recent years. The goal is to determine the coordinates of the speaker in a 3D space, based on audio signals captured by a microphone array. The range of applications is incredibly wide, extending from virtual and real conferencing to autonomous driving, improve speech recognition and speaker verification to enhanced audio-based surveillance and security systems.

However, SSL is a challenging task due to various sources of variability such as the acoustic environment, the microphone array geometry, and the presence of interfering sounds. These factors affect the audio signal in different ways and make it difficult to extract reliable information about the speaker's position including time difference of arrival (TDOA) localization, beamforming localization, and deep learning-based approaches. **TDOA-based** methods rely on the delays between the arrival of the acoustic signal at different positions of a microphone to calculate the sound source's position. **Beamforming-based** methods use a microphone array to direct a beam towards a specific position and identify the sound source's position. **Deep learning-based** methods use neural networks to learn sound representations and identify the sound source's position. To solve this problem, different deep learning architectures have been proposed and they can be divided into two main categories: end-to-end models and feature-based models. In both cases, the models

are trained using labelled data, which consists of audio signals and the corresponding speaker's position.

- End-to-end models receive the raw audio signals as input and output the speaker's position directly. These models typically consist of a deep neural network that learns a mapping from the audio signals to the speaker's position.
- Feature-based models, on the other hand, extract relevant features from the audio signals before performing the estimation. These features can be, for example, the time difference of arrival (TDOA) between the audio signals captured by the microphones or the beamforming vectors. These features are then used as input to a neural network that estimates the speaker's position.

Despite the progress made in SSL research, there are still many challenges that need to be addressed, such as the quality and availability of labelled data, the computational resources required for processing large amounts of audio data, and the dependence of the models on the specific application and environment in which they are used.

Figure 4.2 shows various methods have been proposed in the literature:

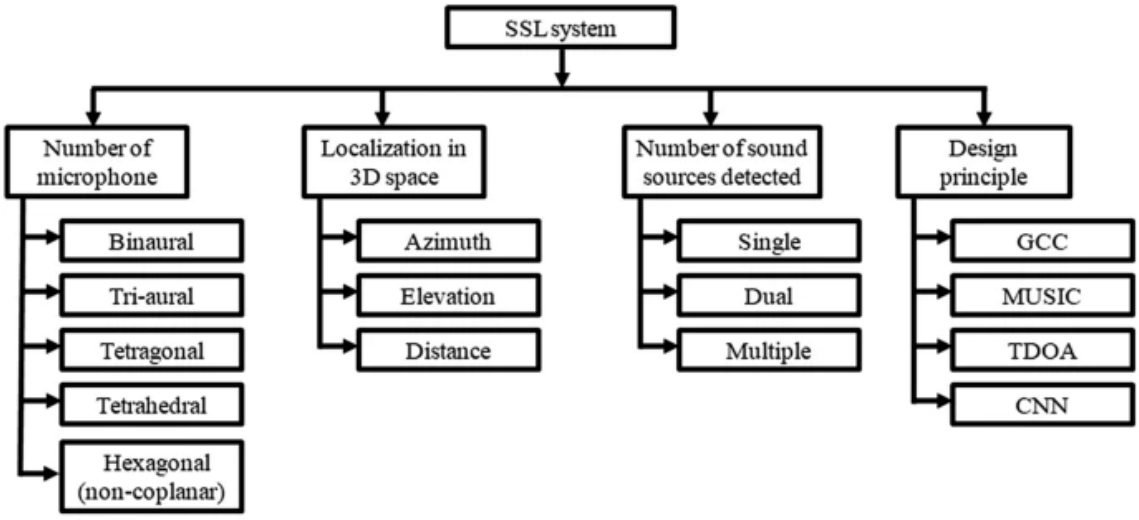


Figure 4.2: Sound Source Localization System anatomy

9.1 Acoustic Environment

This study centers on Sound Source Localization (SSL) within indoor environments. Specifically, it addresses scenarios where the microphone array and sound source(s) are situated within enclosed spaces, typically of moderate size such as office rooms or domestic settings. Such setups introduce reverberation, causing the recorded sound to encompass multiple multi-path components originating from the same source, in addition to the direct source-to-microphone propagation path. These components collectively constitute the Room Impulse Response (RIR), which is dependent on the source and microphone array positions (including orientation) as

well as the room configuration.

The presence of reverberation is commonly seen as a significant challenge, rendering SSL more complex compared to the simplified anechoic scenario where reverberation is absent, as seen in free-field propagation setups. Another factor to consider is noise. Noise can arise from interfering sources in the surrounding environment, such as television, background music, pets, and external sounds passing through open or closed windows. Noise is often treated as diffuse, lacking a specific directional origin. Additionally, imperfections in recording devices contribute to noise, usually in the form of artifacts.

9.2 Source Type

Within the Sound Source Localization (SSL) literature, a significant portion of systems focuses on localizing speech sources due to their pivotal role in associated tasks like speech enhancement or recognition. For instance, systems dedicated to speaker localization are discussed in works [41, 92, 98, 104]. These systems employ neural networks trained to estimate the Direction of Arrival (DoA) for speech sources, thereby acquiring specialization for this specific source category.

Conversely, other systems, particularly those engaged in the DCASE Challenge, encompass a broader array of sound source types [205]. Tailored to the task and dataset of the challenge, these methods exhibit the capability to localize diverse sources such as alarms, crying infants, crashes, barking dogs, screams (both male and female), speech (both male and female), footsteps, door knockings, ringing tones, phone sounds, and piano notes. It's noteworthy that localizing such sources, even when they temporally overlap, might not inherently pose a greater challenge compared to localizing overlapping speakers. This is due to the distinctive spectral characteristics that neural models can leverage to enhance detection and localization accuracy.

9.3 Number of sources

The count of sources (NoS) within a mixed signal is a pivotal parameter in Sound Source Localization (SSL). In the SSL domain, the NoS might be either treated as a known quantity (as an operational assumption) or estimated concurrently with the source location. In the latter scenario, the SSL problem encompasses both detection and localization. Notable examples of conventional SSL studies incorporating NoS estimation are detailed in the works [10, 137].

Single Sound Source Localization

Within numerous Deep Neural Network DNN-based investigations, the focus is often on localizing a solitary source. This simplified setup, termed "single-source SSL," is selected due to its relative simplicity (e.g., Bologni et al., 2021 [28]; Liu et al., 2021[151]; Perotin et al., 2018b[201]). In this context, networks are trained and assessed on datasets containing, at most, one active source (active denoting sound emission, and inactive signifying silence). In terms of NoS, the active sources in this scenario are either 1 or 0. While it is reasonable to control source activity artificially during training using synthetic data, this approach becomes unrealistic

during testing with real-world data. Alternatively, source activity can be estimated, yielding two approaches to tackle the source activity detection challenge.

One method involves employing a source detection algorithm prior to applying the SSL technique only to segments of the signal with an active source. Examples include voice activity detection (VAD) techniques in SSL systems by Chang et al. (2018)[43], Kim and Hahn (2018)[121], Li et al. (2016c)[148], and Sehgal and Kehtarnavaz (2018)[240]. The alternative approach is to simultaneously detect source activity and localize the source. Yalta et al. (2017)[292] augmented their DNN's output layer with an additional neuron, yielding an output of 1 when no source was active and 0 otherwise.

Multiple Sound Sources Localization

Moving to multi-source localization, the challenge becomes notably more intricate than single-source SSL. Presently, state-of-the-art DNN-based techniques address multi-source SSL within challenging environments. In this review, we define multi-source localization as encompassing instances where multiple sources overlap temporally, irrespective of their category (e.g., multiple speakers or distinct sound events). This scenario closely relates to multi-speaker conversations, sometimes with speech overlap, strongly connected to the speaker diarization problem. The connections between speaker localization, diarization, and source separation are intricate and beyond the scope of this review.

In multi-source scenarios, the source detection challenge transforms into a source counting issue, yet similar principles from the single-source scenario apply. In certain works, the NoS is considered a working hypothesis, and sources' Direction of Arrival (DoA) can be directly estimated. In cases where the NoS is unknown, a source counting system can be applied prior to SSL, such as through a dedicated DNN. For instance, Tian (2020)[259] trained a separate neural network to estimate the NoS in the recorded mixture signal, using this information in conjunction with the DoA estimation neural network's output. Alternatively, NoS estimation can proceed alongside DoA estimation, mirroring the single-source approach, based on the SSL network's output. In classification paradigms, the network's output often predicts the presence probability of a source in each space region. Setting a threshold on this estimated probability implies source counting. Alternatively, the ground truth or estimated NoS is typically used to select the corresponding number of classes with the highest probability.

Finally, certain DNN-based systems are deliberately designed to estimate NoS alongside DoAs. For example, Nguyen et al. (2020a)[181] propose a neural architecture featuring two output branches: one estimates the NoS (up to four sources, formulated as a classification task), while the other classifies azimuth into discrete regions. The DCASE Challenge encompasses numerous systems, wherein the SED task, integrated with SSL, inherently provides NoS estimates. Note that this survey extensively reviews numerous systems presented in the DCASE Challenge.

9.4 Moving Sources

The task of source tracking entails estimating the temporal evolution of source position(s), particularly in scenarios involving mobile sources. It is important to note that this survey paper does not delve into tracking as a separate pursuit; tracking is commonly handled by distinct algorithms that utilize sequences of Direction of Arrival (DoA) estimates garnered through applying Sound Source Localization (SSL) on consecutive time windows (Vo et al., 2015)[276]. Nevertheless, several Deep Learning (DL)-based SSL systems have demonstrated enhanced accuracy in localizing moving sources when trained on datasets encompassing this type of source (Adavanne et al., 2019b[5]; Diaz-Guerra et al., 2021b[71]; Guirguis et al., 2020[94]; He et al., 2021b[105]).

In certain instances, due to the limited availability of real-world datasets containing moving sources and the complexities associated with simulating signals involving mobile sources, certain systems trained on static sources have displayed commendable to satisfactory performance in localizing moving sources [90, 186, 252].

9.5 Microphones

Different microphone types are used in Sound Event Detection and Localization.

Microphones Types

The most commonly used types are omnidirectional, directional, and array microphones.

- **Omnidirectional microphones** capture sound from all directions and are suitable for capturing ambient sound or a large group of speakers.
- **Directional microphones** capture sound from a specific direction and are useful when trying to isolate a single speaker in a noisy environment.
- **Microphone arrays** are used to capture sound from multiple directions and can be used to estimate the direction of arrival of sound. They are useful for localizing sound sources in 3D space. The choice of microphone type depends on the specific application and the environment in which it is used.
- **Ambisonics microphones** are a type of microphone that captures sound from all directions in a full-sphere format, allowing for immersive and spatial audio recording. They use multiple capsules to capture sound from different directions, which is then combined into a single audio stream. The order of an ambisonics microphone refers to the number of capsules used to capture sound and the resulting number of audio channels. Higher-order ambisonics microphones capture more spatial detail but also require more processing power and storage space.

    1. First Order: Four-channel microphone for capturing full-sphere sound
    2. Second Order: Six-channel microphone for capturing full-sphere sound with improved spatial resolution
    3. Third Order: Eight-channel microphone for capturing full-sphere sound with even higher spatial resolution

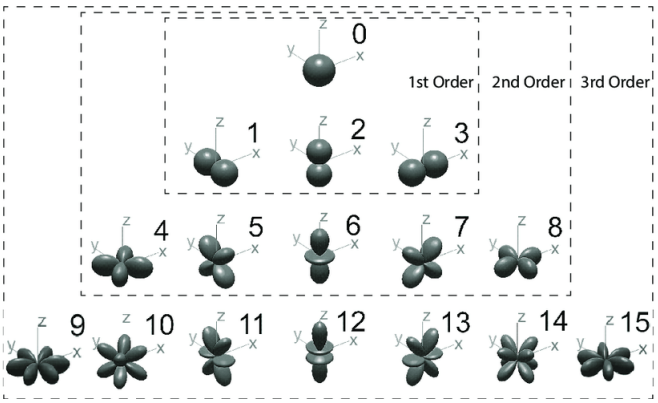


Figure 4.3: Ambisonics Microphones rapresentation

In addition to the microphone type, other factors such as the microphone placement and distance from the sound source also play a crucial role in the quality of the captured sound. For example, placing a directional microphone too close to a sound source can result in distortion, while placing it too far away can result in a weak signal. Moreover, the microphone type and placement also affect the accuracy of Sound Event Detection and Localization. In some cases, it may be necessary to use multiple microphones to capture sound from different directions and improve the accuracy of the estimation. Overall, choosing the appropriate microphone type and placement is crucial for capturing high-quality sound and improving the accuracy of SSL systems.

Microphones Configurations

SSL systems use multiple microphones arranged in different geometrical configurations called microphone array configurations. Researchers can choose any type of configuration such as linear, circular, tetrahedral, etc. Figure 4.4 shows different configurations, including binaural, tri-aural, tetra-aural, and clustered microphone arrays.

Microphones Considerations

It is important to consider the specific application and environment in which the system will be used to make an informed decision. To improve the accuracy of Sound Source Localization and Localization (SSL), it is important to consider several factors such as the environment, the acoustic characteristics of the environment, the microphone arrangement, and the quality of the acoustic signal. Additionally, it may be necessary to use multiple microphones to capture sound from different directions and improve the accuracy of the estimation. While there are several models available for SEDL, it is important to note that these models are often dependent on the specific application and environment in which they are used, and the choice of model should be made after careful consideration of these factors.

9.6 Sound Source Localization Design

Figure 4.5 shows a flowchart of the SSL system design, including extracting sound signal components, recording with different microphones, and analyzing variations in the signal components using algorithms like AML, GCC, MCCC, TDOA, or

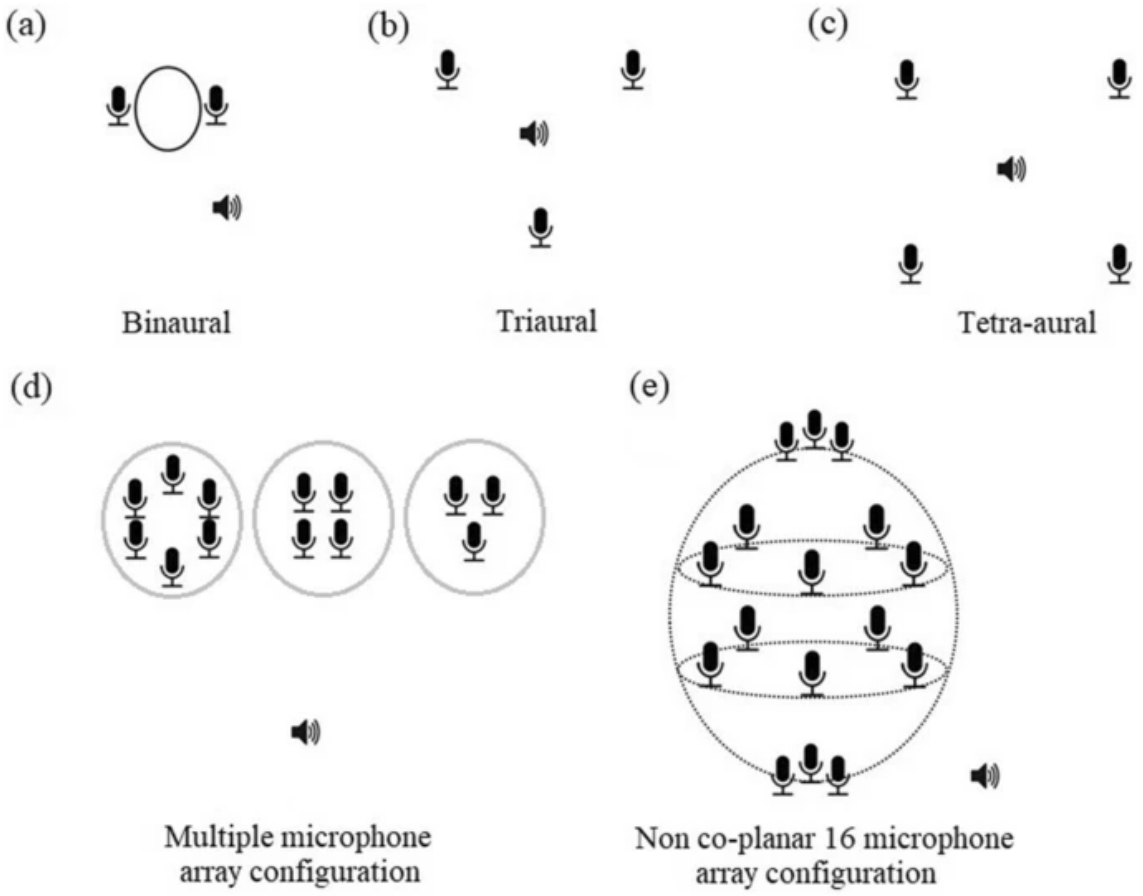


Figure 4.4: Microphones Configurations

CNN. SED and source localization in 1D, 2D, or 3D planes is achieved using these algorithms.

First, multichannel signals recorded with an array of I microphones distributed in space contain information about the location of the source(s). Indeed, when the microphones are close to each other compared to their distance to the source(s), the microphone signal waveforms, although appearing similar from a distance, exhibit more or less notable and complex differences in terms of delay and amplitude, depending on the experimental setup. These interchannel differences are due to distinct propagation paths from the source to the different microphones, for both the direct path (line of sight between source and microphone) and the numerous reflections that compose the reverberation in an indoor environment. In other words, a source signal $s_j(t)$ is convolved with different room impulse responses (RIRs) $a_{i,j}(t)$, which depend on the source position, microphone position and directivity (I denotes the microphone index in the array), and acoustic environment configuration (e.g., room shape):

$$x_i(t) = a_{i,j}(t) * s_j(t) + n_i(t) = \sum_{\tau=0}^{T-1} a_{i,j}(\tau) * s_j(t-\tau) + n_i(t) \tag{4.1}$$

where $x_i(t)$ denotes the resulting recorded signal at microphone i, $n_i(t)$ is the noise signal at microphone i (diffuse, "background" noise and possibly some sensor noise), and $*$ denotes the convolution (note that we work with digital signals and t and τ are discrete-time indexes; T is the effective length of the RIR).

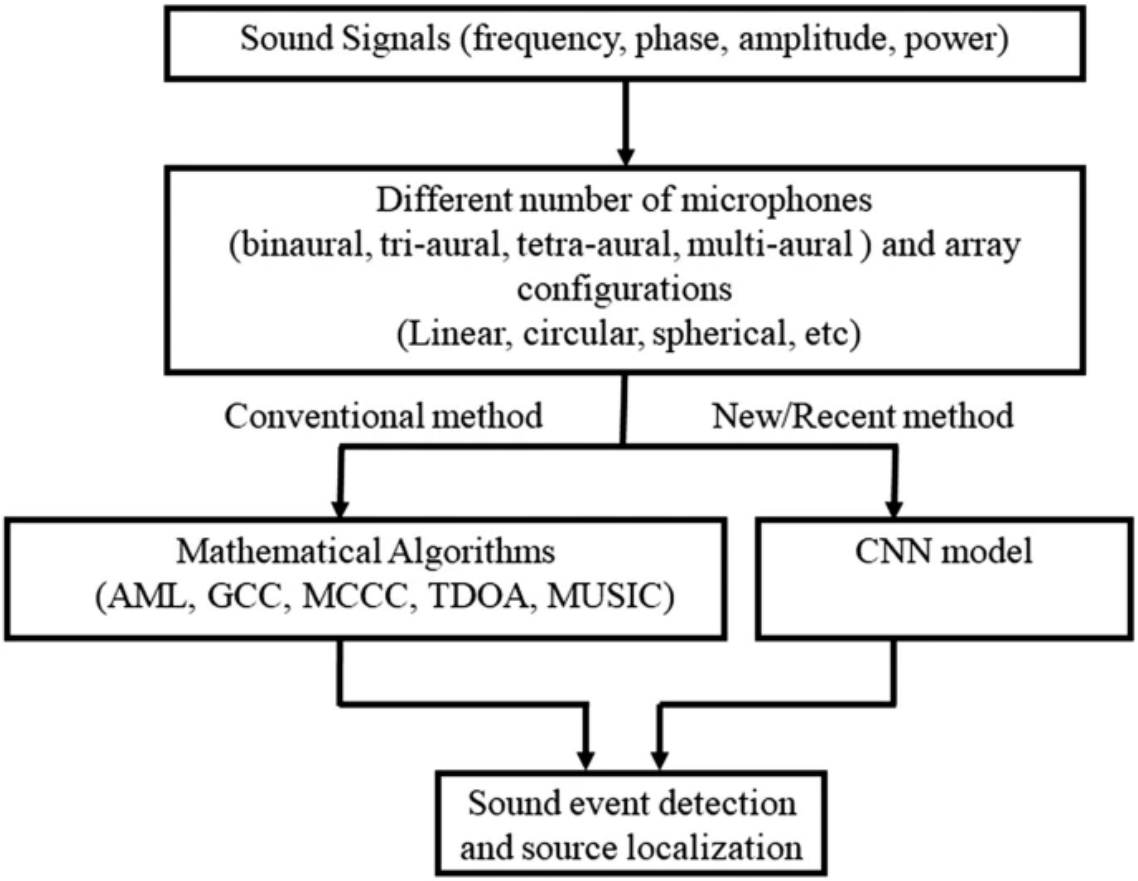


Figure 4.5: Sound Source Localization System Design

Therefore, the recorded signal contains information on the relative source-to-microphone array position. The microphone signals are often expressed in the time-frequency (TF) domain, using the short-term Fourier transform (STFT), where the convolution in Eq. 4.1 is assumed to transform into a product between the STFT of the source signal $S_j(f,n)$ and the acoustic transfer function (ATF) $A_{i,j}(f)$, which is the (discrete) Fourier transform of the corresponding RIR and is thus encoding the source spatial information (f denotes the frequency bin, and n is the STFT frame index)

$$X_i(f,n) = A_{i,j}(f)S_j(f,n) + N_i(f,n) \tag{4.2}$$

When several, say J, sources are present, the recorded signal is the sum of their contribution (plus the noise),

$$x_i(t) = \sum_{j=1}^{J} a_{i,j}(t) * s_j(t-) + n_i(t) \tag{4.3}$$

This latter equation is often reformulated in the Time-Frequency domain in matrix form,

$$\mathbf{X}(f,n) = \mathbf{A}(f)\mathbf{S}(f,n) + \mathbf{N}(f,n) \tag{4.4}$$

where $\mathbf{X}(f,n) = [\mathbf{X}_1(f,n), \ldots, \mathbf{X}_I(f,n)]^T$ is the microphone signal vector, $\mathbf{A}(f)$ is the matrix gathering the ATFs, $\mathbf{S}(f,n) = [\mathbf{S}_1(f,n), \ldots, \mathbf{S}_J(f,n)]^T$ is the source signal vector, and $\mathbf{N}(f,n) = [\mathbf{N}_1(f,n), \ldots, \mathbf{N}_I(f,n)]^T$ is the noise vector.

In that multi-source case, the difficulty of the SSL problem is that the contributions of the different sources generally overlap in time. SSL then requires to proceed to some kind of source clustering, which is generally easier to proceed in the frequency or TF domain due to the natural sparsity of audio sources in that domain (Rickard, 2002).

Then the recorded signals from the microphone array are passed into Sound Source Localization methods to estimate DOA.

9.7 Sound Source Localization Methods

The SSL models are based on various technologies and methods, including traditional Mathematical methods like: Time Difference of Arrival (TDOA) localization, Beamforming localization, and Deep Learning Methods.

Classic Methods

Before the emergence of Deep Learning (DL), a set of signal processing techniques was developed to address Sound Source Localization (SSL). DiBiase et al. (2001) [72] conducted an extensive review of these techniques, while Argentieri et al. (2015)[12] explored them within the context of robotics. This section provides a concise overview of prevalent conventional SSL methods. This presentation serves a dual purpose: firstly, conventional SSL methods are commonly used as reference benchmarks for DL-based approaches; secondly, numerous DL-based SSL methods employ input features extracted using conventional methods (as detailed in Section 9.7).

The time difference of arrival (TDoA) When the microphone array's geometry is known, the estimation of Direction of Arrival (DoA) can be achieved by estimating **the time difference of arrival (TDoA)** of sources between microphones (Xu et al., 2013)[289].

These models use delays between the arrival of the acoustic signal at different positions of a microphone to calculate the speaker's position. The time for the sound signal to reach the microphone is calculated and speed of sound signal is measured. The difference in arrival of sound signal reaching both the microphones is used to calculate the distance of sound source from microphones.

$$\Delta d = c * \Delta t \tag{4.5}$$

where c is speed of light, δt, is difference in arrival times at microphones.

$$\Delta_d = \sqrt{(x_2 - x)^2 - (y_2 - y)^2} - \sqrt{(x_1 - x)^2 - (y_1 - y)^2} \tag{4.6}$$

where (x_1, y_1) and (x_2, y_2) are known positions of beacons. By using non-linear regression, equation is converted to form hyperbola. After calculating many hyperbolas, SSL can be done by finding the intersection.

One of the widely used methods for 2-microphone arrays is the **Generalized Cross Correlation (CC) with Phase Transform (GCC-PHAT)** method, initially introduced by Knapp and Carter (1976) [127]. This method involves calculating the inverse Fourier transform of a weighted cross-power spectrum (CPS) between the microphone signals of the two microphones.

$$r_{1,2}(\tau) = \sum_{f=0}^{F-1} \frac{X_1(f)X_2(f)^*}{|X_1(f)X_2(f)^*|} e^{j2\pi(f\tau/N)} \tag{4.7}$$

where $\mathbf{X}_i(f)$ are the $N - point$ Fourier transform of the microphone signals $x_i(t)$, and $\mathbf{X_1}(f)\mathbf{X_2}(f)^*$ is the CPS ($*$ denotes the complex conjugate). The TDoA estimate is then obtained by finding the time delay between the microphone signals that maximizes the GCC-PHAT function,

$$\hat{\tau} = argmax_{1,2}(\tau) \tag{4.8}$$

The TDoA estimate is subsequently derived by identifying the time delay between microphone signals that maximizes the GCC-PHAT function. The GCC approach has been extended to encompass arrays with more than two microphones, demonstrating that localization could be improved by leveraging multiple microphone pairs (Benesty et al., 2008; DiBiase et al., 2001)[20, 72]. Utilizing an acoustic power map of x represents the spatial coordinates on a regular grid, offers an alternative approach to ascertain the Direction of Arrival (DoA) for one or multiple sound sources, as peaks in this map generally correspond to the sources' DoA.

The Steered-Response Power (SRP) map has seen extensive use in acoustic applications. It involves directing delay and sum beamformers towards candidate grid positions and quantifying the energy originating from these directions. **The PHAT (Phase Transform) version of SRP called SRP-PHAT**, known for its robustness against reverberation, is notably popular. Essentially, it can be obtained by averaging the Generalized Cross-Correlation with Phase Transform (GCC-PHAT) computed for all pairs of microphones [72],

$$\mathbf{P}(\mathbf{x}) = \sum_{m_1=1}^{M} \sum_{m_2=m_1+1}^{M} \mathbf{r_{1,2}}(\tau_{\mathbf{m_1},\mathbf{m_2}}(\mathbf{x})) \tag{4.9}$$

where $\tau_{\mathbf{m_1},\mathbf{m_2}}(\mathbf{x})$ is the delay between the microphones m_1 and m_2 associated with the spatial position x.

An alternative approach to constructing the SRP-based acoustic map, which is often computationally intensive due to grid searching, is source localization through sound intensity. The utilization of sound intensity for source localization has a well-established history (e.g., [18, 107, 115, 179, 211, 257]). In favorable acoustic conditions, sound intensity aligns with the direction of sound wave propagation, enabling efficient Direction of Arrival (DoA) estimation. Regrettably, its precision diminishes rapidly in the presence of acoustic reflections [59].

Subspace methods represent another classical category of localization algorithms. These techniques center around computing the **Cross-Power Spectral density (CPS)** matrix R(f), which is defined as follows:

$$R(f) = \sum_{n=1}^{N} X(f,n)X(f,n)^H \tag{4.10}$$

Here, $X(f,n)$ signifies the Short-Time Fourier Transform (STFT) or, in a broader sense, a local discrete Fourier transform of the multichannel signal vector described in Equation (4), where H denotes the Hermitian operator. Subsequently, these methods involve the eigenvalue decomposition (EVD) of $R(f)$. When considering uncorrelated target source signals and noise, the Multiple Signal Classification (MUSIC) method (Schmidt, 1986)[235] applies EVD to estimate the subspaces for both signals and noise.

Following Equation 4.4, the signal subspace bases are presumed to correspond to the columns of the mixing matrix A(f), which represents the multichannel Acoustic Transfer Functions (ATFs) of the sources, often referred to as steering vectors in this context. These signal or noise subspace bases are then harnessed for investigating a specific direction to detect the presence of a source, employing spatial filtering or beamforming techniques [20, 263].

This time-intensive search process can be alleviated through the use of the Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT)

algorithm (Roy and Kailath, 1989)[226], which leverages the inherent structure of the source subspace to directly infer the Direction of Arrival (DoA) of the source. However, it's worth noting that ESPRIT often sacrifices some predictive accuracy compared to MUSIC (Mabande et al., 2011)[159]. It's important to mention that MUSIC and ESPRIT are designed for narrowband signals, although there have been wideband extensions proposed [73, 110].

Subspace methods are known for their robustness against noise and their capability to yield highly accurate estimates, but they are susceptible to the effects of reverberation.

Methods based on probabilistic generative mixture models have been proposed by, e.g. [74, 147, 165, 166, 223, 237, 285]. Typically, the models are variants of Gaussian mixture models (GMMs), with one Gaussian component per source to be localized or per candidate source position. In very few papers ([166]), the model is trained offline with a dedicated training dataset. But most often, the model parameters are directly estimated "at test time," that is using the multichannel signal containing the sources to localize. This is done by maximizing the data likelihood function with histogram-based or expectation-maximization (EM) algorithms exploiting the sparsity of sound sources in the TF domain [220], which can be computationally intensive. A GMM variant functioning directly in regression mode, i.e., a form of Gaussian mixture regression (GMR), was proposed for single-source localization by [61] and later extended to multi-source localization (and possibly separation) [60, 62]. The GMR is locally linear but globally non-linear and the estimation of the model parameters is done offline on training data. Hence the spirit is close to DNN-based SSL. White noise signals convolved with synthetic RIRs were used for training. The method was shown to generalize well to speech signals, which are sparser than noise in the TF domain, thanks to the use of a latent variable modeling the signal activity in each TF bin.

Mixture models have a strong association with Bayesian inference, a framework that takes into account the posterior distribution of model parameters based on the observed data, involving both the likelihood function and a prior distribution of the model parameters. Escolano et al. (2014) explored the application of Bayesian inference within the context of a Laplacian source mixture model, utilizing GCC-PHAT features in a setup with a two-microphone array. Remarkably, they employed a two-tiered Bayesian inference approach: one for estimating the Number of Sources (NoS), employing Bayesian model selection as an integral part, and another for estimating the model parameters, which subsequently encompass the associated source Directions of Arrival (DoAs), using posterior distribution assessment. In this research, the evaluation of these associated distributions was carried out through sampling techniques, including methods such as Markov Chain Monte Carlo (MCMC).

This same methodology found further application in subsequent studies. Bush and Xiang (2018) extended it to a coprime array configuration, which consists of two superimposed spatially undersampled uniform linear arrays (Vaidyanathan and Pal, 2010). Additionally, Landschoot and Xiang (2019) applied this approach in the spherical harmonics (SH) domain, employing a spherical microphone array setup (as detailed in Sec. V).

Deep Learning Methods

In this section, we delve into the neural network architectures that have been proposed within the literature to tackle the SSL problem. However, we abstain from outlining the fundamental principles of these neural networks, as they have been comprehensively expounded in the broader DL literature (Chollet, 2017 [46]; Goodfellow et al., 2016 [86]; LeCun et al., 2015[142]). Developing DNNs for a specific application often necessitates exploring various architectures and potentially amalgamating them, along with fine-tuning their hyperparameters. This has been the trajectory of SSL over the past decade, mirroring the overarching progression of DNNs towards more intricate architectures or innovative models adopted by both the DL and SP communities at large. These models extend beyond the SSL problem, encompassing realms like attention models. Put differently, the DNN architectures employed in SSL are frequently derived from other studies spanning diverse domains, underpinned by their proven efficacy in audio signals or other signal types. Similarly, the amalgamation of different models is a common practice, whether in parallel or sequentially.

Consequently, our exposition is structured according to the categories of layers integrated into the networks, taking a gradual and comprehensive approach towards complexity: networks within a given category can encompass layers from preceding categories. Thus, our initial focus is on systems grounded in feedforward neural networks (FFNNs). Subsequently, we delve into CNNs and recurrent neural networks (RNNs), often encompassing certain feedforward layers. Following this, we survey architectures that interweave CNNs with RNNs, denominated as convolutional recurrent neural networks (CRNNs). Our attention then shifts to neural networks that incorporate residual connections and those equipped with attention mechanisms. Lastly, we present SSL systems characterized by an encoder-decoder architecture.

FFNNs: Feed Forward Neural Networks A few pioneering SSL methods using shallow neural networks (Perceptron or 1-hidden layer MLP) and applied in "unrealistic" setups (e.g., assuming direct-path sound propagation only). One of the first uses of an MLP for SSL was proposed by Kim and Ling (2011) [123], who actually considered several MLPs. One network estimates the NoS, after which a distinct network is used for SSL for each considered NoS. The authors evaluated their method on reverberant data even though they assumed an anechoic setting. Tsuzuki et al. (2013) [261] proposed using a complex-valued MLP in order to process complex two-microphone-based features, which led to better results than using a real-valued MLP. Youssef et al. (2013) [297] also used an MLP to estimate the azimuth of a sound source from a binaural recording made with a robot head. The interaural time difference (ITD) and the interaural level difference (ILD) values were separately fed into the input layer and were each processed by a specific set of neurons. A single-hidden-layer MLP was used by Xiao et al. (2015)[288], taking GCC-PHAT-based features as inputs and tackling SSL as a classification problem, which showed an improvement over conventional methods on simulated and real data. A similar approach was proposed by Vesperini et al. (2016) [274], but the localization was done by regression in the horizontal plane.

Naturally, MLPs with deeper architecture (i.e., more hidden layers) have also been investigated for SSL. Roden et al. (2015) [222] compared the performance of an MLP with two hidden layers and different input types, the number of hidden

neurons being linked to the type of input features. Yiwere and Rhee (2017) [295] used an MLP with three hidden layers (tested with different numbers of neurons) to output source azimuth and distance estimates. An MLP with four hidden layers was tested by He et al. (2018a) [103] for multi-source localization and speech/non-speech classification, showing similar results as a 4-layer CNN.

Ma et al. (2015)[156] proposed using a different MLP for different frequency sub-bands, with each MLP having eight hidden layers. This idea is based on the assumption that, in the presence of multiple sources, each frequency band is mostly dominated by a single source, which enables the training to be done exclusively on single-source data. The output of each sub-band MLP corresponds to a probability distribution on azimuth regions, and the final azimuth estimations are obtained by integrating the probability values over the frequency bands. Another system in the same vein was proposed by Takeda et al. in several papers (Takeda and Komatani, 2016a,b, 2017; Takeda et al., 2018)[254, 255, 256]. In these works, the eigenvectors of the recorded signal interchannel correlation matrix were separately fed per frequency band into parallel branches of the network, particularly into specific fully-connected layers. Then, several additional fully-connected layers progressively integrated the frequency-dependent outputs. The authors showed that this specific architecture outperforms a more conventional 7-layer MLP and the classical MUSIC algorithm on anechoic and reverberant single- and multi-source signals. Opochinsky et al. (2019) [187] proposed a small 3-layer MLP to estimate the azimuth of a single source using the relative transfer function (RTF) of the signal. Their approach is weakly supervised since one part of the loss function is computed without the ground truth DoA labels.

The indirect use of an MLP was explored by Pak and Shin (2019)[190], who used a 3-layer MLP to enhance the interaural phase difference (IPD) of the input signal, which was then used for DoA estimation.

CNNs: Convolutional Neural Networks Hirvonen (2015) [109] was the first to use a CNN for SSL. He employed this architecture to classify an audio signal containing one speech or musical source into one of eight spatial regions (see Fig. 3). This CNN is composed of four convolutional layers to extract feature maps from multichannel magnitude spectrograms, followed by four fully-connected layers for classification. Classical pooling is not used because, according to the author, it does not seem relevant for audio representations. Instead, a 4-tap stride with a 2-tap overlap is used to reduce the number of parameters. This approach shows good performance on single-source signals and is capable of adapting to different configurations without hand-engineering. However, two topical issues of such a system were pointed out by the author: the robustness of the network with respect to a shift in source location, and the difficulty of interpreting the hidden features.

Chakrabarty and Habets also designed a CNN to predict the azimuth of one (Chakrabarty and Habets, 2017a)[39] or two (Chakrabarty and Habets, 2017b, 2019b)[40, 41] speakers in reverberant environments. The input features are the multichannel STFT phase spectrograms. In Chakrabarty and Habets (2017a), they proposed using three successive convolutional layers with 64 filters of size 2×2 to consider neighboring frequency bands and microphones. In Chakrabarty and Habets (2017b) [40], they reduced the filter size to 2×1 (1 in the frequency axis) because of

the W-disjoint orthogonality (WDO) assumption for speech signals, which assumes that several speakers are not simultaneously active in a same TF bin (Rickard, 2002) [220]. In Chakrabarty and Habets (2019b) [41], they demonstrated that for an M-microphone array, the optimal number of convolutional layers for exploiting phase correlations between the neighboring microphones is M – 1.

He et al. (2018a) [103] compared a 4-layer MLP and a 4-layer CNN for the multi-speaker detection and localization task. The results showed similar accuracy for both architectures. A deeper architecture was proposed by Yalta et al. (2017) [292], with 11 to 20 convolutional layers depending on the experiments. These deeper CNNs showed robustness against noise compared to MUSIC, as well as smaller training time, but this was partly due to the presence of residual blocks. A similar architecture was presented by He et al. (2018b) [102], with many convolutional layers and some residual blocks, although with a specific multi-task configuration. The end of the network was split into two convolutional branches, one for azimuth estimation, and the other for speech/non-speech signal classification.

While most localization systems aim to estimate the azimuth or both the azimuth and elevation, Thuillier et al. (2018) [258] investigated the estimation of only the elevation angle using a CNN with binaural input features: the ipsilateral and contralateral head-related transfer function (HRTF) magnitude responses. Vera-Diaz et al. (2018) [271] chose to apply a CNN directly on raw multichannel waveforms, assembled side by side as an image, to predict the Cartesian coordinates (x, y, z) of a single static or moving speaker. The successive convolutional layers contain around a hundred filters from size 7×7 for the first layers to 3×3 for the last layer. Ma and Liu (2018) [157] also used a CNN to perform regression, but they used the CPS matrix as an input feature. To estimate both the azimuth and elevation, Nguyen et al. (2018) used a relatively small CNN (two convolutional layers) in regression mode, with binaural input features. A similar approach was considered by Sivasankaran et al. (2018) [247] for speaker localization based on a CNN. They showed that injecting a speaker identifier, particularly a mask estimated for the speaker uttering a given keyword, alongside the binaural features at the input layer improved the DoA estimation.

A joint VAD and DoA estimation CNN was developed by Vecchiotti et al. (2018)[270]. They showed that both problems can be handled jointly in a multi-room environment using the same architecture, although considering separate input features (GCC-PHAT and log-mel-spectrograms) in two separate input branches. These branches are then concatenated in a further layer. Vecchiotti et al. (2019b) [269] extended this work by exploring several variant architectures and experimental configurations, and Vecchiotti et al. (2019a) [268] developed an end-to-end auditory-inspired system based on a CNN, with Gammatone filter layers included in the neural architecture. A method based on mask estimation was proposed by Zhang et al. (2019b)[304], in which a TF mask was estimated and used to either clean or be appended to the input features, facilitating the DoA estimation by a CNN.

Nguyen et al. (2020a) [181] presented a multi-task CNN containing ten convolutional layers with average pooling, inferring both the NoS and the sources' DoA. They evaluated their network on signals with up to four sources, showing very good performance in both simulated and real environments. A small 3-layer CNN was employed by Varanasi et al. (2020) [264] to infer both azimuth and elevation using

signals decomposed with third-order SH. The authors tried several combinations of input features, including using only the magnitude and/or the phase of the spherical harmonic decomposition.

In the context of hearing aids, a CNN was applied to both VAD and DoA estimation by Varzandeh et al. (2020) [265]. This system is based on two input features, GCC-PHAT and periodicity degree, both fed separately into two convolutional branches. These two branches are then concatenated in a further layer, which is followed by feedforward layers. Fahim et al. (2020) applied an 8-layer CNN to the so-called modal coherence of first-order Ambisonics input features for the localization of multiple sources in a reverberant environment. They proposed a new method to train a multi-source DoA estimation network with only single-source training data, showing an improvement over the system of Chakrabarty and Habets (2019b) [41], especially for signals with three speakers. Hao et al. (2020) [98] investigated a real-time implementation of SSL using a CNN with a relatively small architecture (three convolutional layers).

Krause et al. (2020a) [133] investigated the use of several types of convolution. They reported that networks using three-dimensional (3D) convolutions (on the time, frequency, and channel axes) achieved better localization accuracy compared to those based on two-dimensional (2D) convolutions, complex convolutions, and depth-wise separable convolutions (all of them on the time and frequency axes), but with a high computational cost. They also showed that the use of depth-wise separable convolutions leads to a good trade-off between accuracy and model complexity (to our knowledge, they were the first to explore this type of convolutions).

Bologni et al. (2021)[28] proposed a neural network architecture including a set of 2D convolutional layers for frame-wise feature extraction, followed by several one-dimensional (1D) convolutional layers in the time dimension for temporal aggregation. Diaz-Guerra et al. (2021b)[71] applied 3D convolutional layers on SRP-PHAT power maps computed for both azimuth and elevation estimation. They also used a couple of 1D causal convolutional layers at the end of the network to perform single-source tracking. Their whole architecture was designed to function in fully causal mode so that it can be adapted for real-time applications. Wu et al. (2021a)[286] proposed using a supervised image mapping approach inspired from computer vision works and referred to as image translation. They used a CNN (completed with residual layers) to map an input 2D image [DoA features extracted by conventional beamforming and reshaped as a function of Cartesian coordinates (x, y)] into an output 2D image of the target source position (in which the pixel intensity is decreasing rapidly with the distance to the source), from which the source location is obtained.

As mentioned in the introduction, the DCASE Challenge includes a SELD task, and CNNs have also been used in some of the challenge candidate systems (Politis et al., 2020b[205]). Chytas and Potamianos (2019)[50] used convolutional layers with hundreds of filters of size 4×10 for azimuth and elevation estimation in a regression mode. Kong et al. (2019)[131] compared different numbers of convolutional layers for SELD, while an 8-layer CNN was proposed by Noh et al. (2019)[185] to improve the results over the baseline.

An indirect use of a CNN was proposed by Salvati et al. (2018)[230]. They

trained the neural network to estimate a weight for each of the narrow-band SRP components fed at the input layer in order to compute a weighted combination of these components. In their experiments, they showed on a few test examples that this allowed for a better fusion of the narrow-band components and reduced the effects of noise and reverberation, leading to better localization accuracy.

In the DoA estimation literature, a few works have explored the use of dilated convolutions in DNNs. Dilated convolutions, also known as atrous convolutions, are a type of convolutional layer in which the convolution kernel is wider than the classical one but zeros are inserted so that the number of parameters remains the same. Formally, a 1D dilated convolution with a dilation factor l is defined by:

$$(x * k)(n) = \sum_i x(n - l_i)k(i) \tag{4.11}$$

where x is the input and k the convolution kernel. The conventional linear convolution is obtained with l=1. This definition extends to multidimensional convolution.

Chakrabarty and Habets (2019a) [91] demonstrate that incorporating dilated convolutions with gradually increasing dilation factors reduces the optimal number of convolutional layers of their original CNN architecture (Chakrabarty and Habets, 2019b)[41] (discussed previously in this section). This leads to an architecture with similar SSL performance and lower computational cost.

RNNs: Recursive Neural Networks There are few published works on SSL using only RNNs, as recurrent layers are often combined with convolutional layers. Nguyen et al. (2021a)[184] used an RNN to align SED and DoA predictions, which were obtained separately for each possible sound event type. The RNN was ultimately used to determine which SED prediction matched which DoA estimation. A bidirectional LSTM network was used by Wang et al. (2019)[281] to estimate a TF mask to enhance the signal, further facilitating DoA estimation by conventional methods such as SRP or subspace methods.

CRNNs: Convolutional Recursive Neural Networks In the series of papers by Adavanne et al. (Adavanne et al., 2019a, 2018, 2019b) [3, 4, 5], the authors used a CRNN for SELD, in a multi-task configuration, with first-order Ambisonics (FOA) input features. In Adavanne et al. (2018)[3], their architecture contained a series of successive convolutional layers, each followed by a max-pooling layer and two bidirectional gated recurrent unit (BGRU) layers. Then, a feedforward layer provided an estimation of the spatial pseudo-spectrum (SPS) provided by the MUSIC algorithm (Schmidt, 1986)[235], acting as an intermediary output (see Fig. 4). This SPS was then fed into the second part of the neural network, which was composed of two convolutional layers, a dense layer, two BGRU layers, and a final feedforward layer for azimuth and elevation estimation by classification. The use of an intermediary SPS output has been proposed to help the neural network learn a representation that has proven to be useful for SSL using traditional methods.

In Adavanne et al. (2019a)[3] and Adavanne et al. (2019b)[5], this intermediary output was no longer used. Instead, the DoA was directly estimated using a block of convolutional layers, a block of BGRU layers, and a feedforward layer. This system is able to localize and detect several sound events even if they overlap in time, provided

they are of different types (e.g., speech and car). This CRNN was the baseline system for Task 3 of the DCASE Challenge in 2019 and 2020. Therefore, it has inspired many other works, and many DCASE Challenge candidate systems were built on the system of Adavanne et al. (2019a)[3] with various modifications and improvements.

For example, Lin and Wang (2019)[150] added Gaussian noise to the input spectrograms to train the network to be more robust to noise. Lu (2019)[152] integrated some additional convolutional layers and replaced the BGRU layers with bidirectional LSTM layers. Leung and Ren (2019)[145] used the same architecture with all combinations of cross-channel power spectra, whereas the replacement of input features with group delays was tested by Nustede and Anemüller (2019). GCC-PHAT features were added as input features by Maruri et al. (2019)[56]. Zhang et al. (2019a)[303] used data augmentation during training and averaged the output of the network for a more stable DoA estimation. Xue et al. (2019)[291] sent the input features separately into different branches of convolutional layers, log-mel, and constant Q-transform features on the one hand, and phase spectrograms and CPS features on the other hand. In [37] they concatenated the log-mel spectrogram and GCC-PHAT features and fed them into two separate CRNNs for SED and DoA estimation. In contrast to the baseline of Adavanne et al. (2019a)[3], more convolutional layers and one single BGRU layer were used. The convolutional part of the DoA network was transferred from the SED CRNN, which was followed by fine-tuning of the DoA branch, labelling this method as two-stage. This led to a notable improvement in localization performance over the DCASE Challenge baseline of Adavanne et al. (2019a)[3]. Small changes to this baseline were also tested by Pratik et al. (2019)[206], such as the use of Bark-scale spectrograms as input features, the modification of the activation function or pooling layers, and the use of data augmentation, resulting in noticeable improvements for some experiments.

The same baseline neural architecture of Adavanne et al. (2019a)[3] was used by Kapka and Lewandowski (2019)[119], with one separate (but identical, except for the output layer) CRNN instance for each subtask: source counting (up to two sources), DoA estimation of source 1 (if applicable), DoA estimation of source 2 (if applicable), and sound type classification. The authors showed that their method was more efficient than the baseline. Krause and Kowalczyk (2019)[132] explored different manners of splitting the SED and DoA estimation tasks in a CRNN. While some configurations showed an improvement in SED, the localization accuracy was below the baseline for the reported experiments. Park et al. (2019b)[194] investigated a combination of a gated linear unit (GLU, a convolutional block with a gated mechanism) and a trellis network (containing convolutional and recurrent layers, see the paper by Bai et al. (2019)[16] for details), yielding better results than the baseline. The authors extended this work for the DCASE 2020 Challenge by improving the overall architecture and investigating other loss functions (Park et al., 2020)[196]. A non-direct DoA estimation scheme was also derived by Grondin et al. (2019)[89], who estimated the TDoA using a CRNN, from which they inferred the DoA.

We also found propositions of CRNN-based systems in the 2020 edition of the DCASE Challenge. Singla et al. (2020)[245] used the same CRNN as in the baseline of Adavanne et al. (2019a)[3], except that they did not use two separated output branches for SED and DoA estimation. Instead, they concatenated the SED output with the output of the previous layer to estimate the DoA. Song (2020)[248] used

separated neural networks similar to the one of Adavanne et al. (2019a)[3] to address NoS estimation and DoA estimation in a sequential way. Multiple CRNNs were trained by Tian (2020)[259]: one to estimate the NoS (up to two sources), another to estimate the DoA assuming one active source, and another (same as the baseline) to estimate the DoAs of two simultaneously active sources. Cao et al. (2020) designed an end-to-end CRNN architecture to detect and estimate the DoA of possibly two instances of the same sound event. The addition of 1D convolutional filters was investigated by Ronchini et al. (2020)[224] to exploit the information along the feature axes. Sampathkumar and Kowerko (2020)[231] augmented the baseline system of Adavanne et al. (2019a)[3] by providing the network with more input features (log-mel spectrograms, GCC-PHAT, and intensity vector).

Independently of the DCASE Challenge, the CRNN of Adavanne et al. (2019a) was adapted by Comminiello et al. (2019)[53] to receive quaternion FOA input features, which slightly improved the CRNN performance. Perotin et al. proposed using a CRNN with bidirectional LSTM layers on the FOA pseudo-intensity vector to localize one (Perotin et al., 2018b)[201] or two (Perotin et al., 2019b)[202] speakers. They showed that this architecture achieves very good performance in simulated and real reverberant environments with static speakers (both types of input features are discussed in Sec. V). This work was extended by Grumiaux et al. (2021a)[90], who obtained a substantial improvement in performance over the CRNN of Perotin et al. (2019b)[202] by adding more convolutional layers with less max-pooling, to localize up to three simultaneous speakers.

Non-square convolutional filters and a unidirectional LSTM layer were used in the CRNN architecture of Li et al. (2018)[146]. Xue et al. (2020)[290] presented a CRNN with two types of input features: the phase of the CPS and the signal waveforms. The former was first processed by a series of convolutional layers before being concatenated with the latter. Another improvement of the network of Adavanne et al. (2019a)[4] was proposed by Komatsu et al. (2020)[130], who replaced the classical convolutional blocks with GLUs, based on the hypothesis that GLUs are better suited for extracting relevant features from phase spectrograms. This has led to a notable improvement of localization performance compared to the baseline of Adavanne et al. (2019a)[3]. Bohlender et al. (2021)[26] proposed an extension of the system of Chakrabarty and Habets (2019b)[41], in which LSTMs and temporal convolutional networks (TCNs) replaced the last dense layer of the former architecture. A TCN was made of successive 1D dilated causal convolutional layers with increasing dilated factors (Lea et al., 2017)[141]. The authors showed that taking the temporal context into account with such temporal layers actually improves localization accuracy.

Finally, we can mention the original approach of Nguyen et al. (2020c)[183] in which a two-step hybrid approach with two CRNNs is used: In the first step, a first CRNN is used for SED and a single-source histogram-based (conventional) method is used for DoA estimation. In the second step, a second CRNN-based network, referred to as a sequence matching network (SMN), is used to match the estimated sequences from the SED and DoA branches. This approach is motivated by the fact that overlapping sounds often have different onsets and offsets, and by matching the outputs of the two branches, an estimated DoA can be associated with the corresponding sound class. This approach was extended to localize moving sources in the framework of the DCASE 2020 Challenge, by adapting the resolution of the

azimuth and elevation histograms and by using an ensemble of SMNs (Nguyen et al., 2020b)[182].

Residual Neural Networks Pujol et al. (2019, 2021)[208, 209] introduced the integration of residual connections alongside 1D dilated convolutional layers featuring increasing dilation factors in their work. They utilized the multichannel waveform as the network input, followed by the segmentation of the architecture into multiple subnetworks, each containing dilated convolutional layers functioning as filter banks.

In another study, Ranjan et al. (2019)[212] combined a modified ResNet architecture (He et al., 2016)[101] with recurrent layers for Single-Channel Acoustic Event Localization and Detection (SELD). This combination exhibited a notable reduction in Direction of Arrival (DoA) error by more than 20° in comparison to the baseline model of Adavanne et al. (2019a)[3]. Similarly, Bai et al. (2021)[207] adopted the ResNet model (He et al., 2016)[101], followed by two Gated Recurrent Unit (GRU) layers and two fully-connected layers for SELD.

Kujawski et al. (2019)[136] applied the original ResNet architecture to address the single-source localization problem.

Naranjo-Alcazar et al. (2020)[178] proposed an architecture, particularly interesting for the DCASE 2020 Challenge, which featured residual connections. Before the recurrent layers, comprising two Bidirectional Gated Recurrent Unit (BGRU) layers, three successive residual blocks processed the input features. These residual blocks included two residual convolutional layers, followed by a squeeze-excitation module (Hu et al., 2020)[111], aiming to enhance the modeling of interdependencies among input feature channels compared to conventional convolutional layers. Sundar et al. (2020)[252] also employed similar squeeze-excitation mechanisms for multi-source localization. Another combination of a residual network with squeeze-excitation blocks was presented by Huang and Perez (2021)[112], who implemented it within the framework of a sample-level Convolutional Neural Network (CNN) (Lee et al., 2017)[143]. These blocks were subsequently followed by two Conformer blocks (details in the next subsection). The motivation behind this fusion of diverse models stemmed from their observed effectiveness in other audio processing tasks, such as Sound Source Localization (SED).

Shimada et al. (2020b, 2020a)[244, 243] adapted the MMDenseLSTM architecture, originally proposed by Takahashi et al. (2018)[253] for sound source separation, for the SELD problem. This architecture comprised a series of blocks featuring convolutions and recurrent layers with residual connections. It exhibited strong performance in the DCASE 2020 Challenge compared to other participants.

Wang et al. (2020)[279] pursued an ensemble learning approach involving various variants of residual neural networks and recurrent layers to estimate DoA. Their approach resulted in the highest performance achieved in the DCASE 2020 Challenge.

Guirguis et al. (2020)[94] devised a neural network that incorporated a Temporal Convolutional Network (TCN) in addition to traditional 2D convolutions and residual connections. Instead of relying on recurrent layers, the architecture employed TCN blocks, which consisted of several residual blocks, including a 1D dilated convolutional layer with an increasing dilation factor. This alteration not

only improved SELD performance slightly compared to the baseline of Adavanne et al. (2019a) but also made the hardware implementation of the network more efficient.

Yasuda et al. (2020)[294] took an indirect approach by leveraging a Convolutional Recurrent Neural Network (CRNN) with residual connections for Direction of Arrival (DoA) estimation using a First-Order Ambisonics (FOA) pseudo-intensity vector input. They initially employed a CRNN to remove the reverberant component of the FOA pseudo-intensity vector. Subsequently, another CRNN was used to estimate a Time-Frequency (TF) mask, which was applied to attenuate TF bins with significant noise levels. Finally, the source DoA was directly estimated from the dereverberated and denoised pseudo-intensity vector.

Attention- based NN The application of attention models has proliferated across a diverse array of Deep Learning (DL) applications, including Sound Source Localization (SSL). In the context of the DCASE 2020 Challenge, Phan et al. (2020a,b)[203] introduced an attention-based neural system. Their architecture comprised multiple convolutional layers, followed by a Bidirectional Gated Recurrent Unit (BGRU), succeeded by a self-attention layer that inferred the activity and Direction of Arrival (DoA) for distinct sound events at each time step. Schymura et al. (2020)[239] introduced an attention mechanism after the recurrent layers of a Convolutional Recurrent Neural Network (CRNN) to estimate sound source activity and azimuth/elevation.

The integration of attention demonstrated enhanced utilization of temporal information for Sound Event Localization and Detection (SELD) compared to the baseline proposed by Adavanne et al. (2019a)[3]. Mack et al. (2020)[161] extended Chakrabarty and Habets' (2019b)[41] system through attention mechanisms, utilizing it to estimate binary masks that emphasize frequency bins where the target source predominates. The initial attention stage is positioned after the input layer (akin to Chakrabarty and Habets, 2019b)[41], employing phase spectrograms as inputs. The second attention stage operates after new features have been extracted via convolutional layers. Adavanne et al. (2021)[6] integrated a self-attention layer following a GRU to estimate the association matrix, facilitating predictions and reference matching. This solution effectively addressed the optimal assignment problem and yielded substantial improvements in localization accuracy.

The concept of Multi-head self-attention (MHSA), entailing the simultaneous application of several Transformer-like attention models (Vaswani et al., 2017)[266], has also spurred the development of methods in Sound Source Localization (SSL). In the DCASE 2021 Challenge, Emmanuel et al. (2021)[77] harnessed an MHSA layer immediately after several convolution modules meticulously designed to capture diverse spectral characteristics. Yalta et al. (2021) proposed leveraging the entire encoder segment of the Transformer architecture, in addition to multiple convolutional layers, for extracting features from input data. Wang et al. (2021)[293] adapted the Conformer architecture, initially conceived by Gulati et al. (2020)[95] for automatic speech recognition, to cater to SSL. This architectural composition encompasses a feature extraction module built upon ResNet and an MHSA module dedicated to acquiring local and global context representations. The authors showcased the advantages of a specific data augmentation technique applied to this model. Zhang et al. (2021)[305] also embraced this architecture within the context of the DCASE 2021 Challenge.

Conformer blocks were also woven into the framework proposed by Huang and Perez (2021)[112]. In this arrangement, Conformer blocks succeed in a sample-level Convolutional Neural Network (CNN) enriched with residual connections and squeeze-excitation. Likewise, a Conformer block found its place in the architecture devised by Rho et al. (2021)[219] for Sound Event Localization and Detection (SELD), positioned after convolutional and fully-connected layers, and preceding Bidirectional Gated Recurrent Unit (BGRU) layers. Cao et al. (2021)[36] introduced an 8-head attention layer after a series of convolutional layers, enabling the tracking of source location predictions over time for different sources (with a maximum of two sources in their experiments).

Schymura et al. (2021)[238] employed three 4-head self-attention encoders along the temporal axis after a sequence of convolutional layers. This configuration was employed before estimating the activity and location of various sound events, resulting in performance improvements over the baseline established by Adavanne et al. (2019a)[3] in the DCASE Challenge. Similarly, Xinghao et al. (2021)[251] substituted the conventional convolutional layers of the baseline with a combination of adaptive convolutional layers, leveraging dilated convolutions with distinct dilation factors, along with attention blocks. Another exemplar of an MHSA-based Transformer model for SSL can be found in the work of Park et al. (2021a)[195], where a pre-trained model is fine-tuned via transfer learning. The output sequence corresponding to each 3-second segment of input data is averaged to yield a DoA estimation. Sudarsanam et al. (2021)[250] enhanced the CRNN baseline introduced by Adavanne et al. (2019a)[3] with a set of several MHSA blocks followed by fully-connected layers. Their analysis delved into the impact of the number and dimensions of MHSA blocks (optimal at 2) and the number of heads (optimal at 8), as well as the effects of positional embedding, normalization layers, and residual connections.

Additionally, Grumiaux et al. (2021b)[92] demonstrated the substitution of recurrent layers within a CRNN with self-attention encoders, leading to a considerable reduction in computation time. Furthermore, the adoption of MHSA brought about a marginal improvement in localization performance when contrasted with the baseline CRNN architecture established by Perotin et al. (2019b)[202] for the specific task of multiple speaker localization.

The utilization of cross-modal attention (CMA) models in the realm of Sound Source Localization (SSL), as proposed by Lee et al. (2021b)[144]. A CMA model represents an extension of self-attention, incorporating two data streams instead of one, a concept originally featured in the Transformer decoder (Vaswani et al., 2017)[266]. Lee et al. (2021b)[144] employed two distinct Convolutional Neural Network (CNN) blocks for Sound Event Detection (SED) and Direction of Arrival (DoA) estimation, resulting in separate SED and DoA embeddings. This diverges from most DCASE (Detection and Classification of Acoustic Scenes and Events) candidate systems, where the initial blocks are shared between SED and DoA estimation. Subsequently, these embeddings are amalgamated, initially through a weighted linear combination, and then through a more intricate alignment process facilitated by two mirrored CMA models. Ultimately, the SED and DoA outputs of the CMA modules are fed into three parallel fully-connected networks to obtain the final estimations. This partitioning is necessitated by the nature of the DCASE 2021 Challenge SELD

Task, where up to three sources can be concurrently active.

In a broader context, it's observable that attention modules, particularly Multi-head Self-Attention (MHSA), exhibit a tendency to supplant recurrent units in contemporary SSL Deep Neural Networks (DNNs). This shift aligns with the groundbreaking notion presented in "Attention is All You Need" by Vaswani et al. (2017)[266]. This transition is driven by the capacity of attention modules to effectively capture long-term dependencies while maintaining a reduced computational overhead, and their aptitude for harnessing parallel computations, particularly during the training phase.

Encoder-Decoder Neural Networks

1. **Autoencoder (AE)**: A notable AE-based technique was introduced by Huang et al. (2020)[113], encompassing an ensemble of AEs trained to replicate the multichannel input signal at the output. Each candidate source position was assigned a dedicated AE. As common latent information across channels corresponds to the dry signal, each encoder approximates the signal's deconvolution from a particular microphone. Localization is achieved by identifying the AE with the most coherent latent representation, presuming the source aligns with the assumed position. However, the model's generalization capability to unseen source positions and acoustic conditions remains uncertain.

   Le Moing et al. (2020)[139] presented an AE with an array of convolutional and transposed convolutional layers, estimating potential source activity for subregions within the (x, y) plane grid, enabling the detection of multiple sources. Different output types (binary, Gaussian-based, and binary followed by regression refinement) were evaluated, each exhibiting promising outcomes on both simulated and real data. This work was expanded upon in Le Moing et al. (2021)[138], introducing adversarial training to enhance network performance on real data and unseen microphone arrays during unsupervised training. An explicit transformation layer was introduced to impart network invariance to microphone array layouts. He et al. (2021b)[105] proposed an encoder-decoder architecture involving a multichannel waveform fed into a filter bank with learnable parameters. A 1D convolutional encoder-decoder network then processed the filter bank output, with separate branches for Sound Event Detection (SED) and DoA estimation.

   Wu et al. (2021b)[287] introduced an encoder-decoder structure featuring a single encoder followed by two distinct decoders. Signals from various microphone arrays were transformed into Short-Time Fourier Transform (STFT) domain and arranged into a 4D tensor. This tensor underwent encoding through convolutional layers and residual blocks, followed by decoding through two separate decoders. The first decoder produced probabilities of source presence for each candidate (x, y) region, while the second incorporated range compensation for increased robustness. A similar encoder-decoder approach was employed by Wu et al. (2021a)[286] for the 2D image mapping method. This architecture encompassed convolutional layers in the encoder and transposed convolutional layers in the decoder, aligning with image mapping applications in computer vision.

   Vera-Diaz et al. (2020)[273] presented an indirect application of an AE, utilizing convolutional and transposed convolutional layers to estimate Time-Difference

of Arrival (TDoA) from Generalized Cross-Correlation (GCC)-based input features. The concept relied on the encoder-decoder's ability to diminish input data dimensionality, compelling the decoder to produce a smoother TDoA version. This technique outperformed the conventional GCC-PHAT method in experiments. An extension was developed for dual-source scenarios (Vera-Diaz et al., 2021)[272].

2. **Variational Autoencoder (VAE)**: Bianco et al. (2020)[25] are credited with the pioneering application of a VAE in the context of SSL. Their VAE, comprised of convolutional layers, was trained to generate the phase of inter-microphone Relative Transfer Functions (RTFs), concurrently with a classifier estimating the speaker's DoA based on RTF phases. The significance of using a VAE stems from its generative model nature, originally designed for unsupervised learning. In this instance, it operates in a semi-supervised setup by employing an extensive dataset of unlabeled RTF data alongside a limited set of labeled data (consisting of RTF values and corresponding DoA labels). In this constrained labeled dataset configuration, this approach demonstrated superiority over SRP-PHAT-based techniques and supervised CNNs in reverberant conditions. An extension of this research has been subsequently presented in Bianco et al. (2021)[24], introducing enhanced network architectures and more realistic acoustic scenarios.

3. **U-Net Architecture**: In the realm of SSL and DoA estimation, numerous studies have drawn inspiration from the original U-Net concept. Chazan et al. (2019)[44] harnessed this architecture to predict individual TF masks for each DoA consideration, associating a specific DoA with each TF bin. Ultimately, these spectral masks were applied to source separation tasks. Hammer et al. (2021)[96] extended this system to accommodate multiple moving speakers. Jenrungrot et al. (2020)[116] introduced a joint localization and separation mechanism based on a U-Net architecture. This implementation employed 1D convolutional layers and Gated Linear Units (GLUs), utilizing the multichannel raw waveform input along with an angular window to enhance separation in designated zones. If the network's output on the window is void, no source is detected; otherwise, the process repeats with a narrower angular window until reaching 2°. This system demonstrated favourable outcomes with both synthetic and real-world reverberant data containing up to eight speakers.

   For the DCASE 2020 Challenge, Patel et al. (2020)[198] proposed a U-Net with several Bidirectional Gated Recurrent Units (BGRUs) within the convolutional blocks for SELD. The ultimate transposed convolutional layer of this U-Net generates a single-channel feature map per sound event, representing its activity and DoA for all frames. This approach showcased enhancements over the Adavanne et al. (2019a)[3] baseline in terms of DoA error. Comanducci et al. (2020a)[52] integrated a U-Net architecture into the second segment of their proposed neural network to estimate source coordinates (x, y). The initial section, comprising convolutional layers, learns to map Generalized Cross-Correlation with Phase Transform (GCC-PHAT) features to ray space (an intermediate representation employing linear patterns, as defined by Bianchi et al., 2016)[23], which serves as input for the U-Net structure.

10 Performance Metrics

For every task described in the last section, there is one or more metrics, to measure the quality of the algorithm used to solve the considered task.

10.1 WER: Word Error Rate

Word error rate (WER)[174] is a common metric of the performance of an automatic speech recognition system. The general difficulty of measuring performance lies in the fact that the recognized word sequence can have a different length from the reference word sequence (supposedly the correct one). The WER [284, 175] is derived from the Levenshtein distance [171], working at the word level instead of the phoneme level. The WER is a valuable tool for comparing different systems as well as for evaluating improvements within one system. This kind of measurement, however, provides no details on the nature of translation errors and further work is therefore required to identify the main source(s) of error and to focus any research effort. This problem is solved by first aligning the recognized word sequence with the reference (spoken) word sequence using dynamic string alignment. Examination of this issue is seen through a theory called the power law that states the correlation between perplexity and word error rate. Word error rate can then be computed as:

$$WER = (S + D + I)/N = (S + D + I)/(S + D + C) \tag{4.12}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, N is the number of words in the reference ($N = S + D + C$).

This value indicates the average number of errors per reference word. The lower the value, the better the performance of the ASR system with a WER of 0 being a perfect score.

10.2 CER: Character Error Rate

Character error rate (CER) is a common metric of the performance of an automatic speech recognition system. CER is similar to Word Error Rate (WER), but operates on character instead of word[55]. Please refer to docs of WER for further information. Character error rate can be computed as:

$$CER = (S + D + I)/N = (S + D + I)/(S + D + C) \tag{4.13}$$

where:
- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of correct characters,
- N is the number of characters in the reference ($N = S + D + C$).

CER's output is not always a number between 0 and 1, in particular when there is a high number of insertions. This value is often associated to the percentage of characters that were incorrectly predicted. The lower the value, the better the performance of the ASR system with a CER of 0 being a perfect score.

10.3 DER: Diarization Error Rate

Diarization Error Rate The main metric that is used for speaker diarization experiments is the Diarization Error Rate (DER) as described and used by NIST in the RT evaluations (NIST Fall Rich Transcription on meetings 2006 Evaluation Plan, 2006). It is measured as the fraction of time that is not attributed correctly to a speaker or to non-speech. To measure it, a script names MD-eval-v12.pl (NIST MD-eval-v21 DER evaluation script, 2006), developed by NIST, was used.

As per the definition of the task, the system hypothesis diarization output does not need to identify the speakers by name or definite ID, therefore the ID tags assigned to the speakers in both the hypothesis and the reference segmentation do not need to be the same. This is unlike the non-speech tags, which are marked as non labelled gaps between two speaker segments, and therefore do implicitly need to be identified.

The evaluation script first does an optimum one-to-one mapping of all speaker label ID between hypothesis and reference files. This allows the scoring of different ID tags between the two files. The Diarization Error Rate score is computed as

$$DER = \frac{\sum_{s=1}^{S} \mathrm{dur}(s) \cdot (\max(N_{ref}(s), N_{nyp}(s)), N_{hyp}(s)) - N_{correct}(s))}{\sum_{s=1}^{S} \mathrm{dur}(s) \cdot N_{ref}} \tag{4.14}$$

where S is the total number of speaker segments where both reference and hypothesis files contain the same speaker/s pair/s. It is obtained by collapsing together the hypothesis and reference speaker turns. The terms $N_{ref}(s)$ and $N_{sys}(s)$ indicate the number of speaker speaking in segment s, and $N_{correct}(s)$ indicates the number of speakers that speak in segment s and have been correctly matched between reference and hypothesis. Segments labelled as non-speech are considered to contain 0 speakers. When all speakers/non-speech in a segment are correctly matched the error for that segment is 0.
The DER error can be decomposed into the errors coming from the different sources, which are:

Speaker error: percentage of scored time that a speaker ID is assigned to the wrong speaker. This type of error does not account for speakers in overlap not detected or any error coming from non-speech frames. It can be written as

$$E_{Spkr} = \frac{\sum_{s=1}^{S} \mathrm{dur}(s) \cdot (\min(N_{ref}(s), N_{hyp}(s)) - N_{correct}(s))}{T_{score}} \tag{4.15}$$

where $T_{score} = \sum_{s=1}^{S}$ and $dur(s) \cdot N_{ref}$ is the total scoring time, in the denominator in eq. 4.12.

False alarm speech: percentage of scored time that a hypothesized speaker is labelled as a non-speech in the reference. It can be formulated as

$$E_{FA} = \frac{\sum_{s=1}^{S} \mathrm{dur}(s) \cdot (N_{hyp}(s) - N_{ref}(s))}{T_{score}} \quad \forall\ (N_{hyp}(s) - N_{ref}(s)) > 0 \tag{4.16}$$

computed only over segments where the reference segment is labelled as non-speech.

Missed speech: percentage of scored time that a hypothesized non-speech segment corresponds to a reference speaker segment. It can be expressed as

$$E_{MISS} = \frac{\sum_{s=1}^{S} \mathrm{dur}(s) \cdot (N_{ref}(s) - N_{hyp}(s))}{T_{score}} \quad \forall\ (N_{ref}(s) - N_{hyp}(s)) > 0 \quad (4.17)$$

computed only over segments where the hypothesis segment is labelled as non-speech.

Overlap speaker: percentage of scored time that some of the multiple speakers in a segment do not get assigned to any speaker. This errors usually fuses either into the E_{MISS} or E_{FA}, depending on wether it is the reference or the hypothesis containing non assigned speakers. If multiple speakers appear in both the reference and the hypothesis the error produced belongs to E_{spkr}. Given all possible errors one can rewrite equation 4.12 as

$$DER = E_{spkr} + E_{MISS} + E_{FA} + E_{ovl} \quad (4.18)$$

When evaluating performance, a collar around every reference speaker turn can be defined which accounts for inexactitudes in the labelling of the data. It was estimated by NIST that a ±250ms collar could account for all these differences. When there is people overlapping each other in the recording it is stated so in the reference file, with as many as 5 speaker turns being assigned to the same time instant. As pointed out in the denominator of eq. 4.18, the total evaluated time includes the overlaps. Errors produced when the system does not detect any or some of the multiple speakers in overlap count as missed speaker errors.

Once the performance is obtained for each individual meeting excerpt, the time weighted average is done among all meetings in a given set to obtain an overall average score. The scored time is the one used for such weighting, as it indicates the total (overlapped speaker included) time that has been evaluated in each excerpt.

10.4 PESQ

Previous objective speech quality assessment models, such as bark spectral distortion (BSD), the perceptual speech quality measure (PSQM), and measuring normalizing blocks (MNB), have been found to be suitable for assessing only a limited range of distortions. A new model has therefore been developed for use across a wider range of network conditions, including analogue connections, codecs, packet loss and variable delay.

Known as **perceptual evaluation of speech quality (PESQ)**, it is the result of integration of the perceptual analysis measurement system (PAMS) and PSQM99, an enhanced version of PSQM. PESQ is expected to become a new ITU-T recommendation P.862, replacing P.861 which specified PSQM and MNB.
Perceptual Evaluation of Speech Quality (PESQ) is a family of standards comprising a test methodology for automated assessment of the speech quality as experienced by a user of a telephony system. It was standardized as Recommendation ITU-T P.862[221] in 2001. PESQ is used for objective voice quality testing by phone manufacturers, network equipment vendors and telecom operators. Its usage requires

a license. The first edition of PESQ's successor POLQA (Recommendation ITU-T P.863[2]) entered into force in 2011.

10.5 SDR: Source-to-Distortion Ratio

Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR), and Source-to-Artifact Ratio (SAR) are, to date, the most widely used methods for evaluating a source separation or enhancement system's output.
An estimate of a Source $\hat{s}_i$ is assumed to actually be composed of four separate components,

$$\hat{s}_i = s_{target} + e_{interf} + e_{noise} + e_{artif}, \tag{4.19}$$

where s_{target} is the true source, and e_{interf}, e_{noise}, and e_{artif} are error terms for interference, noise, and added artefacts, respectively. The actual calculations of these terms is quite complex, so we refer the curious reader to the original paper [275] for their exact calculation.

Using these four terms, we can define our measures. All of the measures are in terms of decibels (dB), with higher values being better. To calculate they require access to the ground truth isolated sources and are usually calculated on a signal that has been divided into short windows of a few seconds long.

$$SDR = 10\log_{10}\left(\frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2}\right) \tag{4.20}$$

10.6 SI-SDR

The Signal-to-Distortion Ratio (SI-SDR) is a metric used to quantify the quality of a source signal compared to the distortion introduced by a separation or denoising process. It is expressed as follows:

$$\text{SI-SDR} = 10 \cdot \log_{10}\left(\frac{\sum_{n=1}^{N} s_n^2}{\sum_{n=1}^{N}(s_n - \hat{s}_n)^2}\right) \tag{4.21}$$

In this formula, s_n represents the original source signal, and $\hat{s}_n$ represents the estimated or separated signal. The summation is taken over all time or sample points. The result is scaled by a factor of 10 to express SI-SDR in decibels (dB). SI-SDR measures the relative strength of the source signal compared to the distortion or interference caused by the separation or denoising process. Higher SI-SDR values indicate better signal quality, with positive values indicating that the estimated signal (s^n) is more similar to the original source signal (s_n) than the distortion. Conversely, negative values suggest that the distortion is stronger than the source signal.

10.7 SI-SDRi: Scale-Invariant Source-to-Distortion Ratio

It measures the improvement in SI-SDR between the estimated and reference signals, taking into account the amplitude scaling of the signals. SI-SDRi is a modification of the SI-SDR metric, which is a modification of the SDR (Signal-to-Distortion Ratio) metric and has been shown to be more robust and reliable than the

SDR metric, especially in single-channel separation tasks. The formula for SI-SDRi is:

$$SI-SDRi = 10 * log_{10}(\frac{(SI-SDR_{estimated} - SI-SDR_{reference})^2}{(SI-SDR_{reference})^2}) \tag{4.22}$$

where $SI-SDR_{estimated}$ is the $SI-SDR$ of the estimated signal and $SI-SDR_{reference}$ is the $SI-SDR$ of the reference signal. The $SI-SDRi$ metric is used to evaluate the quality improvement of the estimated signal compared to the reference signal. A higher $SI-SDRi$ score indicates a better quality improvement of the estimated signal. The SI-SDRi metric is a modification of the SI-SDR metric, which is a modification of the SDR (Signal-to-Distortion Ratio) metric, and has been shown to be more robust and reliable than the SDR metric, especially in single-channel separation tasks 1 2 3 4 5 . [140]

10.8 SAR: Source-to-Artifact Ratio

This is usually interpreted as the amount of unwanted artifacts a source estimate has with relation to the true source.

$$SAR = 10\log_{10}\left(\frac{\|s_{target} + e_{interf} + e_{noise}\|^2}{\|e_{artif}\|^2}\right) \tag{4.23}$$

10.9 SIR: Source-to-Interference Ratio

This is usually interpreted as the amount of other sources that can be heard in a source estimate. This is most close to the concept of "bleed", or "leakage".

$$SIR = 10\log_{10}\left(\frac{\|s_{target}\|^2}{\|e_{interf}\|^2}\right) \tag{4.24}$$

10.10 SNR: Signal to Noise Ratio

This is not used as widely, but does appear sometimes in source separation:

$$SNR = 10\log_{10}\left(\frac{\|s_{target}\|^2}{\|s_{target} - \hat{s}\|^2}\right) \tag{4.25}$$

where $\hat{s}$ is the estimate of s_{target}.

10.11 SI-SNR: Scale Invariant- Signal to Noise Ratio

The objective of training the end-to-end system is maximizing the scale-invariant source-to-noise ratio (SI-SNR), which has commonly been used as the evaluation metric for source separation replacing the standard source-to-distortion ratio (SDR)[275]. SI-SNR is defined in [154] as:

$$\begin{cases} s_{target} := \frac{\langle \hat{s}, s \rangle s}{\|s\|^2} \\ e_{noise} := \hat{s} - s_{target} \\ SI-SNR = 10\log_{10}\left(\frac{\|s_{target}\|^2}{\|e_{noise}\|^2}\right) \end{cases} \tag{4.26}$$

where $\hat{s} \in \mathbb{R}^{1xT}$ and $s \in \mathbb{R}^{1xT}$ are the estimated and original clean sources, respectively, and $s = \langle \hat{s}, s \rangle$ denotes the signal power. Scale invariance is ensured by normalizing $\hat{s}$ and s to zero-mean prior to the calculation. Utterance-level permutation invariant training (uPIT) is applied during training to address the source permutation problem.

10.12 Accuracy

In the vast field of Machine Learning, the general focus is to predict an outcome using the available data. The prediction task is also called a "classification problem" when the outcome represents different classes, otherwise is called a "regression problem" when the outcome is a numeric measurement.
As regards classification, the most common setting involves only two classes, although there may be more than two. In this last case, the issue changes his name and is called "multi-class classification".
From an algorithmic standpoint, the prediction task is addressed using the state of the art mathematical techniques. There are many different solutions, however each one shares a common factor: they use available data (X variables) to obtain the best prediction $\hat{Y}$ of the outcome variable Y .
In Multi-class classification, we may regard the response variable Y and the prediction $\hat{Y}$ as two discrete random variables: they assume values in $\{1, ..., K\}$ and each number represents a different class.

The algorithm comes up with the probability that a specific unit belongs to one possible class, then a classification rule is employed to assign a single class to each individual. The rule is generally very simple, the most common rule assigns a unit to the class with the highest probability. A classification model gives us the probability of belonging to a specific class for each possible units. Starting from the probability assigned by the model, in the two-class classification problem a threshold is usually applied to decide which class has to be predicted for each unit.
True Positive: A true positive is an outcome where the model correctly predicts the positive class.
True Negative: True negative is an outcome where the model correctly predicts the negative class.
False Positive: A false positive is an outcome where the model incorrectly predicts the positive class.
False Negative: False negative is an outcome where the model incorrectly predicts the negative class. [170]
There are many metrics that come in handy to test the ability of any multi-class classifier and they turn out to be useful for: i) comparing the performance of two different models, ii) analysing the behaviour of the same model by tuning different parameters. [87]
Accuracy is one of the most popular metrics in multi-class classification and it is directly computed from the confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.27}$$

The formula of the Accuracy considers the sum of True Positive and True Negative elements at the numerator and the sum of all the entries of the confusion matrix at the denominator. True Positives and True Negatives are the elements correctly

classified by the model and they are on the main diagonal of the confusion matrix, while the denominator also considers all the elements out of the main diagonal that have been incorrectly classified by the model. In simple words, consider to choose a random unit and predict its class, Accuracy is the probability that the model prediction is correct.

10.13 F-score

In statistical analysis of binary classification, the F-score or F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive. Precision is also known as positive predictive value, and recall is also known as sensitivity in diagnostic binary classification.
The F1 score is the harmonic mean of the precision and recall. The more generic F_β score applies additional weights, valuing one of precision or recall more than the other.
The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero. The F1 score is also known as the Sørensen–Dice coefficient or Dice similarity coefficient (DSC).

$$F_{score} = 2\frac{precision * recall}{precision + recall} \tag{4.28}$$

where $precision = \frac{TruePositive}{TruePositive+FalsePositive}$ and $recall = \frac{TruePositive}{TruePositive+FalseNegative}$

10.14 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is a measure of the average size of the errors in a collection of predictions, without taking their direction into account. It is calculated as the mean of the absolute differences between the predicted and actual values. MAE is a linear score, meaning all individual differences contribute equally to the mean. It provides an estimate of the size of the inaccuracy, but not its direction (e.g., over or under-prediction) . MAE is commonly used as a performance metric for regression models because it is intuitive, interpretable, resistant to outliers, and offers information about the error size . The formula for calculating MAE is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.29}$$

where n is the number of predictions, y_i is the actual value, and $\hat{y}_i$ is the predicted value 3. MAE is not identical to Root Mean Squared Error (RMSE), although some researchers report and interpret it that way 1. The MAE is conceptually simpler and easier to interpret than RMSE: it is simply the average absolute vertical or horizontal distance between each point in a scatter plot and the $Y = X$ line 1. Furthermore, each error contributes to MAE in proportion to the absolute value of error 1.In Python, the mean_absolute_error() method of the sklearn.metrics module can be used to compute the MAE of a series of predictions 2.

11 Datasets

11.1 LibriSpeech

LibriSpeech [191] is one of the most frequently used open-source speech-to-text corpus. This dataset consists of 1000 h of audiobooks along with their transcriptions. Because of the large magnitude of the collected data, it was divided into three sets. The first set is comprised of 100 h of training data, the second contains 360 h of training data, and the last set has 500 h of training data. The development set and the testing set have 10.8 and 10.1 hours' worth of data, respectively.

11.2 TIMIT

The TIMIT [83] corpus of read speech is designed to provide speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences, where 30% of them are female, and the rest are male speakers. The training set consists of 3.14 h of recording; the rest is divided into the test and development sets respectively. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16kHz speech waveform file for each utterance. Corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI). The speech was recorded at TI, transcribed at MIT and verified and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST). The TIMIT corpus transcriptions have been hand-verified. Test and training subsets, balanced for phonetic and dialectal coverage, are specified. Tabular computer-searchable information is included as well as written documentation.

11.3 CommonVoice

The Common Voice corpus described in [11], is a massively multilingual collection of transcribed speech intended for speech technology research and development. Common Voice is designed for Automatic Speech Recognition purposes but can be useful in other domains (e.g. language identification). To achieve scale and sustainability, the Common Voice project employs crowdsourcing for both data collection and data validation. The most recent release includes 29 languages, and as of November 2019, there are a total of 38 languages collecting data. Over 50,000 individuals have participated so far, resulting in 2,500 hours of collected audio. To our knowledge, this is the largest audio corpus in the public domain for speech recognition, both in terms of the number of hours and the number of languages. As an example use case for Common Voice, they presented speech recognition experiments using Mozilla's DeepSpeech Speech-to-Text toolkit. By applying transfer learning from a source English model, we find an average Character Error Rate improvement of 5.99 +/- 5.48 for twelve target languages (German, French, Italian, Turkish, Catalan, Slovenian, Welsh, Irish, Breton, Tatar, Chuvash, and Kabyle). For most of these languages, these are the first ever published results on end-to-end Automatic Speech Recognition.

11.4 AISHELL

In [34] an open-source Mandarin speech corpus called AISHELL-1 is released. It is by far the largest corpus which is suitable for conducting the speech recognition research and building speech recognition systems for Mandarin. The recording procedure, including audio capturing devices and environments are presented in details. The preparation of the related resources, including transcriptions and lexicon are described. The corpus is released with a Kaldi recipe. Experimental results implies that the quality of audio recordings and transcriptions are promising.

11.5 VoiceBank

In [267] the University of Edinburgh has started the development of a new speech database, the Voice Bank corpus, specifically designed for the creation of personalised synthetic voices for individuals with speech disorders. This corpus already constitutes the largest corpora of British English currently in existence, with more than 300 hours of recordings from approximately 500 healthy speakers. Recordings are continuously being made in order to get the best coverage of the different combinations of regional accents, social classes, age and gender across Britain. This paper describes the motivation and the processes involved in the design and recording of this corpus as well as some analysis of its content. The paper concludes with our future plans to further extend this corpus and to overcome its current limitations.

11.6 WSJ: Wall Street Journal

The DARPA Spoken Language System (SLS) community has long taken a leadership position in designing, implementing, and globally distributing significant speech corpora widely used for advancing speech recognition research. The Wall Street Journal (WSJ) CSR Corpus described in [199]. In contrast to previous corpora, the WSJ corpus provide DARPA its general-purpose English, large vocabulary, natural language, high perplexity, corpus containing significant quantities of both speech data (400 hrs.) and text data (47M words), thereby providing a means to integrate speech recognition and natural language processing in application domains with high potential practical value. This paper presents the motivating goals, acoustic data design, text processing steps, lexicons, and testing paradigms incorporated into the multi-faceted WSJ CSR Corpus.

11.7 WHAM!

Recent progress in separating the speech signals from multiple overlapping speakers using a single audio channel has brought us closer to solving the cocktail party problem. However, most studies in this area use a constrained problem setup, comparing performance when speakers overlap almost completely, at artificially low sampling rates, and with no external background noise. In [283], the researchrs strive to move the field towards more realistic and challenging scenarios. To that end, they created the WSJ0 Hipster Ambient Mixtures (WHAM!) dataset, consisting of two speaker mixtures from the wsj0-2mix[106] dataset combined with real ambient noise samples. The samples were collected in coffee shops, restaurants, and bars in the San Francisco Bay Area, and are made publicly available. In this work the authors benchmark various speech separation architectures and objective functions to evaluate their robustness to noise. While separation performance decreases as a

result of noise, in the paper they still observe substantial gains relative to the noisy signals for most approaches.

11.8 WHAMR!

While significant advances have been made with respect to the separation of overlapping speech signals, studies have been largely constrained to mixtures of clean, near anechoic speech, not representative of many real-world scenarios. Although the WHAM! dataset introduced noise to the ubiquitous wsj0-2mix[106] dataset, it did not include reverberation, which is generally present in indoor recordings outside of recording studios. The spectral smearing caused by reverberation can result in significant performance degradation for standard deep learning-based speech separation systems, which rely on spectral structure and the sparsity of speech signals to tease apart sources. To address this, in [160] it was introduced WHAMR!, an augmented version of WHAM! with synthetic reverberated sources, and provide a thorough baseline analysis of current techniques as well as novel cascaded architectures on the newly introduced conditions.

11.9 LibriMix: An Open-Source Dataset for Generalizable Speech Separation

In recent years, wsj0-2mix has become the reference dataset for single-channel speech separation. Most deep learning-based speech separation models today are benchmarked on it. However, recent studies have shown important performance drops when models trained on wsj0-2mix are evaluated on other, similar datasets. To address this generalization issue, we created LibriMix, an open-source alternative to wsj0-2mix [106], and to its noisy extension, WHAM!. Based on LibriSpeech, LibriMix [57] consists of two- or three-speaker mixtures combined with ambient noise samples from WHAM!. Using Conv-TasNet, we achieve competitive performance on all LibriMix versions. In order to fairly evaluate across datasets, we introduce a third test set based on VCTK for speech and WHAM! for noise. Our experiments show that the generalization error is smaller for models trained with LibriMix than with WHAM!, in both clean and noisy conditions. Aiming towards evaluation in more realistic, conversation-like scenarios, we also release a sparsely overlapping version of LibriMix's test set.

11.10 IEMOCAP

Since emotions are expressed through a combination of verbal and non-verbal channels, a joint analysis of speech and gestures is required to understand expressive human communication. To facilitate such investigations, the paper [35] describes a new corpus named the "interactive emotional dyadic motion capture database" (IEMOCAP), collected by the Speech Analysis and Interpretation Laboratory (SAIL) at the University of Southern California (USC). This database was recorded from ten actors in dyadic sessions with markers on the face, head, and hands, which provide detailed information about their facial expressions and hand movements during scripted and spontaneous spoken communication scenarios. The actors performed selected emotional scripts and also improvised hypothetical scenarios designed to elicit specific types of emotions (happiness, anger, sadness, frustration and neutral state). The corpus contains approximately 12 h of data. The detailed motion

capture information, the interactive setting to elicit authentic emotions, and the size of the database make this corpus a valuable addition to the existing databases in the community for the study and modeling of multimodal and expressive human communication.

Chapter 5

Deep Audio Analyzer

Recently, applications of novel deep learning solutions to forensics investigations have experienced unprecedented growth in interest and obtained results [300, 19, 163, 93, 162, 85], with many researchers developing innovative algorithms and models to solve complex problems. However, reproducing published experiments and results remains a significant challenge due to the programming skills required. This challenge is further compounded by the lack of (or an extremely limited) standardization in the way experiments are conducted. This issue results in a significant amount of time being spent by researchers and practitioners to reproduce previous works and results, which leads to a significant waste of resources. The development of speech-processing technologies has been largely driven by open-source toolkits [114, 114, 227, 213].

However, with the emergence of general-purpose deep learning libraries like TensorFlow [2] and PyTorch [197], more flexible speech recognition frameworks have emerged, such as DeepSpeech [97], RETURNN [302], PyTorch-Kaldi [213], Espresso [280], Lingvo [241], Fairseq [189], ESPnet [282], NeMo [135], Asteroid [193], Speechbrain [217] and hub where scientists load trained models for others to download [54]. While it can be challenging for non-expert users to prototype new deep learning methods, as it requires knowledge of coding and environmental setup.

The objective of this research is to provide a tool that covers in a single framework the three main goals of Authenticity, Enhancement and Interpretation. Exploiting the new technologies and enables users to visualize audio features, evaluate the performance of pre-trained models, and create new audio analysis workflows by combining deep neural network models.

Through the use of Deep Audio Analyzer, users can perform these features without the need to develop any code.

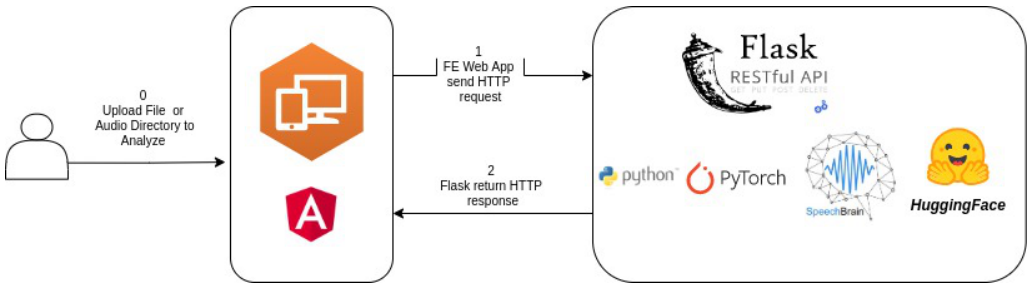


Figure 5.1: **Angular Front End:** The User Interface of Deep Audio Analyzer is divided into pages and components in order to categorize all the functions separately. All the modules of Deep Audio Analyzer are developed on different pages. **Flask Backend:** Deep Learning Audio Analyzer employs a simple software stack (i.e., Python → PyTorch → SpeechBrain → HuggingFace → Flask → Angular) to avoid dealing with too many levels of abstraction. It is developed on top of SpeechBrain and HuggingFace directly, with external APIs that can retrieve the newest model uploaded from the SpeechBrain community and other Companies.

The tool also provides dedicated modules to test state-of-the-art models on customized data and also combine models to create a new deep learning audio processing pipeline, combing for tasks such as Automatic Speech Recognition, Speech Enhancement, Speaker Separation, Speaker Verification and Voice Activity Detection.

Open Source Code of this research project is available at:
`https://github.com/valeriopuglisi/deep-audio-analyzer`

The remainder of this work is organized as follows. In Section 1, we delve into the technologies comprising the architecture of the Audio Analyzer. Section 3 reports the proposed features and modules developed in Deep Audio Analyzer. Section 4 presents the considered experiments and discusses the obtained results. Section 5 concludes the paper and proposes future works.

1 Architecture

The overall Architecture of Deep Learning Audio Analyzer is actually composed of a Backend service where all the artificial intelligence tasks are implemented and a Front-End module as shown in Fig.5.1.

Angular FrontEnd framework is concerned to make easier the development and maintenance of the platform while the Backend Flask RESTful API was chosen because it is fast to develop and is written in Python, which comprises the Artificial Intelligence libraries used to develop Deep Audio Analyzer platform.

2 Audio Features Visualization Module

Through the preprocessing module of Deep Audio Analyzer, it is possible to graphically analyze all the features extracted through the application of the functions present in the librosa library [167] whose functions have been implemented in the Backend of the application (Fig.5.2).

Among others visualization tools include simultaneous presentation of the time waveform, spectrogram, and audio playback, as was shown in Fig. 5.5. This allows a very flexible system for critical listening and visual assessment of signal characteristics, and this capability is highly recommended. [164, 162, 300]

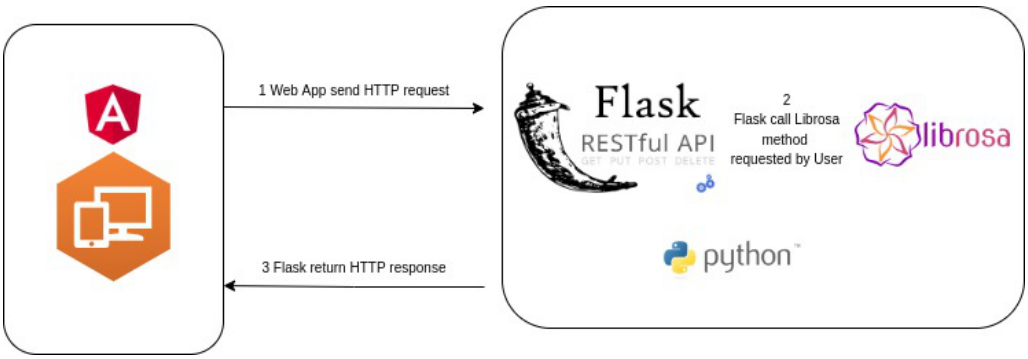


Figure 5.2: Architecture of Audio Feature Visualization Module.

2.1 Preprocessing Audio Features

The developed functions are shown in the fig. 5.3 and explained in the next list :

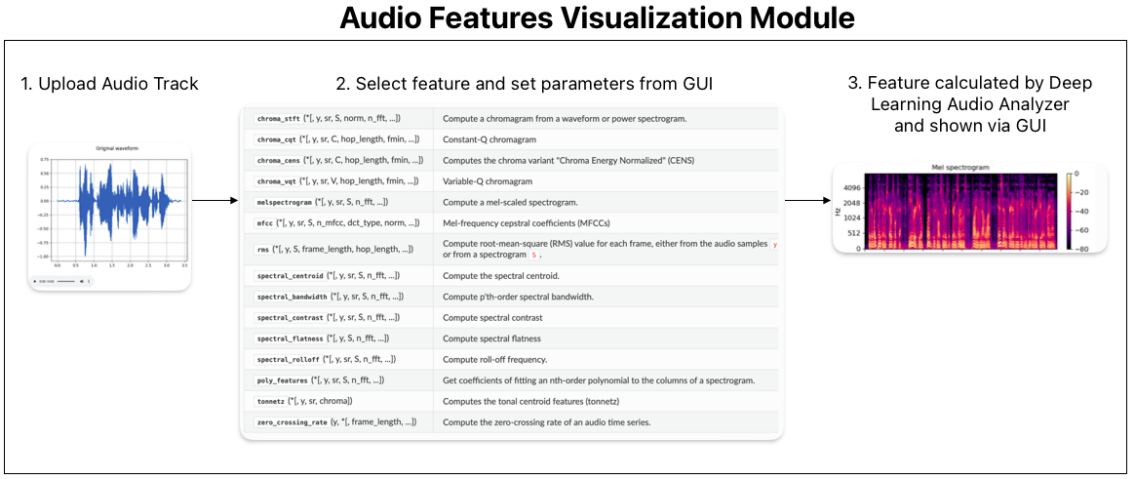


Figure 5.3: Audio Feature Visualization Module.

- **Linear-frequency power spectrogram:** The linear-frequency power spectrogram is an important tool in the field of audio forensics. It represents time on the X-axis, frequency in Hz on a linear scale on the Y-axis, and power in dB [188]. It is used to identify specific events in an audio recording by allowing experts to analyze the spectral characteristics. It also enables voice analysis by studying features such as pitch, formants, and harmonics, which can aid in speaker identification and voice comparison. Furthermore, the spectrogram can reveal hidden artefacts, noise, or disturbances in an audio recording, which can then be mitigated by applying appropriate filtering techniques, thereby enhancing the desired audio content. In audio authentication, the spectrogram can be instrumental in detecting signs of audio tampering or manipulation. The linear frequency power spectrogram of a signal x(t) is computed using the following steps:

  1. Divide the signal into overlapping segments of length N.
  2. Compute the short-time Fourier transform (STFT) of each segment.
  3. Square the magnitude of the STFT to obtain the power spectrum.
  4. Average the power spectra of all segments to obtain the final spectrogram.

The following mathematical formula describes the linear frequency power spectrogram:

$$S(f,t) = \frac{1}{M}\sum_{m=0}^{M-1} |X(f, t - m\tau)|^2 \tag{5.1}$$

where $S(f,t)$ is the linear frequency power spectrogram, f is the frequency, t is the time, $X(f,t)$ is the STFT of the signal, M is the number of segments and τ is the overlap factor. The overlap factor τ controls the time resolution of the spectrogram. A higher overlap factor results in better time resolution, but it also reduces the frequency resolution. The linear frequency power spectrogram is a powerful tool for visualizing the time-frequency distribution of a signal. It can be used to identify the frequencies that are present in the signal and how they change over time.

- **Log-frequency power spectrogram:** Such features can be obtained from a spectrogram by converting the linear frequency axis (measured in Hertz) into a logarithmic axis (measured in pitches). Its logarithmic representation of frequency content enables experts to extract unique voice characteristics, classify sounds, segment audio recordings, enhance transcription accuracy, and detect potential tampering or manipulation. The resulting representation is also called log-frequency spectrogram.

  Log Frequency Power Spectrogram is a time-frequency representation of a signal that is widely used in audio signal processing and analysis [177, 38]. It is a variation of the traditional spectrogram, where the frequency axis is redefined to correspond to the logarithmically spaced frequency distribution of the equal-tempered scale. The logarithmic perception of frequency motivates the use of a time-frequency representation with a logarithmic frequency axis labelled by the pitches of the equal-tempered scale. The formula for computing a log-frequency spectrogram is:

$$Y_{LF}(p,t) = \sum_{k=1}^{K} Y(k,t) \tag{5.2}$$

  where $Y(k,t)$ is the magnitude or power spectrogram of the signal at time t and frequency bin k, K is the number of frequency bins, and $Y_{LF}(p,t)$ is the log-frequency spectrogram at pitch p and time t

- **Chroma STFT:** Chroma STFT features are useful for analyzing the harmonic content of an audio signal and can be used in a variety of applications such as music information retrieval, audio classification, and speech recognition. They provide a way to represent the pitch content of an audio signal in a compact and efficient way and can be used to compare and classify different audio signals based on their harmonic content. Chroma STFT is a useful tool in audio signal processing for analyzing the chromatic content of an audio signal and can be used in a wide range of applications. This implementation is derived from chromagram E [76].

  The Chroma Short Time Fourier Transform (C-STFT) is computed using the following steps [176]:

1. Divide the signal into overlapping segments of length N.
2. Compute the STFT of each segment.
3. Warp the STFT to a logarithmic frequency scale.
4. Sum the STFT magnitude values for each pitch class.
5. Normalize the pitch class values.

The following mathematical formula describes the C-STFT:

$$C(c,t) = \frac{1}{N_c} \sum_{k=0}^{N_f-1} |X(f_k,t)|^2, \text{ where } f_k = f_0 2^{kc/N_c} \tag{5.3}$$

where c is the pitch class, t is the time, $X(f,t)$ is the STFT of the signal, N_c is the number of pitch classes, N_f is the number of frequency bands and f_0 is the lowest frequency band.

- **Chroma CQT :** The Constant-Q chromagram is a type of chroma feature representation commonly used in music analysis and processing. It is based on the Constant-Q transform, which is a frequency-domain transformation that uses a logarithmic frequency scale that approximates the way that humans perceive sound [33]. The Constant-Q chromagram further processes the Constant-Q transform by grouping the spectral information into bins that correspond to specific pitches, using a mapping similar to the way that musical notes are organized in a piano keyboard. The resulting representation is a two-dimensional matrix that shows the energy distribution of each pitch class over time, similar to a spectrogram but with a greater emphasis on the harmonic content of the audio signal. The Constant-Q chromagram is often used as a feature for tasks such as music genre classification, chord recognition, and melody extraction.[33]

  The Chroma CQT (Constant Q Transform) is computed using the following steps:

  1. Divide the signal into overlapping segments of length N.
  2. Compute the Constant Q Transform (CQT) of each segment.
  3. Sum the CQT magnitudes for each pitch class.
  4. Normalize the pitch class values.

  The following mathematical formula describes the Chroma CQT:

$$C(c,t) = \frac{1}{N_c} \sum_{k=0}^{N_f-1} |X_C QT(k,t)|^2, \text{ where } k = f_0 2^{kc/N_c} \tag{5.4}$$

  where $C(c,t)$ is the Chroma CQT, c is the pitch class, t is the time, $X_C QT(k,t)$ is the CQT magnitude at frequency band k and time t, N_c is the number of pitch classes, N_f is the number of frequency bands and f_0 is the lowest frequency band

- **Chroma CENS:** Computes the chroma variant "Chroma Energy Normalized" (CENS)[78]. To compute CENS features, following steps are taken after obtaining chroma vectors using chroma-cqt :

1. L-1 normalization of each chroma vector,
2. Quantization of amplitude based on "log-like" amplitude thresholds,
3. (optional) Smoothing with sliding window.
4. Default window length = 41 frames.

CENS features are robust to dynamics, timbre and articulation, thus these are commonly used in audio matching and retrieval applications .

- **Melspectrogram:** CMel Spectrograms are widely used in speech processing and music processing applications. They have been shown to be effective for tasks such as speech recognition, speaker recognition, and music genre classification [167] w Mel Spectrograms are also used in other fields, such as machine learning and data mining. For example, Mel Spectrograms have been used to develop machine learning models for predicting human emotions from speech data.

  The Mel Spectrogram of a signal x(t) is computed using the following steps:

  1. Divide the signal into overlapping segments of length N.
  2. Compute the short-time Fourier transform (STFT) of each segment.
  3. Map the powers of the STFT coefficients onto the mel scale using a mel filterbank.
  4. Take the logarithm of the mel filterbank outputs to obtain the mel spectrogram.

  The following mathematical formula describes the Mel Spectrogram:

$$S(m,t) = \log\left(\sum_{k=1}^{K} H_m(k)|X(k,t)|^2\right) \tag{5.5}$$

  where $S(m,t)$ is the Mel Spectrogram, m is the mel frequency index, t is the time index, $X(k,t)$ is the STFT coefficient at frequency k and time t, $H_m(k)$ is the m-th mel filterbank coefficient at frequency k and K is the number of mel filterbank coefficients.

- **Mel-frequency spectrogram**: Display of mel-frequency spectrogram coefficients, with custom arguments for mel filterbank construction (default is $f_{max} = sr/2$). Mel-frequency spectrograms are valuable in forensic audio analysis for visualizing and analyzing the characteristics of an audio recording relevant to a legal case. They help identify specific sounds, voices, recording quality, and potential tampering. By extracting features like pitch, spectral content, and temporal characteristics, it enables comparisons between different recordings to determine their common source. They are useful for speaker identification, voice matching, and background noise analysis. Mel-frequency spectrograms provide a perceptually relevant representation of audio and allow forensic analysts to determine important details about the origin and authenticity of recordings.

- **Mel-frequency cepstral coefficients (MFCCs)**: Mel-frequency cepstral coefficients (MFCCs) are a type of feature representation commonly used in audio signal processing and analysis [169], particularly in speech recognition

and forensic audio analysis. MFCCs are derived from the Mel-frequency spectrogram, which is a spectrogram that uses a frequency scale that is more aligned with human perception of sound. In forensic audio analysis, MFCCs can be used as a feature representation to compare and analyze different audio recordings. By computing the MFCCs for different segments of an audio recording, forensic audio analysts can identify characteristic patterns and features that may be relevant to a legal case [75]. The Mel Frequency Cepstral Coefficients (MFCCs) are computed using the following steps:

1. Divide the signal into overlapping segments of length N.
2. Compute the short-time Fourier transform (STFT) of each segment.
3. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows or alternatively, cosine overlapping windows.
4. Take the logs of the powers at each of the mel frequencies.
5. Take the Discrete Cosine Transform (DCT) of the log mel filterbank energies to give 26 cepstral coefficients.

The following mathematical formula describes the MFCCs:

$$MFCC_i = \sum_{j=1}^{M} c_j \cos\left(\frac{\pi i}{M}(j - 0.5)\right) \tag{5.6}$$

where $MFCC_i$ is the i-th MFCC coefficient, c_j is the j-th mel filterbank energy, M is the number of MFCC coefficients

- **Compare different DCT bases**: In audio signal processing, the discrete cosine transform (DCT) is a widely used method for transforming time-domain audio signals into a frequency-domain representation. There are different types of DCTs that use different basis functions, or sets of orthogonal functions, to represent the signal in the frequency domain. The choice of DCT basis functions depends on the specific application and the trade-offs between computational efficiency, frequency resolution, and energy compaction. In many cases, the standard DCT-II is a good choice for audio signal processing applications, but other DCT bases may be more appropriate for certain types of signals or processing tasks.

- **Root-Mean-Square (RMS)**: Compute root-mean-square (RMS) value for each frame, either from the audio samples y or from a spectrogram S. Computing the RMS value from audio samples is faster as it doesn't require an STFT calculation. However, using a spectrogram will give a more accurate representation of energy over time because its frames can be windowed, thus prefer using S if it's already available.

- **Spectral Centroid**: Compute the spectral centroid.Each frame of a magnitude spectrogram is normalized and treated as a distribution over frequency bins, from which the mean (centroid) is extracted per frame [126]. The spectral centroid is a measure used in audio forensics to characterize an audio signal, often indicating the perceived "brightness" of a sound. It aids in differentiating sounds and identifying unique voices, thus assisting in speaker identification.

The spectral centroid can also reveal potential audio tampering, as inconsistencies might suggest alterations. Additionally, it serves as a tool for audio quality assessment, with higher values indicating clearer recordings. More precisely, the centroid at frame t is defined as

$$\text{centroid}[t] = \frac{\sum_k S[k,t] \cdot \text{freq}[k]}{\sum_j S[j,t]} \tag{5.7}$$

- **Spectral Bandwidth**: Compute $p'th - order$ spectral bandwidth [126]. In the realm of audio enhancement, knowledge of the spectral bandwidth can aid in developing strategies to filter out unwanted components from a recording. If noise or other undesirable signals are limited to a specific bandwidth, a band-stop filter could be effectively employed to remove it. It also can be used as a fingerprint, the spectral bandwidth of an individual's voice can be unique. This characteristic can be analyzed to potentially match a voice to a specific person, which can prove extremely useful in forensic investigations. The spectral bandwidth 1 at frame t is computed by:

$$(\sum_k S[k,t] \cdot (freq[k,t] - centroid[t])^p)^{\frac{1}{p}} \tag{5.8}$$

- **Spectral Contrast**: Compute spectral contrast. Each frame of a spectrogram S is divided into sub-bands. For each sub-band, the energy contrast is estimated by comparing the mean energy in the top quantile (peak energy) to that of the bottom quantile (valley energy)[117]. High contrast values generally correspond to clear, narrow-band signals, while low contrast values correspond to broad-band noise.

3 Deep Learning Audio Inference Module

Deep Audio Analyzer implements several audio analysis tasks using deep learning methods. The neural networks present in Deep Audio Analyzer are state of the art for the different tasks, and their implementation of is currently supported by the SpeechBrain [215] framework that implements interfaces through which it is possible to download and execute neural network models through the HuggingFace [54] aggregator. Table 5.1 summarizes the various neural network models for the different tasks and related datasets on which they were trained and the obtained performance.

4 Pipeline Creation and Saving

Through Deep Audio Analyzer, it is possible to perform analysis of an audio file dynamically by creating an audio analysis pipeline. Fig 5.4 shows the flowchart expressing the working principle of audio analysis with Deep Audio Analyzer.

The following list represents the process of analysis and pipeline creation:

1. First, the input audio file is selected (Fig 5.5).

2. Once the file is selected, the task to be performed and consequently, the neural network model is chosen from those available for that task, as show in fig. 5.6.

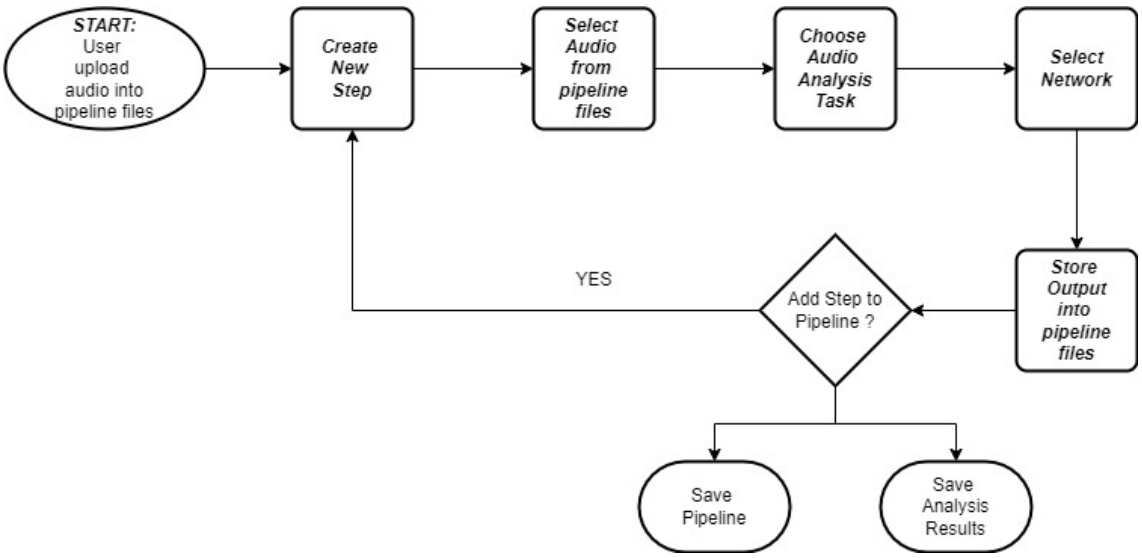


Figure 5.4: Pipeline Creation and Dynamic Audio Analysis Flowchart

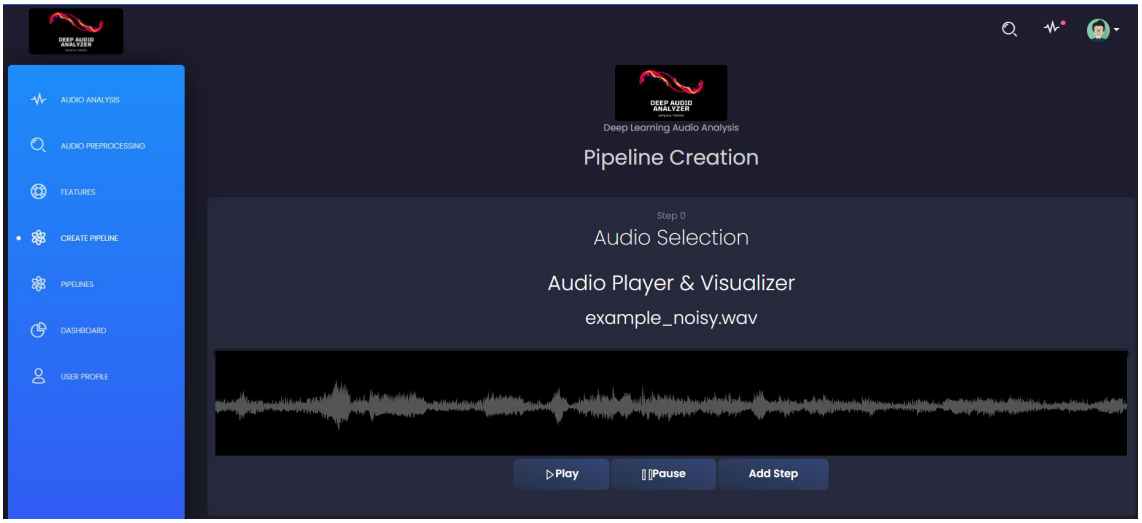


Figure 5.5: Architecture of Audio Feature Visualization Module.

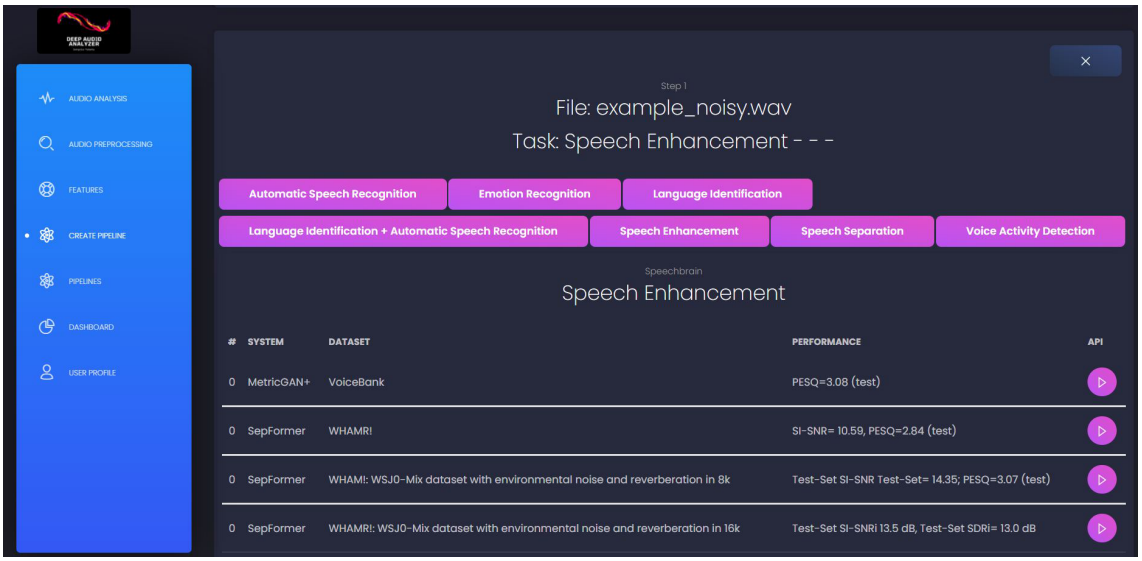


Figure 5.6: Select Task and Model to create a step of audio analysis pipeline.

3. Once the step is defined, it is possible to execute it by means of a POST request sent to the server, which will execute the neural network in inference and return the result of the task performed to the client (fig 5.7).

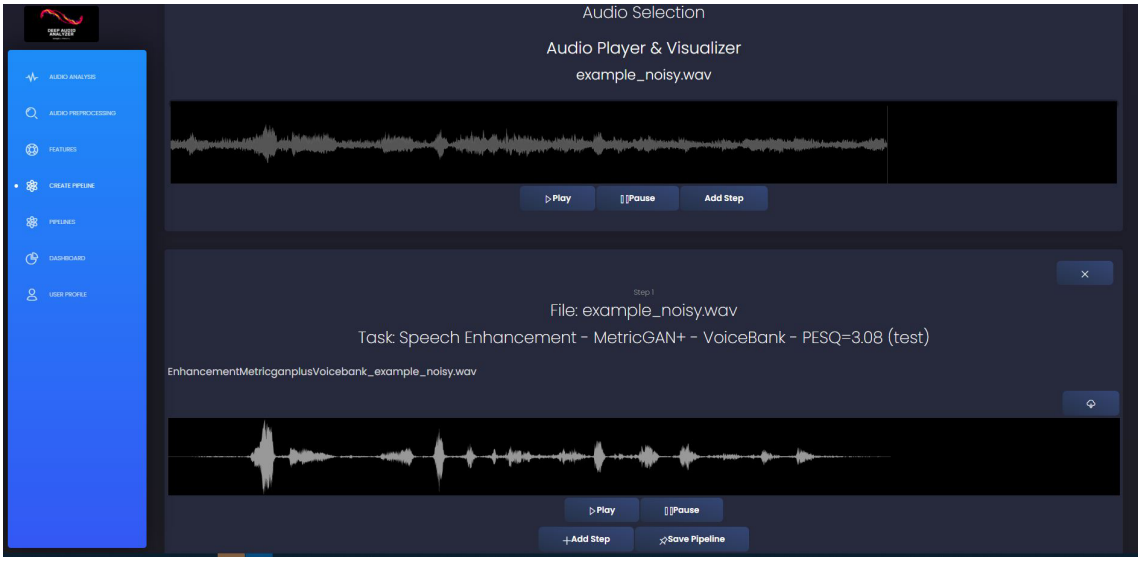


Figure 5.7: Pipeline Creation and Dynamic Audio Analysis Flowchart

4. Then it is possible to add a new step to the pipeline by choosing on which file to perform the analysis or save the pipeline from executing it later on different files (Fig 5.8).

5 Pipeline Execution and Download Report

Audio analysis pipelines that have been previously saved by the user are available in the ”pipelines” section. It is possible to run a previously created pipeline, on one or more (previously recorded) audio files, or to perform a recording of an audio file using the GUI application. Once the type of input to be analyzed has been selected, it is possible to choose the type of pipeline and view its details. Then Deep Audio Analyzer will display via Frontend the results of the inferences (performed on the Backend side). After the analysis process, it is possible to download the reports containing the pipeline executed on each file and its results for each step that is part of it. Figure 5.9 describes the flowchart for pipeline execution and reporting.

6 Experiments and Results

In this section, some examples of generated pipelines and the related tests performed by tests different neural networks available for the different tasks and the obtained results using the Deep Audio Analyzer, are described.

6.1 New pipeline generation

In this section, we present two examples of pipeline creation that can be used for investigative purposes in interception contexts. The first example concerns the transcription of speech from multiple people speaking different languages, while the second example concerns the transcription of speech in noisy environments using speech enhancement models.

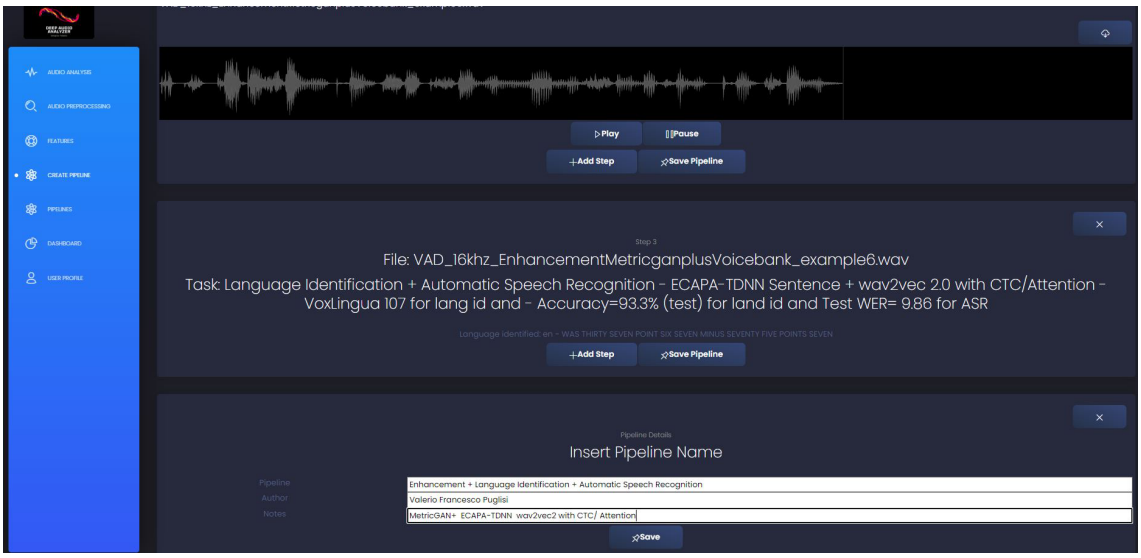


Figure 5.8: Save Pipeline created during the analysis.

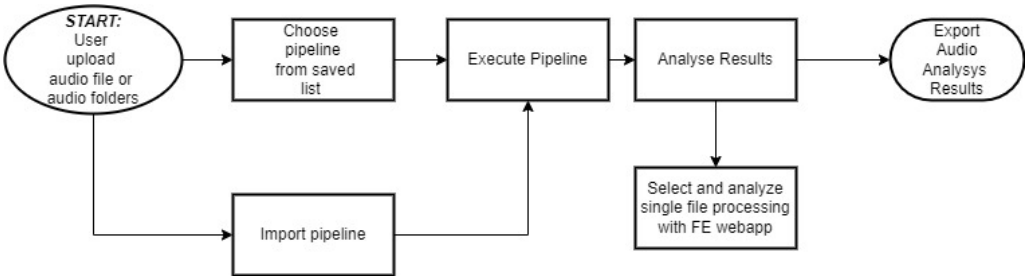


Figure 5.9: Pipeline Execution Audio Analysis Flowchart

Multi-speaker Multi-Language ASR : Speech Separation + Language ID + ASR

This pipeline dedicated to transcribing speech in different languages from a maximum of three speakers is composed of the following steps(Fig. 5.10):

1. Addition of the file of interest, selection of the audio separation model, and execution.

2. This step provides three output files; thus, it is necessary to create three new voice activity detection (VAD) steps one per output file of step one.

3. The three steps will each output a file where silence has been removed. It will then be necessary to introduce three additional steps with our implemented language identification and automatic speech recognition module, taking as input the audio processed with VAD in the previous steps.

Automatic Speech Recognition in noisy environment

In forensic investigations, it is often necessary to transcribe highly noisy audio. This context is often overlooked in academic settings, as the focus is on evaluating transcriptions in clean or low-noise/echo environments. Therefore, creating a pipeline that involves the use of enhancement models and voice activity detection improves the results of automatic speech recognition. The creation of this pipeline consists of the following steps (Fig. 5.11):

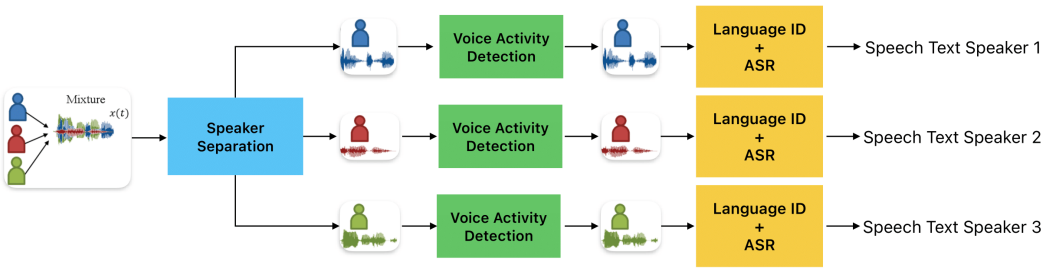


Figure 5.10: Pipeline Multi-speaker Multi-Language ASR : Speech Separation + Language ID + ASR

1. Loading the audio file and selecting the desired model for the speech enhancement task.
2. Adding a new step that takes as input the improved signal produced by step 1 and select the desired Voice Activity Detection model.
3. Adding the language identification and automatic speech recognition task, implemented by us.

Figure 5.11: Pipeline Automatic Speech Recognition in noisy environment

6.2 Experiments

Deep Audio Analyzer is an application designed as a support tool for audio analysis in forensic and also academic fields. For these reasons, several experiments have been implemented including validating models related to different tasks on different datasets through the implementation of appropriate evaluation metrics using the library [69].

Validate performance of a pre-trained neural network on different task The first test case consists of evaluating by means of the metrics set out in the introduction chapter, the behaviour of the various networks with datasets that are different from the training datasets, but which have been realised for the same task to see how are robust the networks, varying the datasets for the same type of task. For example, to evaluate the performance of the Automatic Speech Recognition networks in Deep Audio Analyzer. The current Automatic Speech Recognition networks are trained mainly on Librispeech, Voxpopuli and Common Voice and it is possible to see the performance on different datasets in the table 5.2 by the implementation of Character Error Rate (CER) and Word Error Rate (WER) [284, 175]. We also implemented evaluation for speech separation models in table 5.3 by employing five different metrics.

User validation of the performance of the deep neural networks available for a given task Suppose we wanted to test the quality of the neural networks available in the automatic speech recognition application for a specific language, using files not belonging to datasets. It is possible to do this by the pipeline creation section. The user creates a pipeline and add as many steps as necessary to compare the neural networks available for that language and save the pipeline. Then the users can run the various comparison pipelines (previously created) to test the behaviour of the various networks for tasks in examples that are not included in the training datasets. In this use case, it is not possible to perform a validation according to the metrics related to the task being analysed, because the relevant ground-thoughts are missing. For this reason, it was decided to predefine a perceptual quality index ranging from 1 to 10 for the tasks on the platform.

Subsequently, a pipeline was created for each task in order to compare all the available networks in a single process and then manually evaluate the performance of the individual network from 1 to 10. In this way, it is easy to sample perceptual opinions from experts in the field in order to assess robustness not only in the various existing datasets for the generic task but also with audio files recorded in real 'into the wild' situations.

Automatic Speech Recognition in noisy environment

For this experiment, we decided to test the quality of Automatic Speech Recognition (ASR) models trained on two different clean datasets, without noise. We then compared them to the pipeline described in Figure 5.11, using the same datasets but augmented with noise at various dB levels that simulate real-world conditions present in Forensic cases. Therefore, the ASR models in the pipeline were the same ones used in the noise-free conditions. Specifically, we evaluated two models for this experiment:

1. Wav2Vec 2.0 with CTC trained on LibriSpeech[192], whose metrics are at a Word Error Rate (WER) of 1.90%. This ASR system consists of two distinct yet interconnected blocks: This ASR system is composed of 2 different but linked blocks:
   (a) Tokenizer (unigram) that transforms words into characters and trained with the train transcriptions (EN).
   (b) Acoustic model (wav2vec2.0 + CTC). A pretrained wav2vec 2.0 model (wav2vec2-large-960h-lv60-self) is combined with two DNN layers and finetuned on LibriSpeech. The obtained final acoustic representation is given to the CTC.
2. Wav2Vec 2.0 with CTC trained on CommonVoice [11] English (No LM), which achieves a Word Error Rate (WER) of 15.69% on the CommonVoice [11] Test set without noise. This ASR system is composed of 2 different but linked blocks:
   (a) Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions (train.tsv) of CommonVoice [11] (EN).
   (b) Acoustic model (wav2vec2.0 + CTC). A pretrained wav2vec 2.0 model (wav2vec2-lv60-large) is combined with two DNN layers and finetuned on CommonVoice [11] En. The obtained final acoustic representation is given to the CTC decoder.

Next, we took the sample noise, which can be downloaded using the download_asset function from the torchaudio.utils library at the path [1], and resampled the file to a frequency of 16kHz, which is the sampling frequency used as input for the above-mentioned ASR models. After that, we repeated the background noise using the repeat function from torchaudio for the length of each file within the datasets (resampling the CommonVoice English dataset files to 16kHz since they had a higher sampling frequency). Finally, using the 'add_noise' function from torchaudio.functional, we added the noise at six different signal-to-noise ratio (SNR) levels: -6dB, -4dB, -2dB, 0dB, 2dB, 4dB and 6dB. This allowed us to observe how the models trained on the respective clean datasets perform in the presence of noise. Once we obtained the results of the metrics applied to these tests, we proceeded to apply the pipeline represented in Figure 5.11 to the noise-augmented datasets. This pipeline consists of the Enhancement model MetricGAN+ [81] and the aforementioned ASR models, in order to compare the CER and WER [284, 175] metrics for the two described cases.

Model Evaluation on different Datasets

Tables 5.2, 5.3 show the Evaluation module applied Automatic Speech recognition task and Speech Separation task with pre-trained models on some datasets. However, evaluations conducted on different datasets show that even though a network may show good performance on the training dataset, it may not perform well on other data from different contexts. With Deep Audio Analyzer is possible to upload customized trained models in order to achieve better performance on private datasets.

6.3 Results

Validate performance of a pre-trained neural network on different tasks

Tables 5.2, 5.3 show the Evaluation module applied Automatic Speech recognition task and Speech Separation task with pre-trained models on some datasets. However, evaluations conducted on different datasets show that even though a network may show good performance on the training dataset, it may not perform well on other data from different contexts. With Deep Audio Analyzer is possible to upload customized trained models in order to achieve better performance on private datasets.

Automatic Speech Recognition in noisy environment

In this section, we present the results of the experiments. The results of the first model, Wav2Vec 2.0 + CTC, trained on Librispeech and tested on Librispeech augmented with different signal-to-noise ratio (SNR) levels are in Fig. 5.12, along with the metric values calculated on the tests of the pipeline that includes an upstream enhancement model, as shown in Fig. 5.13. In Fig. 5.13, you can see the trend of the CER and WER errors as the signal-to-noise ratio (SNR) varies in the pipeline shown in Fig. 5.11 using the same ASR model trained on Librispeech. The results are also presented in tabular form in the following tables: Table 5.4 and Table 5.5 in order to be able to compare results. As can be observed from both the graphs and the numbers reported in the tables, the application of the pipeline referenced in Fig. 5.11 results in a significant reduction in CER and WER errors. This implies that the use of enhancement models in noisy environments eliminates the problem of unintelligibility,

[1] tutorial-assets/Lab41-SRI-VOiCES-rm1-babb-mc01-stu-clo-8000hz.wav

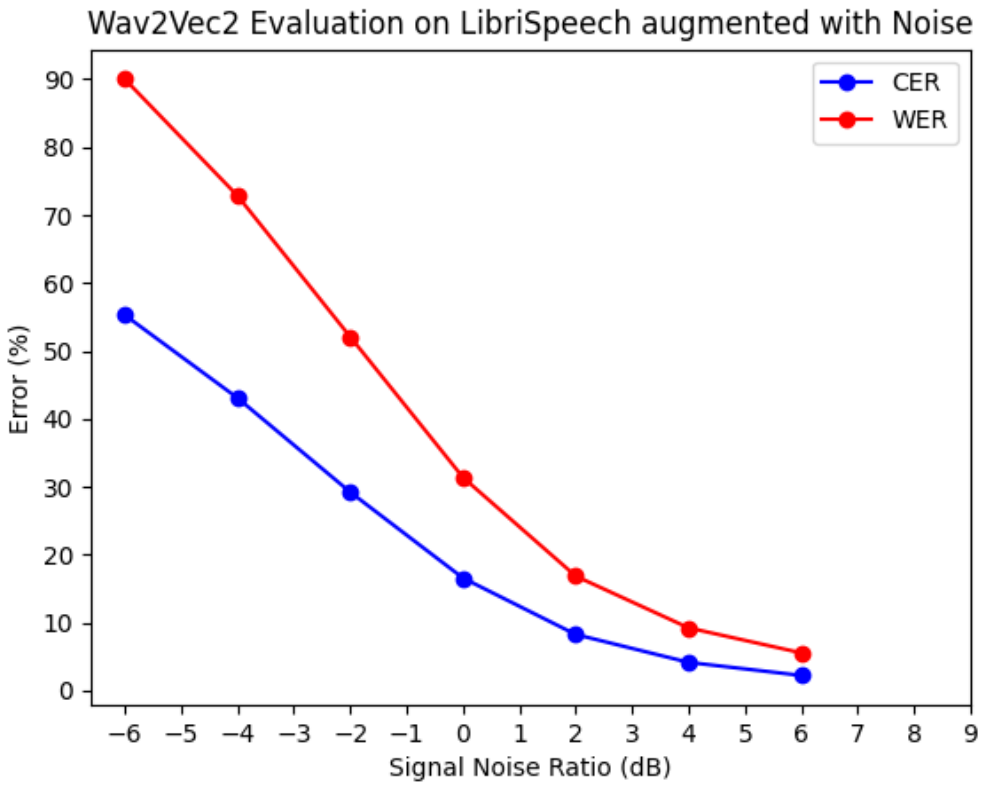


Figure 5.12: Automatic Speech Recognition with Wav2Vec 2.0 + CTC model trained on Librispeech and tested on Augmented Noisy Librispeech

particularly for high signal-to-noise ratios (e.g., -6dB, -4dB, -2dB), dramatically improving transcription accuracy in real-world contexts. It is worth noting that transcription of noise-free audio files is not a common occurrence in real life. In particular, the application of deep learning-based enhancement methods offers highly effective dynamic noise reduction. This provides a valuable tool for forensic analysis, enhancing both enhancement capabilities (which in forensic literature primarily refer to static noise reduction techniques) and interpretation. Utilizing the various models available in different languages on DeepAudioAnalyzer, it provides a useful support tool for examiners who do not have technical programming knowledge.

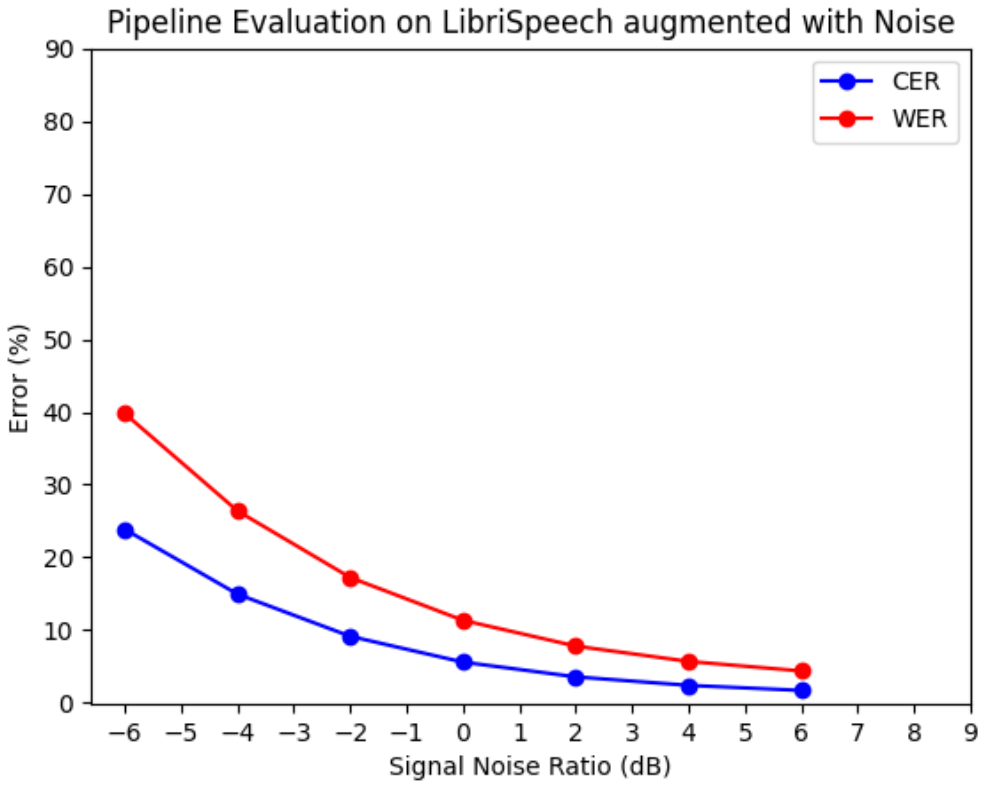


Figure 5.13: Audio Enhancement with MetricGAN+ and Automatic Speech Recognition with Wav2Vec 2.0 + CTC model trained on Librispeech and tested on Augmented Noisy Librispeech

Table 5.1: Deep Learning Models: ASR: Automatic Speech Recognition, ER: Emotion Recognition, LI: Language Identification, SE: Speech Enhancement, SS: Speech Separation, SV: Speaker Verification, VAD: Voice Activity Detection

Task	**System**	**Dataset**	**Performance**
ASR	wav2vec2[14]	*LibriSpeech*[192]	WER=1.90%
ASR	CNN + Transformer	*LibriSpeech*[192]	WER=2.46%
ASR	CRDNN + distillation	*TIMIT*[83]	PER=13.1%
ASR	CRDNN + RNN+ LM	*Librispeech*[83]	WER=3.09% (test-clean)
ASR	Conformer + Transf. LM	*Librispeech*[192]	WER=3.09% (test-clean)
ASR	CRDNN + Transf. LM	*Librispeech*[192]	WER=8.51% (test-clean)
ASR	wav2vec2 + CTC/Att.[14, 63]	*TIMIT*[83]	PER=8.04%
ASR	wav2vec2 + CTC	*CV (English)*[11]	WER=15.6%
ASR	wav2vec2 + CTC	CV (German)[11]	WER=9.54%
ASR	wav2vec2 + CTC	CV (French)[11]	WER=9.96%
ASR	wav2vec2 + seq2seq	CV (Italian)[11]	WER=9.86%
ASR	wav2vec2 + seq2seq	AISHELL[34]	5.58%
ER	wav2vec [236]	IEMOCAP[35]	Acc.=79.8%
ER	wav2vec [236]	CommonLang.[246]	Acc.=84.9%
LI	ECAPA-TDNN[68]	CommonLang.[246]	Acc.=84.9%
SE	MetricGAN+ [81]	VoiceBank	PESQ=3.08 (test)
SE	SepFormer [249]	WHAMR! [160]	SI-SNR= 10.59, PESQ=2.84 (test)
SE	SepFormer [249]	WHAM! (8k) [283]	SI-SNR= 14.35, PESQ=3.07 (test)
SE	SepFormer [249]	WHAM! (16k) [283]	SI-SNRi 13.5 dB, SDRi= 13.0 dB
SS	SepFormer[249]	WSJ2MIX[106]	SDRi=22.6 dB
SS	SepFormer[249]	WSJ3MIX[106]	SDRi=20.0 dB
SS	SepFormer[249]	WHAM![283]	SDRi= 16.4 dB
SS	SepFormer[249]	WHAMR![160]	SDRi= 14.0 dB
SS	SepFormer[249]	Libri2Mix[57]	SDRi= 20.6 dB
SS	SepFormer[249]	Libri3Mix[57]	SDRi= 18.7 dB
SV	ECAPA-TDNN [68]	VoxCeleb2 [48]	EER=0.69%
VAD	CRDNN [216]	LibriParty [214]	F-score=0.94%

Deep Learning Audio Analysis Features. .

Table 5.2: Evaluation of Automatic Speech Recognition Models on different Test Datasets

System	***Training dataset & Evaluation Metrics***	**Test dataset & Evaluation Metrics**
wa2vec2 + CTC	Voxpopuli DE WER=18.91%	CommonVoice [11] 10 DE CER=20.13% WER=53.15%
CRDNN with Transformer LM	Librispeech EN WER=8.51%	CommonVoice [11] 10 EN CER=25.08% WER=47.37%
CRDNN + RNN+ LM	Librispeech EN CER= – WER= –	CommonVoice [11] 10 EN CER=28.88% WER=50.05%
wa2vec2 + CTC	Librispeech EN CER= – WER=15.69%	CommonVoice [11] 10 EN CER= 19.78% WER=32.06%
wa2vec2 + CTC	Voxpopuli EN CER=– WER=–	CommonVoice [11] 10 EN CER=32.03% WER= 64.52%
wa2vec2 + CTC	Voxpopuli ES CER=– WER=15.69%	CommonVoice [11] 10 ES CER=17.2822% WER=46.31%
wa2vec2 + CTC	Voxpopuli FR Test CER=3.19 WER=9.96%	CommonVoice [11] 10 FR CER=25.97% WER=58.70%
CRDNN with CTC/Attention	CommonVoice [11] 9 FR CER=6.54%, WER=17.70%	CommonVoice [11] 10 FR CER= 9.55% WER=30.82%
CRDNN with CTC/Attention	CommonVoice [11] 9 IT CER=5.40% WER=16.61%	CommonVoice [11] 10 IT CER=7.78% WER=27.69%
wa2vec2	CommonVoice [11] 9 IT Test CER=– WER=9.86%	CommonVoice [11] 10 IT CER=7.30% WER=21.66%
wa2vec2	VoxPopuli 9 IT Test CER=– WER= 45.2%	CommonVoice [11] 10 IT CER=16.00% WER=52.57%

Table 5.3: Evaluation of Speech Separation Models on different Test Datasets

System	***Training dataset & Evaluation Metrics***	**Test dataset & Evaluation Metrics**
Sepformer	WSJ2MIX SDRi=22.6 dB (test)	Libri2Mix 16K Min SNR= -9.3865, SDR = -0.2170, SI-SNR = -2.5669, SI-SDR= -2.5678, PESQ= 2.0454, STOI= 0.5051
Sepformer	WSJ2MIX SDRi=22.6 dB (test)	Libri2Mix 16K Max SNR = -9.1042, SDR = -0.0988, SI-SNR = -2.0402, SI-SDR = -2.0445, PESQ = 2.0879, STOI = 0.5297
Sepformer	WSJ3MIX SDRi=20.0 dB (test)	Libri3Mix 16K Min SNR = -8.2628 SDR = -5.3410, SI-SNR = -4.8382, SI-SDR= -4.8382, PESQ = 1.5473, STOI = 0.3136
Sepformer	WSJ3MIX SDRi=20.0 dB (test)	Libri3Mix 16K Max SNR = -8.3537 SDR = -5.3429, SI-SNR = -7.8382, SI-SDR= -7.8382, PESQ = 1.6473, STOI = 0.3903

Table 5.4: Evaluation of Wav2Vec 2.0 + CTC Automatic Speech Recognition Model on Noisy Librispeech

SNR (dB)	***CER(%)***	***WER(%)***
-6dB	55.30	90.02
-4dB	43.07	72.85
-2dB	29.21	52.06
0dB	16.56	31.26
2dB	8.24	16.83
4dB	4.14	9.22
6dB	2.21	5.53

Table 5.5: Evaluation of Wav2Vec 2.0 + CTC Automatic Speech Recognition Model on Noisy Librispeech

SNR (dB)	***CER(%)***	***WER (%)***
-6dB	23.87	39.79
-4dB	14.96	26.40
-2dB	9.15	17.24
0dB	5.61	11.35
2dB	3.58	7.81
4dB	2.21	5.70
6dB	1.72	4.39

Chapter 6

Sound Source Localization

Sound Source Localization is a complex challenge in the field of audio signal processing. Here are some of the issues and difficulties associated with this area:

- Complex Acoustic Environment: In complex acoustic environments, such as rooms with strong reverberation or open spaces, accurately determining the exact direction of the sound source can be challenging due to numerous reflections and interferences.
- Limited Microphones: With a limited number of microphones available, especially with a single monophonic recording, obtaining accurate information about source localization can be challenging.
- Environmental Variability: Environmental characteristics, such as furniture arrangement or the presence of obstacles, can impact sound propagation and make precise source localization more difficult.
- Background Noise: Background noise, such as ambient noise or equipment noise, can mask the sound source signal and make its detection more challenging.
- Ambiguity: Sometimes, different combinations of source directions and distances can result in similar signal patterns, leading to ambiguous localization solutions.
- Dynamic Changes: If the sound source or microphones move during recording, the localization problem becomes more complex due to changes in the sound captured by the microphones.
- Microphone Directional Characteristics: The directional sensitivity of microphones can influence the perception of source direction. If microphones are not omnidirectional, localization might require accurate calibration.
- Measurement Error: Even with sophisticated algorithms, there is always a certain degree of error in localization due to signal processing imperfections and limitations of the algorithms themselves.

Addressing these challenges requires the use of advanced signal processing algorithms, high-quality microphones, analysis of the acoustic environment, and, in some cases, the integration of multiple microphones to achieve more accurate results.

This study on Sound Source Localization aims to recognize the best way to localize single or multiple sources inside an office room. The choice of a specific environment will reduce the variability of this problem in order to understand what is the best

practice in this context. Different experiments have been done by changing the microphone configuration and the different algorithms applied in order to understand what is the most accurate algorithm to apply in this scenario.

1 PyRoomAcoustics: Audio Virtual Environment Configurations for Simulations

Pyroomacoustics is a software package designed to facilitate the swift development and experimentation of audio array processing algorithms. The package can be categorized into three key components:

1. An easy-to-use, object-oriented Python interface that allows for the rapid creation of various simulation scenarios involving multiple sound sources and microphones within both 2D and 3D environments.

2. A high-speed C++ implementation of the image source model and ray tracing, tailored for general polyhedral rooms. This component efficiently generates room impulse responses and simulates sound propagation between sources and receivers.

3. Ready-to-use implementations of widely-used algorithms encompassing Short-Time Fourier Transform (STFT), beamforming, direction finding, adaptive filtering, source separation, and single channel denoising.

These components combine to form a package that has the potential to accelerate algorithm development cycles by significantly reducing the implementation overhead during performance evaluation. To see the various facets of this package in action, please consult the provided notebook.

At the heart of the software package lies a generator for room impulse responses (RIR), built upon the image source model framework, capable of accommodating a diverse range of room configurations:

1. Convex and non-convex room shapes.

2. Both 2D and 3D room dimensions.

The core components responsible for the image source model and ray tracing functionalities are implemented in C++, a choice that enhances computational efficiency.

The package's underlying philosophy revolves around encapsulating all essential aspects of an experiment through an object-oriented programming paradigm. Each constituent element is represented using a distinct class, allowing for the assembly of experiments akin to real-world scenarios.

Consider a scenario where a delay-and-sum beamformer is to be simulated using a linear array composed of four microphones situated within a room shaped like a shoe box, hosting a solitary sound source. Initially, a room object is instantiated,

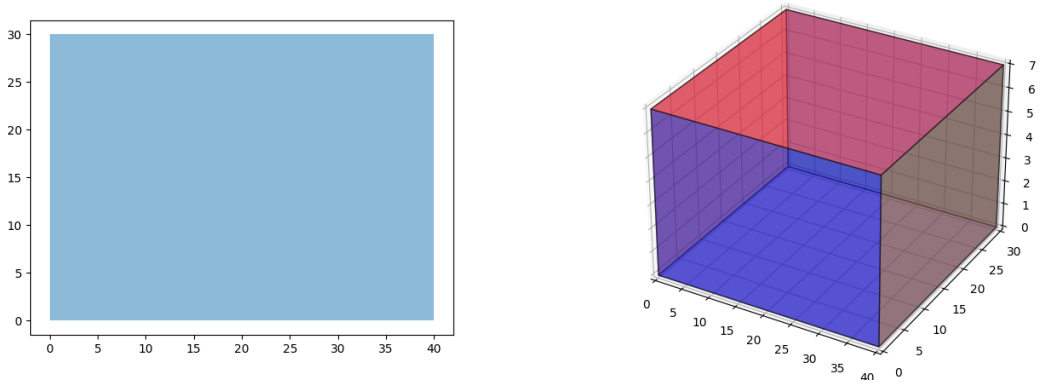
to which a microphone array object and a sound source object are added. Subsequently, the room object boasts methods to compute the RIR connecting the source and the receiver. The beamformer object extends the microphone array class and encompasses diverse methods to compute parameters like delay-and-sum weights. Refer to the provided code example to gain a clearer grasp of the implementation.

Furthermore, the Room class facilitates sound sample processing emanating from sources, effectively emulating the propagation of sound between the sources and microphones. As the signals reach the microphones constituting the beamformer, a Short-Time Fourier Transform (STFT) engine becomes instrumental in swiftly processing these signals through the beamformer and evaluating the resultant output.

The research of Sound Source Localization field starts with the exploration of different microphone configurations. As a first approach, the problem was studied by attempting to localize an audio source using a single microphone with a single source at a time. This approach was adopted in order to evaluate which audio features were the most discriminative in solving a regression or classification problem. The ability to solve a regression/classification problem, both in Machine Learning and Deep Learning, depends on the data distribution that will be fed into the models under consideration.

2 Single Microphone Sound Source Localization

For the creation of the dataset, the implementation of an empty museum room with dimensions of 30x40 meters and a height is planned. Fig. 6.1 is a representation of the room virtualized for dataset creation. Dataset was created by taking speech recordings from LibriSpeech and simulating audio inside the virtual room changing the virtual location of the speaker.



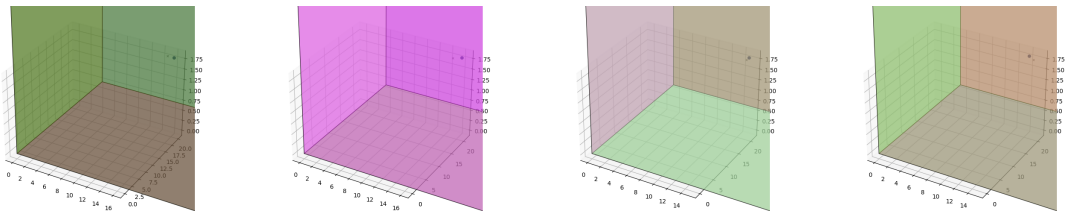
(a) 2D Room Visualisation

(b) 3D Room Visualisation

Figure 6.1: Empty Room Museum Size

The simulated measurements take into account 8 angles on the same plane: 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°.

As for the distances, a range from 1 to 10 meters is considered with a step of 1 meter. Therefore, for each audio sample of a Speaker ID, 80 measurements are taken = 10 distances x 8 angles (Fig. 6.2).

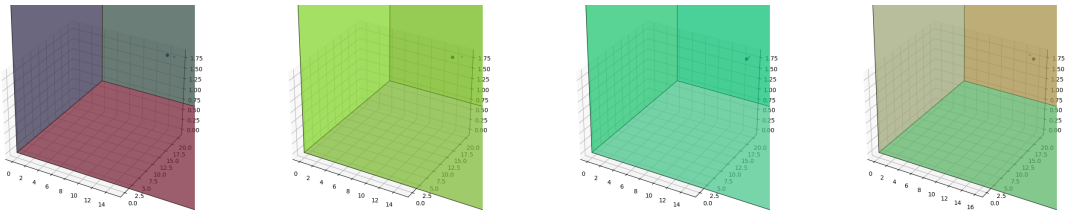


(a) 3D room planimetry with microphone at the center of room and source at 1m and 0°

(b) 3D room planimetry with microphone at the center of room and source at 1m and 45°

(c) 3D room planimetry with microphone at the center of room and source at 1m and 90°

(d) 3D room planimetry with microphone at the center of room and source at 1m and 135°



(e) 3D room planimetry with microphone at the center of room and source at 1m and 180°

(f) 3D room planimetry with microphone at the center of room and source at 1m and 225°

(g) 3D room planimetry with microphone at the centre of room and source at 1m and 270°

(h) 3D room planimetry with microphone at the centre of room and source at 1m and 315°

Figure 6.2: The simulated measurements take into account 8 angles on the same plane: 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°. As for the distances, a range from 1 to 10 meters is considered with a step of 1 meter. Therefore, for each audio sample of a Speaker ID, 80 measurements are taken = 10 distances × 8 angles.

The following example graphs show the variation of angles between the microphone and the source at a distance of 1 meter, with a fixed height of 1.75 meters to simulate the average human height.

The Data Visualization experiments are reported in the following list:

- **T-SNE**
  - Single Speaker Multiple Utterance Chunks
  - Multiple Speaker Single Utterance Chunk
  - Multiple Speaker Multiple Utterance Chunks
- **PCA**
  - Single Speaker Multiple Utterance Chunks
  - Multiple Speaker Single Utterance Chunk
  - Multiple Speaker Multiple Utterance Chunks

2.1 Data Visualization

After Dataset creation it was necessary apply algorithms like T-SNE and PCA to the different features of these recordings like Raw Waveform, Waveform, Spectrogram, MFCC and Wav2Vec2 Features to understand if these features are able to visualize data into something similar of a function that can be learnt via regression problem or clusters to resolve the problem as a classification problem.

T-SNE: Single Speaker Multiple Chunks

The Data Visualization started with the calculation of T-SNE algorithm to different recordings of the same speaker on different speech chunks.

Fig. 6.5 represent the data visualization of different speech chunks of the same speaker at different distances from the microphone (from 1 to 10 meters) with an azimuth of 0°.

In the next plots, the darker points stand for the minimum distance while the yellow points stand for the maximum distance in order to represent distances in an intuitive way.

Even if the same speaker is talking in the same room the representation of different chunks is quite different from a function or cluster that is learnable from a Deep Learning method. This happens because the clusters needed to learn distance from different chunks should be separated by colours that indicate distance. In other words, the clusters should be separated by distance and not by chunk as in this plot.

The plots

Raw Waveform Analysis First fig. 6.3 calculates T-SNE on Raw Waveform and is not possible to distinguish chunks by distance.

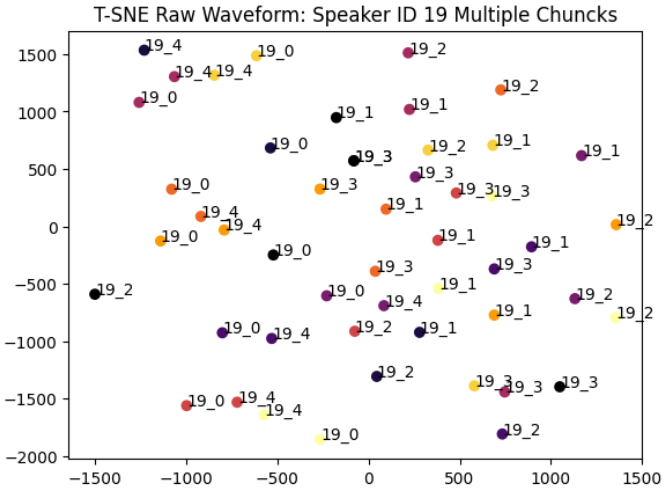


Figure 6.3: Multiple Speech Chunks processed on Raw Waveform of Speaker ID 19 at different distances from microphone

Spectrogram Analysis Fig. 6.4 shows that using a Spectogram as a feature to resolve the localization problem is not a good point.

MFCC Analysis ig. 6.5 shows a data visualization of T-SNE applied on MFCC Features extracted from a Single Utterance Chunk of a Single Speaker.

Wav2Vec2 Features Analysis Fig. 6.6 shows a data visualization of T-SNE applied on Wav2Vec2 Features extracted from Single Utterance Chunk of a Single Speaker.

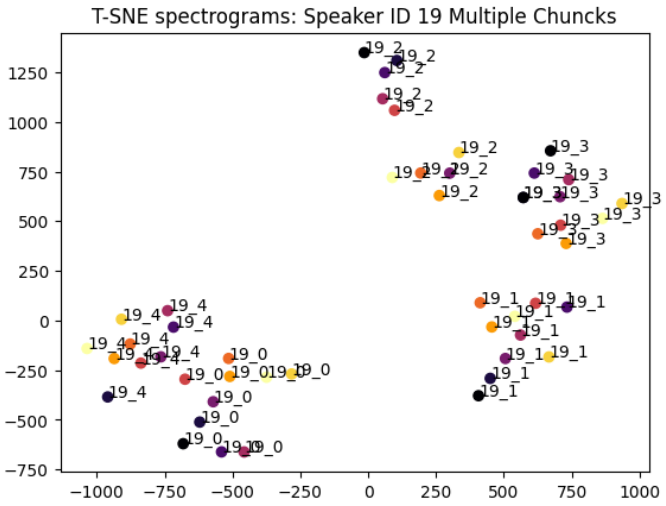


Figure 6.4: Multiple Speech Chunks processed with Spectrogram of Speaker ID 19 at different distances from microphone

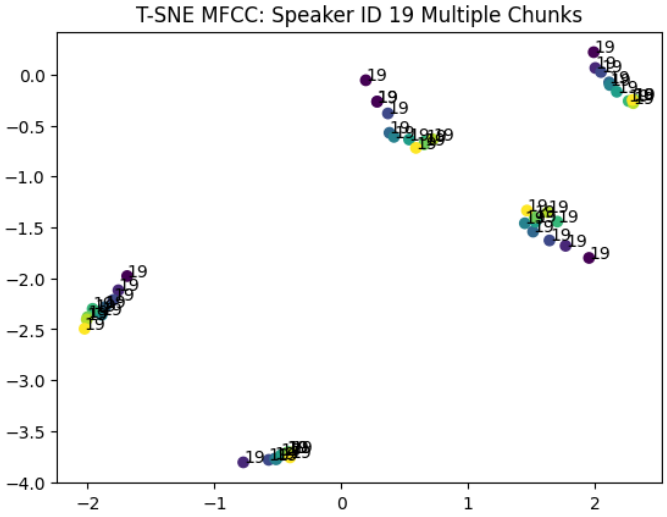


Figure 6.5: Multiple Speech Chunks processed with MFCC of Speaker ID 19 at different distances from microphone

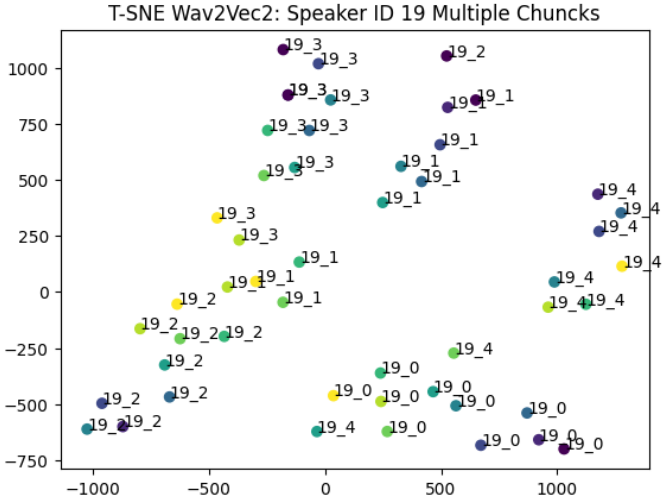


Figure 6.6: Multiple Speech Chunks processed with Wav2Vec2 of Speaker ID 19 at different distances from microphone

T-SNE: Multiple Speaker Single Chunk

The Data Visualization started with the calculation of the T-SNE algorithm for different recordings of multiple speakers on different speech chunks.

Raw Waveform Analysis First fig. 6.7 calculates T-SNE on Raw Waveform and is not possible to distinguish chunks by distance.

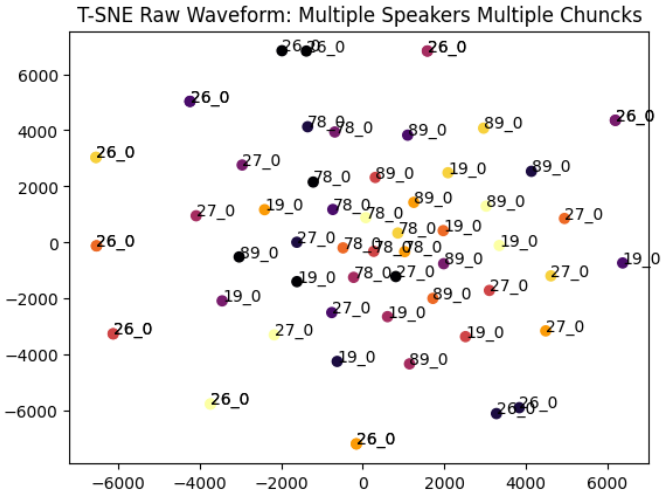


Figure 6.7: Multiple Speech Chunks processed of Multiple Speaker at different distances from microphone

Spectrogram Analysis Second Fig. 6.8 shows T-SNE applied on Spectrogram of Multiple Speaker Utterance.

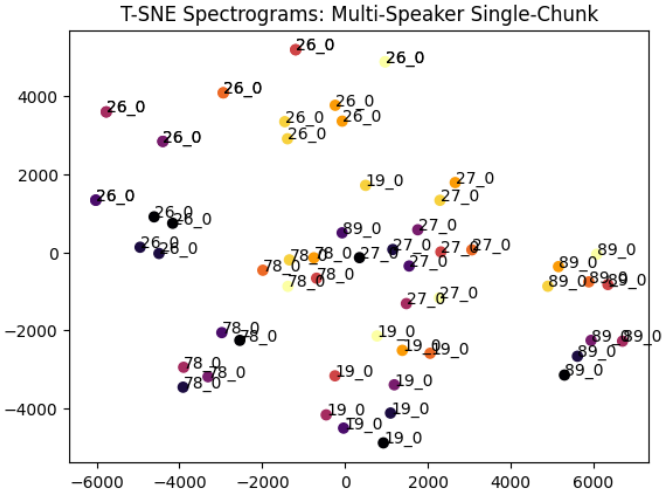


Figure 6.8: Multiple Speech Chunks processed with Spectrogram of Multiple Speaker at different distances from microphone

Mel Frequency Cepstrum Coefficents Analysis Fig.6.9 shows a data visualization of T-SNE applied on MFCC extracted by Multiple Utterance Chunks of Multiple Speakers.

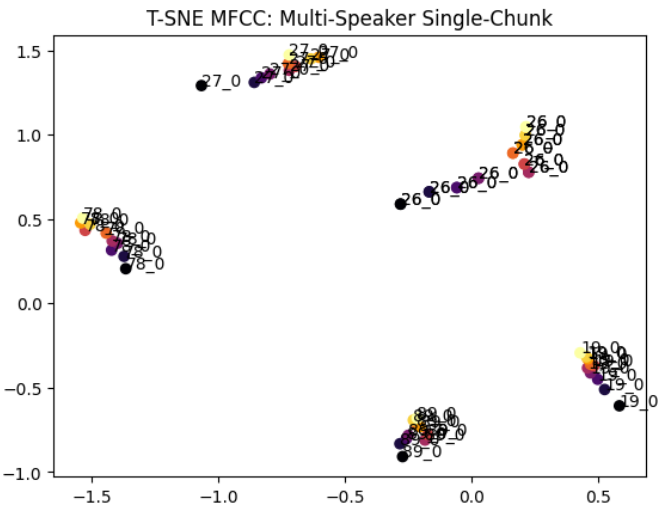


Figure 6.9: Multiple Speech Chunks processed with MFCC of Speaker ID 19 at different distances from microphone

Wav2Vec2 Features Analysis Fig. 6.10 shows a data visualization of T-SNE applied on Wav2Vec2 Features extracted by Single Utterance Chunk of Multiple Speakers.

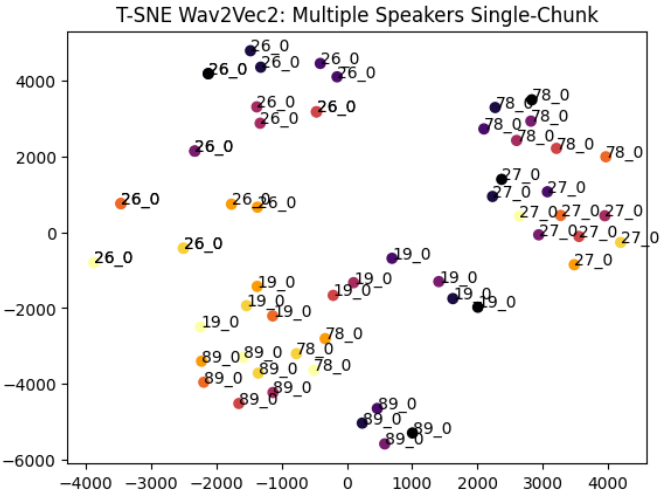


Figure 6.10: Single2 Speech Chunks processed with Wav2Vec2 of Multiple Speaker at different distances from microphone

T-SNE: Multiple Speaker Multiple Chunks

Raw Waveform Analysis This experiment shows Data visualized by T-SNE applied on Raw Waveform of different utterance chunks of different speakers recorded at different distances from the microphone in a range of 1 to 10 meters. As shown in Fig. 6.11, the Raw Waveform does not represent a discriminative feature to represent data in order to create a regression or classification problem.

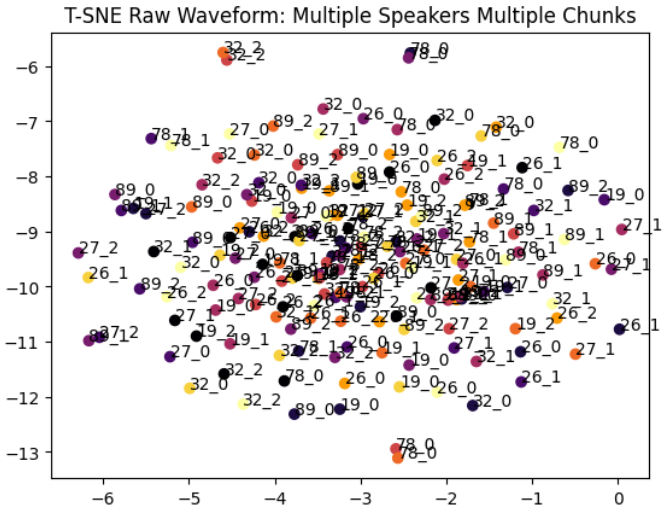


Figure 6.11: Data Visualization with T-SNE applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Spectrogram Analysis Fig. 6.12 shows a data visualization of T-SNE applied on Spectrogram extracted by a Single Utterance Chunk of Multiple Speakers.

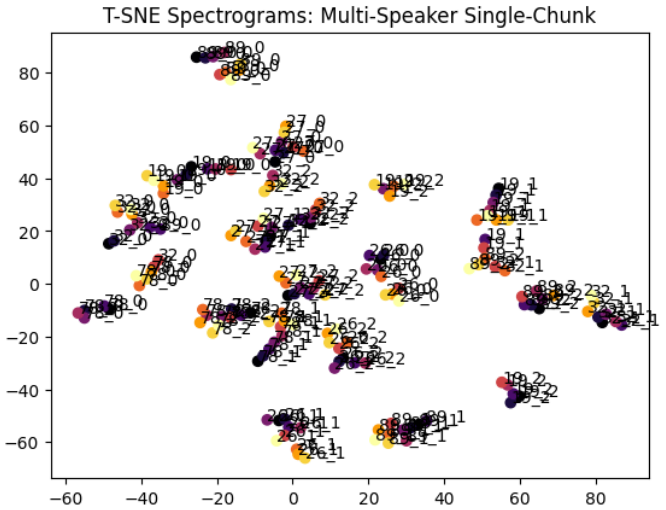


Figure 6.12: Data Visualization with T-SNE applied on Spectrograms of a Single Speech Chunk of Multiple Speakers at different distances from microphone

Mel Frequency Cepstrum Coefficents Analysis Fig. 6.13 shows a data visualization of T-SNE applied on MFCC extracted by a Single Utterance Chunk of Multiple Speakers.

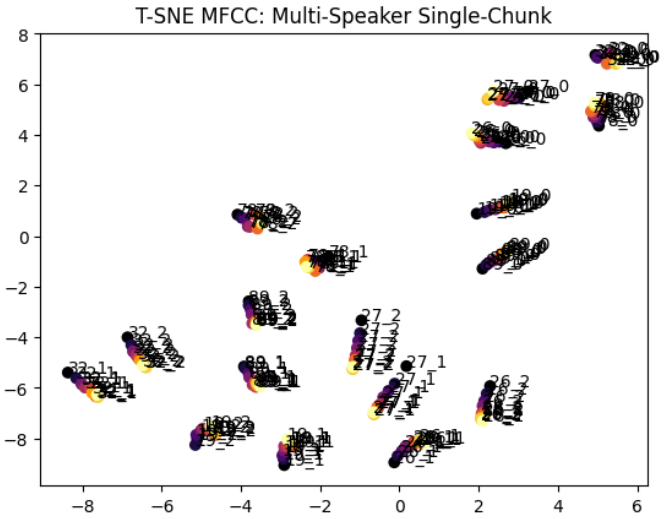


Figure 6.13: Data Visualization with T-SNE applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Wav2Vec2 Features Analysis Fig. 6.14 shows a data visualization of T-SNE applied on Wav2Vec2 Features extracted by Single Utterance Chunk of Multiple Speakers.

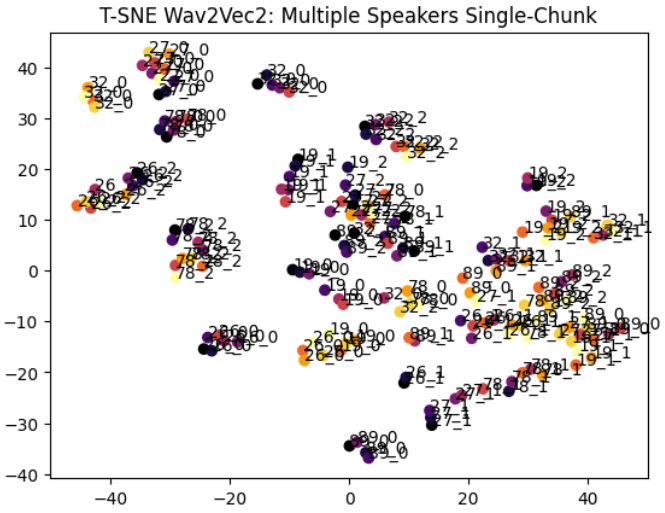


Figure 6.14: Data Visualization with T-SNE applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Principal Component Analysis (PCA): Single Speaker Multiple Chunk

Raw Waveform Analysis This experiment calculates the PCA of Raw Waveforms of different utterance chunks of the same speaker recorded at different distances from the microphone. As shown in Fig. 6.15 using just the waveform for a distance analysis is not discriminative because the outcome is not similar to any learning model.

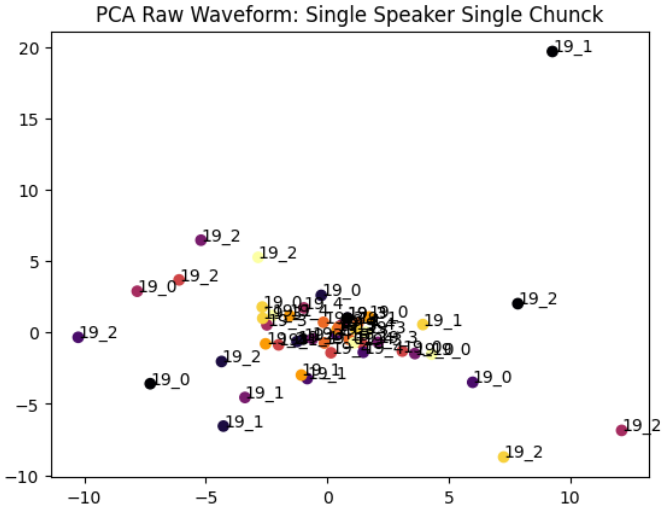


Figure 6.15: Data Visualization with PCA applied on Raw Waveform of Multiple Speech Chunks of a Single Speaker at different distances from microphone

Waveform Spectrogram Analysis This experiment calculates the PCA of Spectrograms calculated on different utterance chunks of the same speaker recorded at different distances from the microphone. The PCA is calculated in 2D because it can be visualized in a 2D axis. As shown in Fig. 6.16 using just the spectrogram for a distance analysis is not discriminative because the outcome is not similar to any learning model.

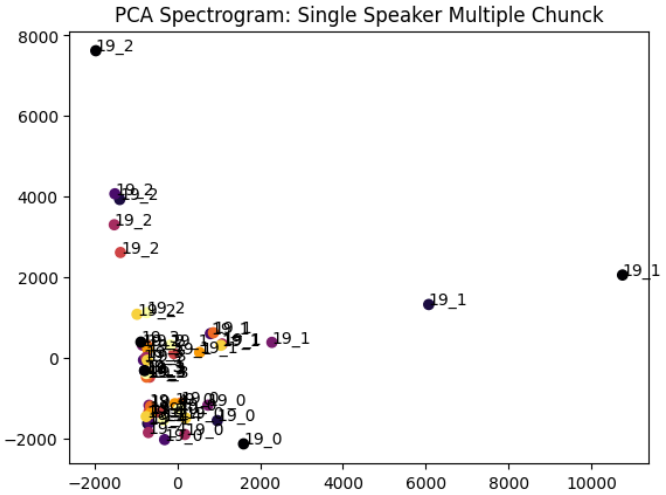


Figure 6.16: Data Visualization with PCA applied on Spectrogram of Multiple Speech Chunks of a Single Speaker at different distances from microphone

Mel Frequency Cepstrum Coefficents Analysis This experiment calculates the PCA of MFCC calculated on different utterance chunks of the same speaker recorded at different distances from the microphone. The PCA is calculated in 2D because it can be visualized in a 2D axis. As shown in Fig. 6.18 using this feature for a distance analysis produces an outcome that is similar to a cluster, but this cluster is not well-formed so is not learnable by any learning model. The reason is the data distribution that is not representative of creating a regression or classification model.

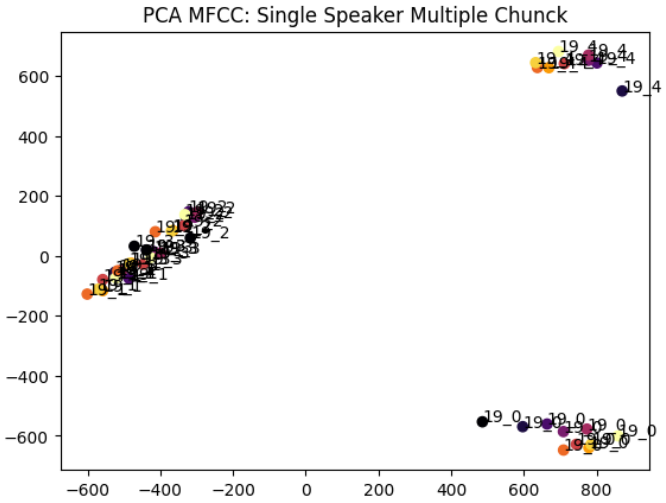


Figure 6.17: Data Visualization with PCA applied on MFCC of Multiple Speech Chunks of a Single Speaker at different distances from microphone

Wav2Vec2 Features Analysis This experiment calculates the PCA of MFCC calculated on different utterance chunks of the same speaker recorded at different distances from the microphone. The PCA is calculated in 2D because it can be visualized in a 2D axis. As shown in Fig. 6.18 using this feature for a distance analysis produces an outcome that is similar to a cluster, but this cluster is not well-formed so is not learnable by any learning model. The reason is the data distribution that is not representative of creating a regression or classification model.

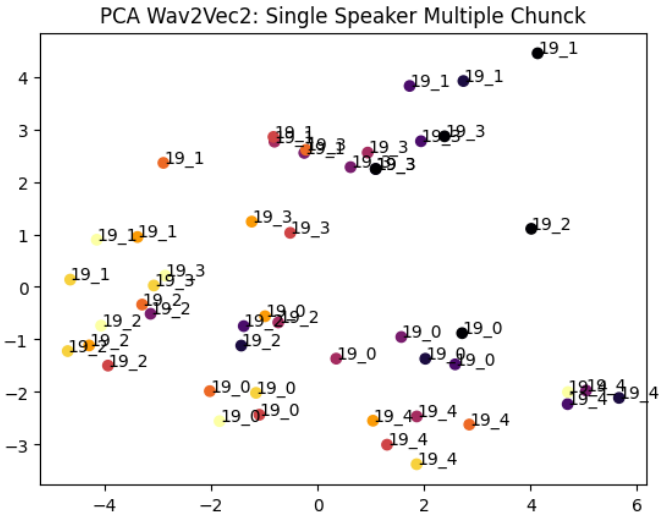


Figure 6.18: Data Visualization with PCA applied on Wav2Vec features extracted from Multiple Speech Chunks of a Single Speaker at different distances from microphone

Principal Component Analysis (PCA): Multiple Speaker Single Chunk

Raw Waveform Analysis This experiment shows Data visualized by PCA applied on Raw Waveform of different utterance chunks of different speakers recorded at different distances from the microphone in a range of 1 to 10 meters. As shown in Fig. 6.19, the Raw Waveform does not represent a discriminative feature to represent data in order to create a regression or classification problem.

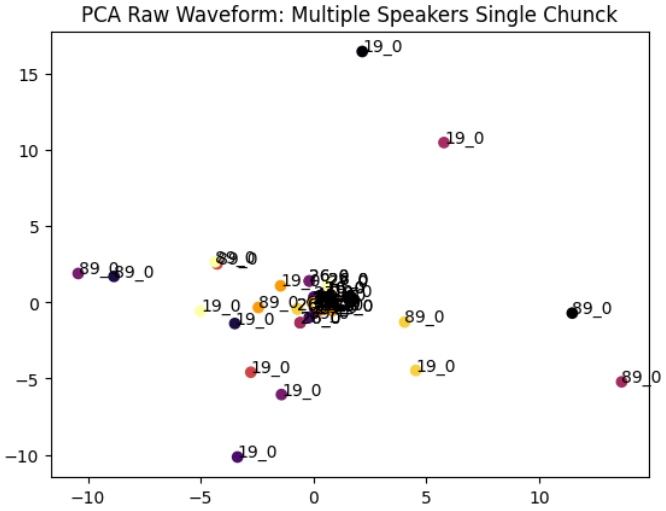


Figure 6.19: Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Spectrogram Analysis Fig. 6.20 shows a data visualization of PCA applied on Spectrogram extracted by a Single Utterance Chunk of Multiple Speakers.

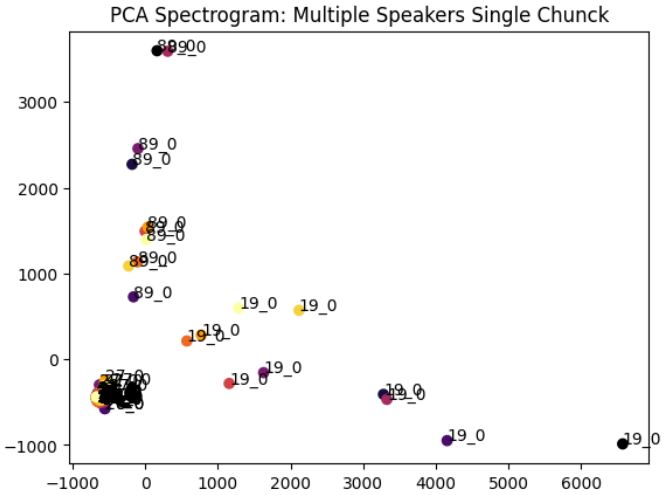


Figure 6.20: Data Visualization with PCA applied on Spectrograms of a Single Speech Chunk of Multiple Speakers at different distances from microphone

Mel Frequency Cepstrum Coefficents Fig. 6.21 shows a data visualization of PCA applied on MFCC extracted by a Single Utterance Chunk of Multiple Speakers.

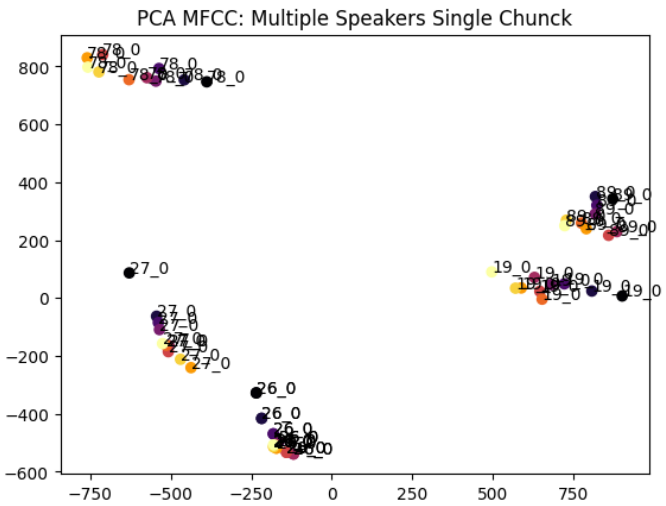


Figure 6.21: Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Wav2Vec2 Features Analysis Fig. 6.22 shows a data visualization of PCA applied on Wav2Vec2 Features extracted by Single Utterance Chunk of Multiple Speakers.

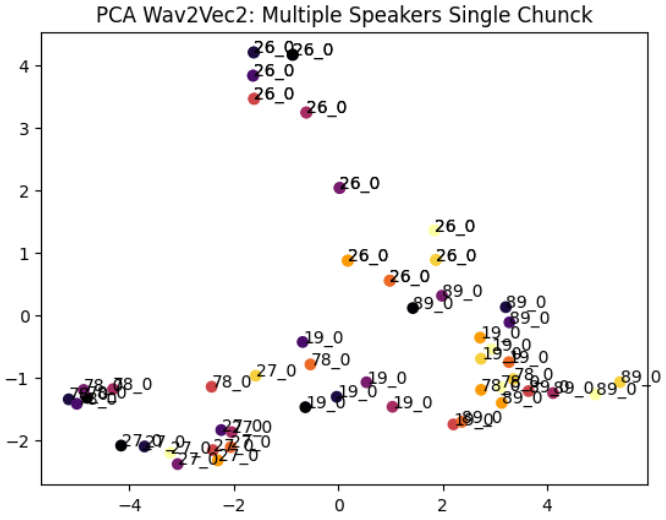


Figure 6.22: Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Principal Component Analysis (PCA): Multiple Speaker Multiple Chunks

Raw Waveform Analysis This experiment shows Data visualized by PCA applied on Raw Waveform of different utterance chunks of different speakers recorded at different distances from the microphone in a range of 1 to 10 meters. As shown in Fig. 6.23, the Raw Waveform does not represent a discriminative feature to represent data in order to create a regression or classification problem.

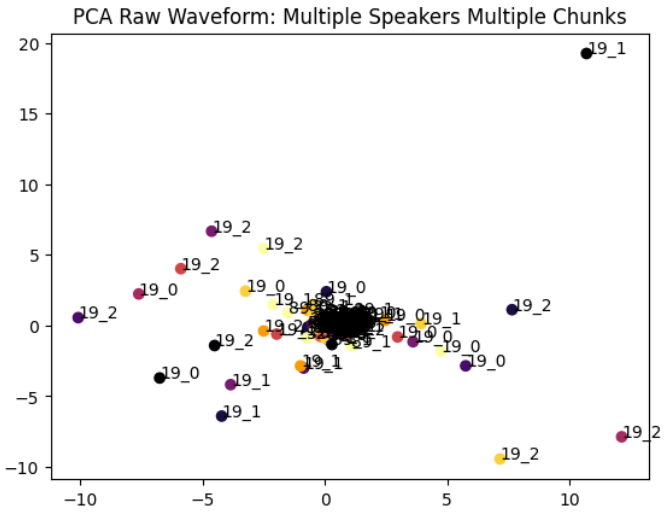


Figure 6.23: Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Spectrogram Analysis Fig. 6.24 shows a data visualization of PCA applied on Spectrogram extracted by a Single Utterance Chunk of Multiple Speakers.

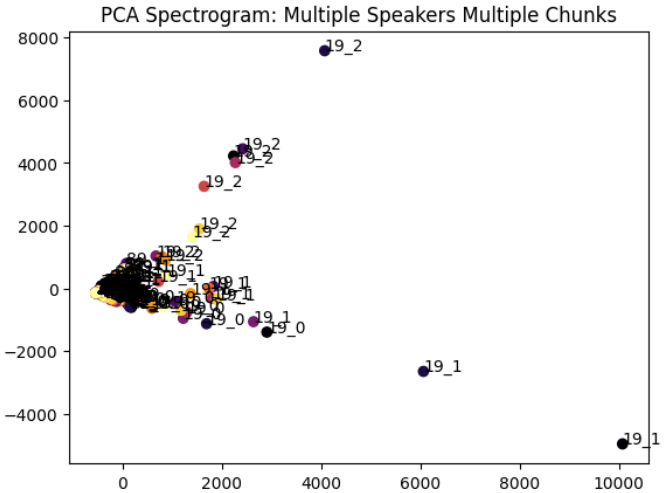


Figure 6.24: Data Visualization with PCA applied on Spectrograms of a Single Speech Chunk of Multiple Speakers at different distances from microphone

Mel Frequency Cepstrum Coefficents Fig. 6.25 shows a data visualization of PCA applied on MFCC extracted by a Single Utterance Chunk of Multiple Speakers.

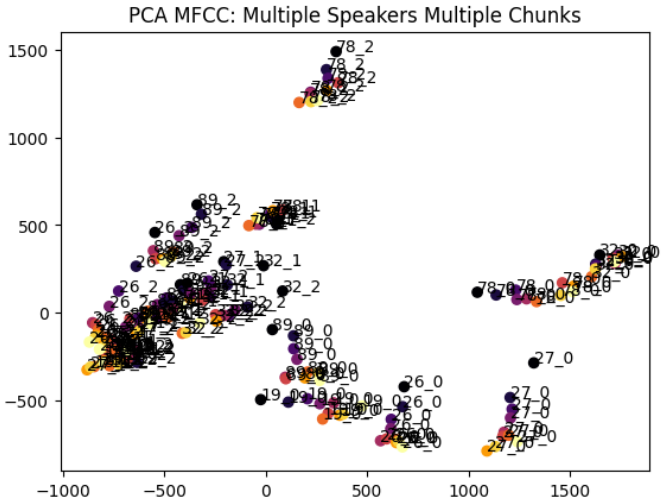


Figure 6.25: Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

Wav2Vec2 Features Analysis Fig. 6.26 shows a data visualization of PCA applied on Wav2Vec2 Features extracted by Single Utterance Chunk of Multiple Speakers.

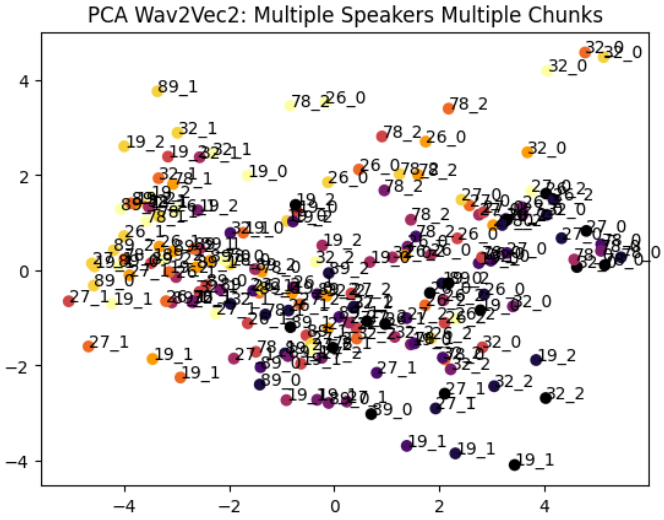


Figure 6.26: Data Visualization with PCA applied on Wav2Vec features extracted from a Single Speech Chunk of Multiple Speakers at different distances from microphone

3 Multiple Microphone Sound Source Localization

3.1 Room

As seen in the research activities chapter, particularly in the section related to sound source localization 4.9, there are numerous factors that influence the outcomes of the algorithms applied to the defined environment. Since recording audio datasets in academic and urban settings can be exceedingly challenging due to potential interferences in these locations, synthetic data is often preferred. In this study, it was feasible to conduct the experiments that will follow through the utilization of the aforementioned PyRoomAcoustics platform. This library enables the simulation of all the acoustic physics present in user-defined rooms, simplifying the process of generating data for testing the algorithms considered in this study.

An environment can be defined to create an office room as shown in the next figure (Fig. 6.27)

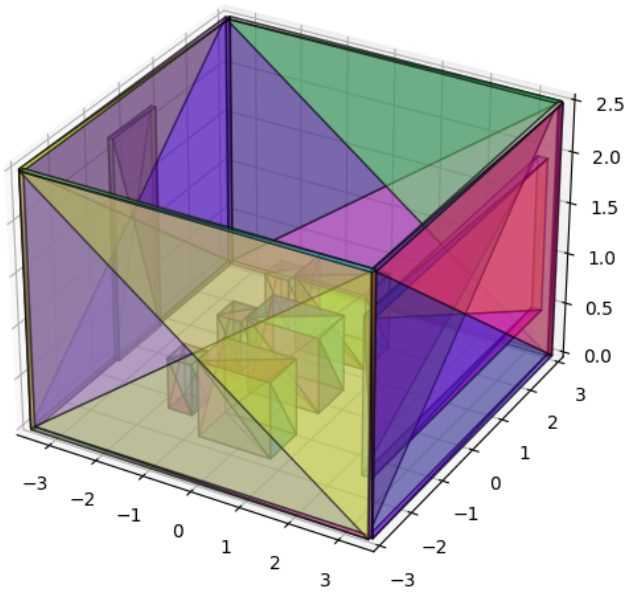


Figure 6.27: An Office Room simulated with PyRoom Acoustic with 3 Stations with chair and desk

To simplify the testing of sound source localization algorithms, the test environment has been reduced to a currently vacant room with the dimensions of an office. This adjustment allows for an initial evaluation while considering the variability of a limited set of parameters. The room size, expressed in three dimensions (3D), is (3.0, 2.5, 2.5) as shown in Fig.6.28. The room is reverberant with a Signal Noise Ratio (SNR) of 5 dB.

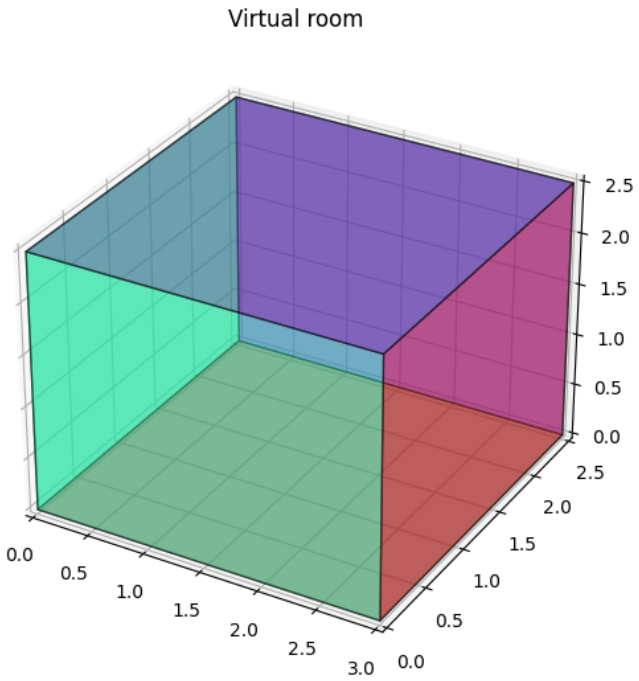


Figure 6.28: Empty Room simulated with PyRoom Acoustic

3.2 Microphone Type

As seen in 2.4.3, and 4.9.5, there exist various types of microphones, which could potentially be employed in simulations. In this study, cardioid microphones, i.e., directional microphones, have been considered. This choice stems from the fact that cardioid microphones are the most commonly found in security cameras and, generally, in everyday devices. They are preferred due to their cost-effectiveness and ease of production (Fig.6.29).

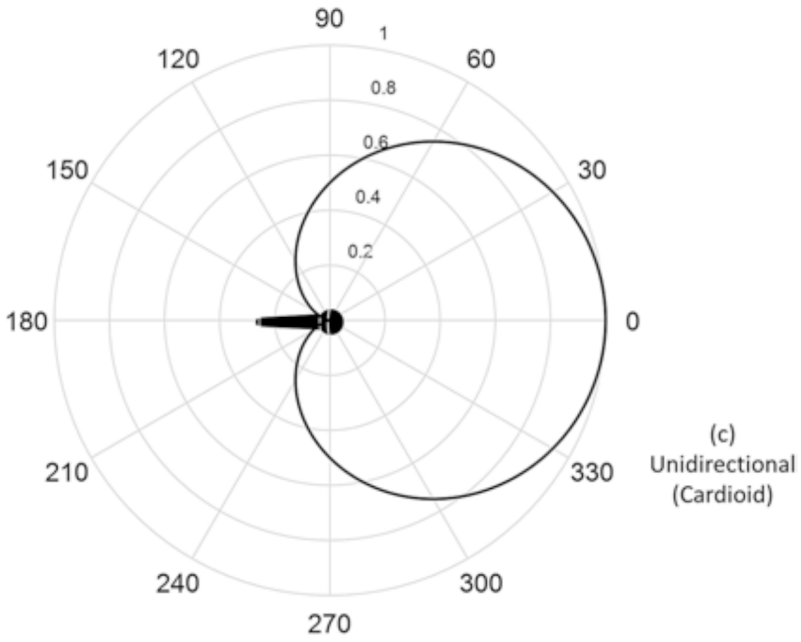


Figure 6.29: Cardioid Microphone for simulations

Nowadays Sound Source Localization Challenges like LDAS2023 and important conferences like ICASSP are focusing on this problem using ambisonics microphones that capture audio from the three different axes. But in normal context, they are not used because are really expansive and actually is really difficult to find this kind of microphone included in electronic devices.

3.3 Type of sources

Sound Source Localization problem can be divided into Single Source Sound Source Localization where a single source is present inside the environment and Multiple Sound Source Localization, in particular, is done with 2 speakers.

3.4 Microphones Configurations

As evident from the research phase conducted in the initial problem investigation, the various microphone configurations and the Signal-to-Noise Ratio (SNR) have a significant impact on source localization. This is because each algorithm leverages spatial properties and the resulting recordings to compute the position of one or more sources. The configurations adopted for sound source localization in the conducted experiments are as follows: Single microphone, Binaural, Triaural, Tetra-aural,

Linear Microphone Array, and Circular Microphone Array. As for the Non-coplanar 16-microphone configuration, was not considered due to its impracticality in non-specialized environments, rendering it of limited utility in forensic contexts.

Binaural Microphones

In this configuration microphones were placed in a binaural configuration, emulating Human Earing System. This experiment is done to prove the poor accuracy of this microphone's position. In Fig. 6.30 is possible to see the position of microphones and their relative position of 22 cm, which simulates ears distance. In Fig. 6.31 is possible to see area of recording of this configuration.

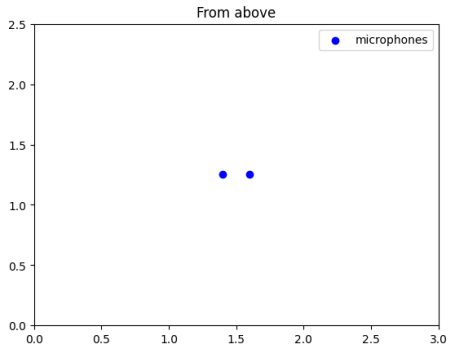


Figure 6.30: 2D perspective of microphones configuration

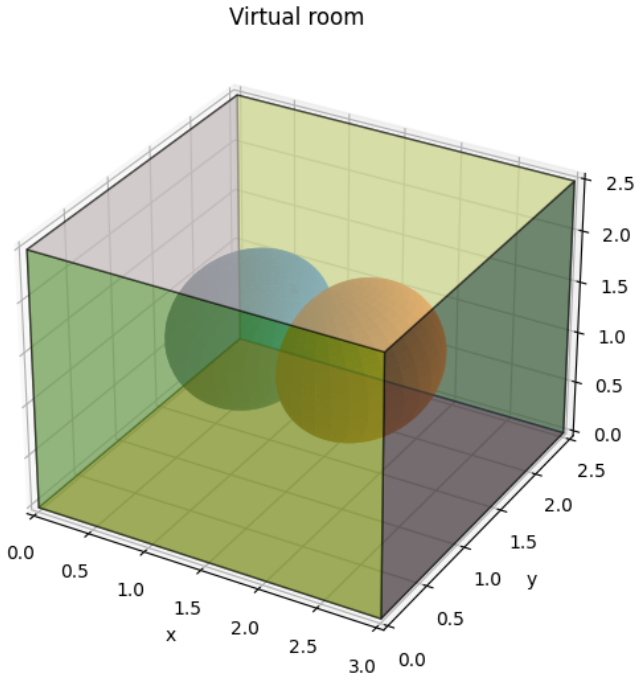


Figure 6.31: 3D perspective of microphones configuration with area covered by cardioid microphones

Triaural Microphones

Triaural is a configuration where three microphones stand outside the source location. For this reason in this configuration, microphones are placed near the corner of the room in this way the recorded area is wider with the consequence that the source can be placed in a bigger set of positions inside the room. In the next figure, on the left (Fig. 6.32) is shown microphones configuration in the 2D plane that represent the room, while on the right 6.33 is shown a 3D representation of microphones configuration inside the room and the area covered by microphones during the simulations.

Tetra-aural Microphones

Tetra-aural is a configuration where four microphones stand in the corners of the room. Same for the Triaural configuration, here the source location is inside the area delimitated by microphones. Here the microphones coordinates are: [0.0, 0.0, 1.80], [0.0, 2.5, 1.80], [3.0, 2.5, 1.80], [3.0, 0.0, 1.80]; while the microphones orientation (that

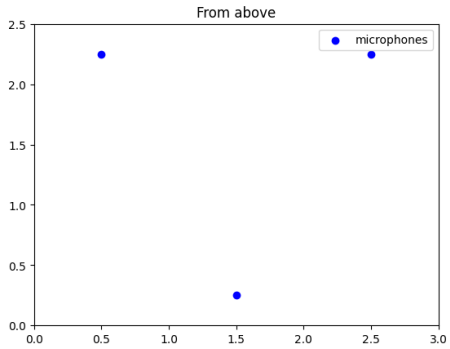


Figure 6.32: 2D perspective of Triaural microphones configuration

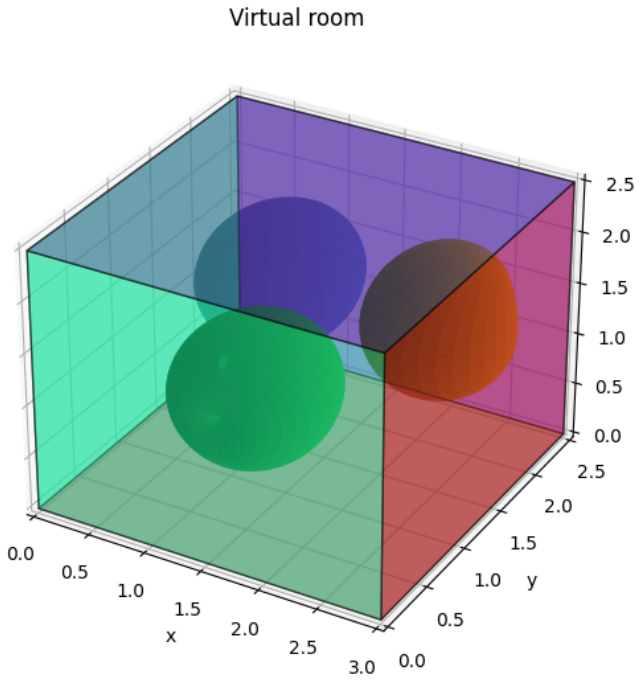


Figure 6.33: 3D perspective of Triaural microphones configuration with area covered by cardioid microphones

follow microphone coordinates) are [azimuth=45, colatitude=90], [azimuth=315, colatitude=90], [azimuth=225, colatitude=90], [azimuth=135, colatitude=90]. The configuration in 2D is shown in Fig. 6.34 and in 3D in Fig. 6.35

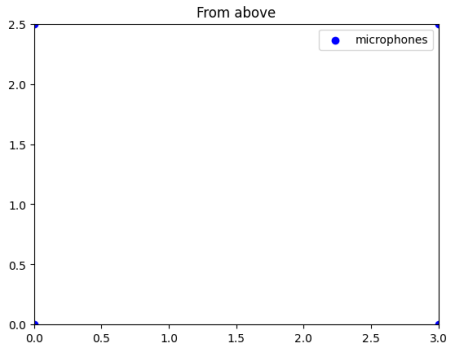


Figure 6.34: 2D perspective of microphones configuration

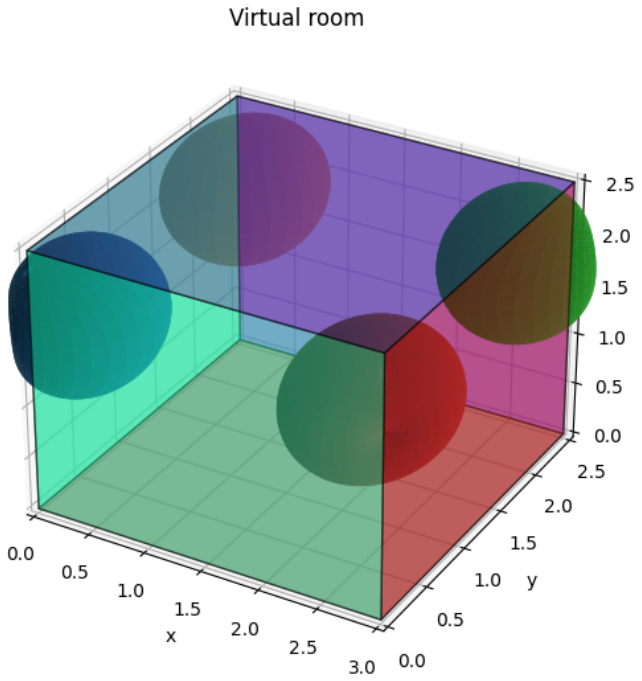


Figure 6.35: 3D perspective of microphones configuration with area covered by cardioid microphones

Circular Array Microphones

The circular Array Microphones configuration is composed of eight microphones placed in the centre of the room with a radius of 10cm. The configuration came from [252] where they used a Deep Neural Network based on SampleCNN trained with recordings that came from this kind of configuration to separate space in more subspaces.

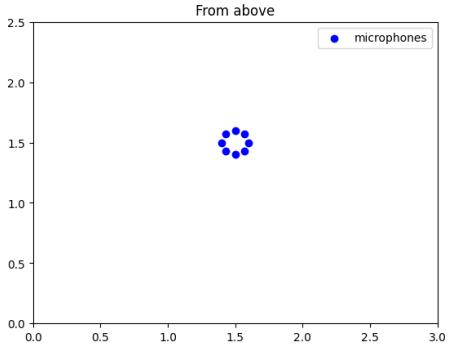


Figure 6.36: 2D perspective of microphones configuration

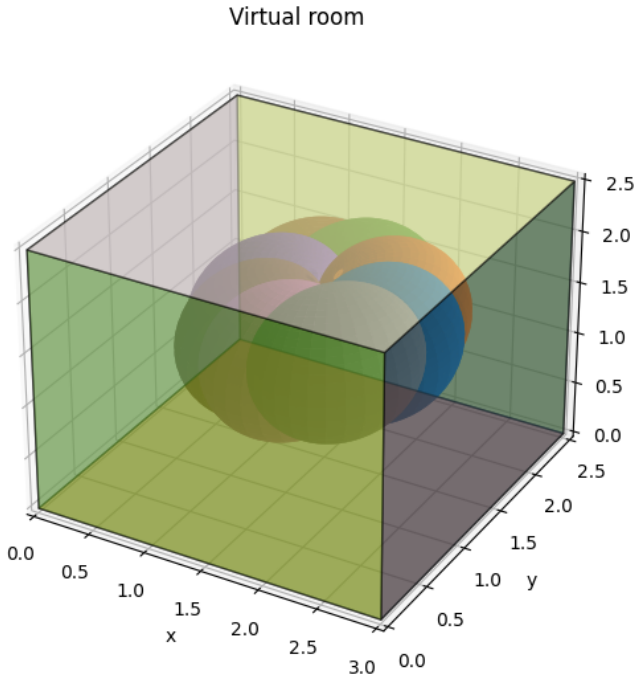


Figure 6.37: 3D perspective of microphones configuration with area covered by cardioid microphones

3.5 Methods Applied

MUSIC

MUltiple SIgnal Classification algorithm is used for SSL. The number of sound sources can be one or more. There are let's say M sensors that receive time-delayed signals from sound sources with reference to a particular sensor [235]. Matrix X is a function of incident signal, center frequency, incident signal angle and number of sensors M. Matrix X represents the received sound signals. The auto-correlation matrix R_{XX} of X finds the correlation between rows of X. Number of eigenvectors that are associated with signal subspace is equal to the number of sources. If the number of sensors and sound sources is known, matrix $U_n \in C^{M \times M-D}$ can be formed. U_n consists of a set of eigenvectors associated with noise subspace. The matrix U_n consists of eigenvectors whose eigenvalues λ_{min} are the variance of the noise. λ_{min} occurs in clusters which decreases when more data is processed. Steering vectors corresponding to Difference of Arrival (DOA) are present in the signal subspace and are orthogonal to the noise subspace. Hence, we have $a^H(\hat{\Theta})U_n U_n^H a^H(\hat{\Theta})$ for $\hat{\Theta}$ corresponding to the DOA of the multi-path component. DOAs can be known by locating peaks of MUSIC spatial spectrum:

$$P_{MUSIC}(\hat{\Theta}) = \frac{1}{a^H(\hat{\Theta})U_n U_n^H a^H(\hat{\Theta})} \tag{6.1}$$

1 Speaker Figure 6.38 displays the results of Sound Source Localization experiments using the MUSIC algorithm for a scenario involving a single speaker. The objective is to estimate the azimuth angles (horizontal direction) of the sound source, measured in degrees. The results are presented for four different microphone configurations: Binaural Configuration: In this configuration, the real azimuth angle was 321.08 degrees, while the algorithm recovered an angle of 303 degrees. This resulted in an error of 18.07 degrees. Triaural Configuration: For the triaural configuration, the real azimuth angle was -26.56 degrees, but the algorithm estimated an angle of 270 degrees, resulting in an error of 63.43 degrees. Tetra-Aural Configuration: In the tetra-aural configuration, the real azimuth angle was -38.92 degrees, and the algorithm estimated an angle of 318 degrees. The error in this case was 3.07 degrees. Circular Array Configuration: In the circular array configuration, the real azimuth angle was 45 degrees, and the algorithm estimated an azimuth angle of 41 degrees, resulting in a small error of 4 degrees.

2 Speaker Figure 6.39 presents the results of Sound Source Localization experiments using the MUSIC algorithm for a scenario involving two speakers. The objective is to estimate the azimuth angles (horizontal direction) of the two sound sources, measured in degrees. The results are presented for four different microphone configurations: Binaural Configuration: In the binaural configuration, the real azimuth angles for the two speakers were [270, 45] degrees. The algorithm estimated angles of [30, 330] degrees, resulting in errors of [15, 60] degrees for the respective speakers. Triaural Configuration: For the triaural configuration, the real azimuth angles for the two speakers were [45, 270] degrees. The algorithm estimated angles of [181, 218] degrees, leading to errors of [136, 52] degrees for the respective speakers. Tetra-Aural Configuration: Similarly, in the tetra-aural configuration, the real azimuth angles for both speakers were [45, 270] degrees. The algorithm recovered angles of [90, 270] degrees, resulting in errors of [45, 0] degrees. Circular

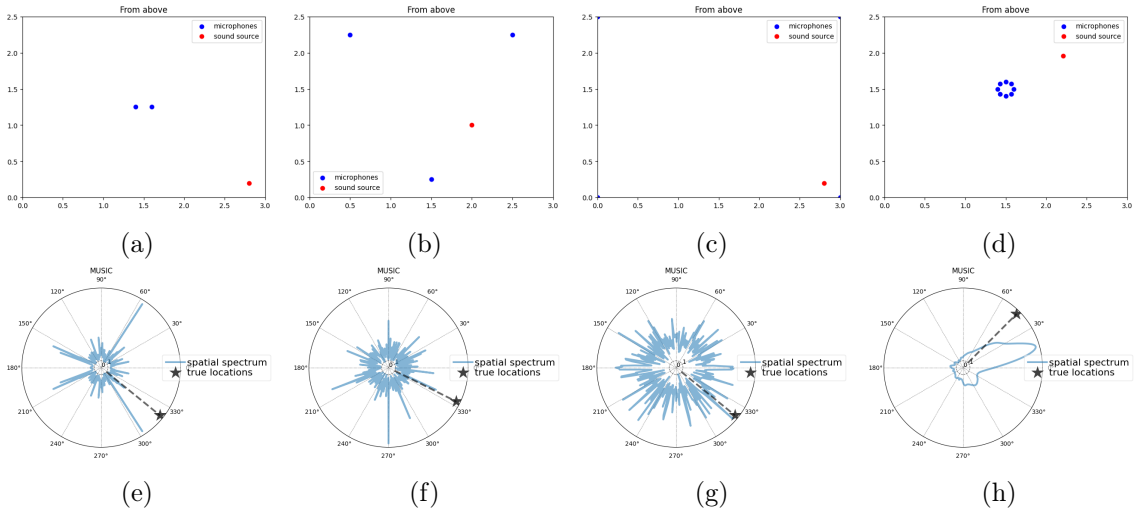


Figure 6.38: **MUSIC with 1 speaker** values of azimuth for configuration experiments in degrees:
a) Binaural configuration: Real: 321.08, Recovered: 303, Error: 18.07
b) Triaural configuration: Real: -26.56, Recovered: 270, Error: 63.43
c) Tetra-aural configuration: Real: -38.92, Recovered: 318, Error: 3.07
d) Circular Array configuration: Real: 45, Recovered azimuth: 41, Error: 4

Array Configuration: In the circular array configuration, the real azimuth angles for the two speakers remained [45, 270] degrees. The algorithm estimated angles of [43, 274] degrees, resulting in small errors of [2, 4] degrees for the respective speakers.

Certainly, I can explain the notation used in the results. In the provided results, the notation [a1, a2] represents the azimuth angles of two different speakers or sound sources. Specifically:

"a1" refers to the azimuth angle of the first speaker or sound source. "a2" refers to the azimuth angle of the second speaker or sound source. Azimuth angles represent the horizontal direction or angle at which a sound source is located relative to a reference point, usually measured in degrees. Therefore, [a1, a2] provides a pair of azimuth angles, one for each speaker, to describe their respective locations in the horizontal plane.

For example, in the binaural configuration, the real azimuth angles [270, 45] indicate that the first speaker is located at an azimuth angle of 270 degrees, while the second speaker is located at an azimuth angle of 45 degrees. The estimated angles [30, 330] represent the algorithm's attempt to determine the positions of these two speakers.

This notation helps distinguish between multiple sound sources and provides a clear representation of their azimuth angles for analysis and evaluation.

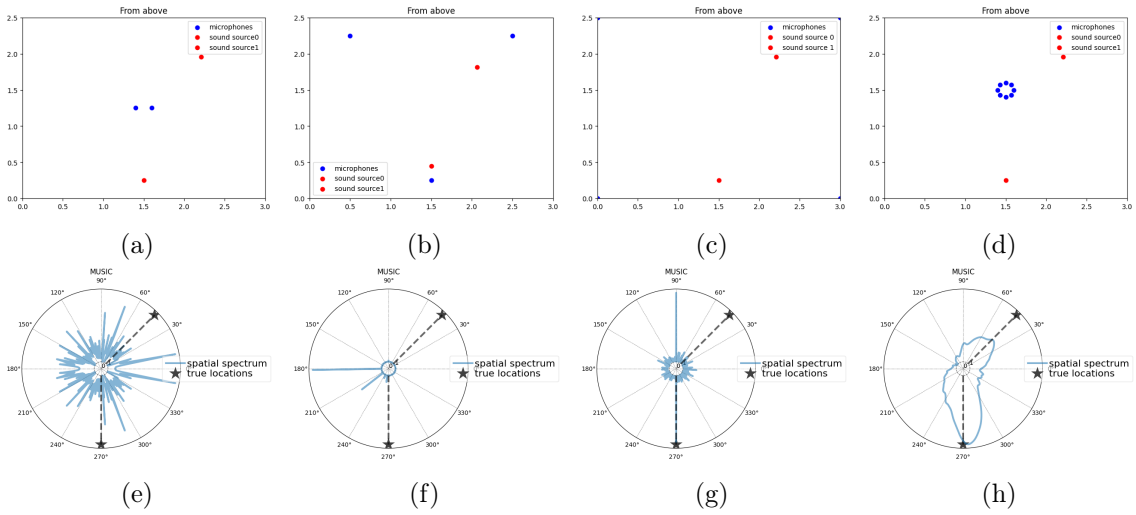


Figure 6.39: **MUSIC with 2 speakers:** values of azimuth for configuration experiments in degrees:
a) Binaural configuration: Real azimuth: [270, 45], Recovered: [30, 330], Error: [15, 60]
b) Triaural configuration: Real: [45, 270], Recovered: [181, 218], Error: [136, 52]
c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0]
d) Circular Array configuration: Real: [45, 270], Recovered: [43, 274], Error: [2, 4]

NormMUSIC

Since NormMUSIC has a more robust performance, is recommended to use NormMUSIC over MUSIC. When MUSIC is used as a baseline for publications, we recommend to use both NormMUSIC and MUSIC.[229]

1 Speaker : Figure 6.40 illustrates the results of Sound Source Localization experiments using the NormMUSIC algorithm for a single-speaker scenario. The objective of these experiments is to accurately estimate the azimuth angle (horizontal direction) of the sound source, measured in degrees. The results are presented for four different microphone configurations:

1. Binaural Configuration: In the real scenario, the azimuth angle was 321.08 degrees, while the algorithm recovered an angle of 286 degrees. This resulted in an error of 35.07 degrees.

2. Triaural Configuration: In the real scenario, the azimuth angle was -26.56 degrees, but the algorithm estimated an angle of 90 degrees. This led to an error of 116.56 degrees.

3. Tetra-Aural Configuration: In the real scenario, the azimuth angle was -38.92 degrees, and the algorithm estimated an angle of 284 degrees. The error in this case was 37.07 degrees.

4. Circular Array Configuration: In this configuration, the real azimuth angle was 45 degrees, while the algorithm estimated an angle of 39 degrees, resulting in a small error of 6 degrees.

These results provide insights into the performance of the NormMUSIC algorithm for sound source localization across different microphone setups. It is evident that the accuracy of azimuth estimation varies based on the microphone configuration, with some configurations exhibiting higher errors than others.

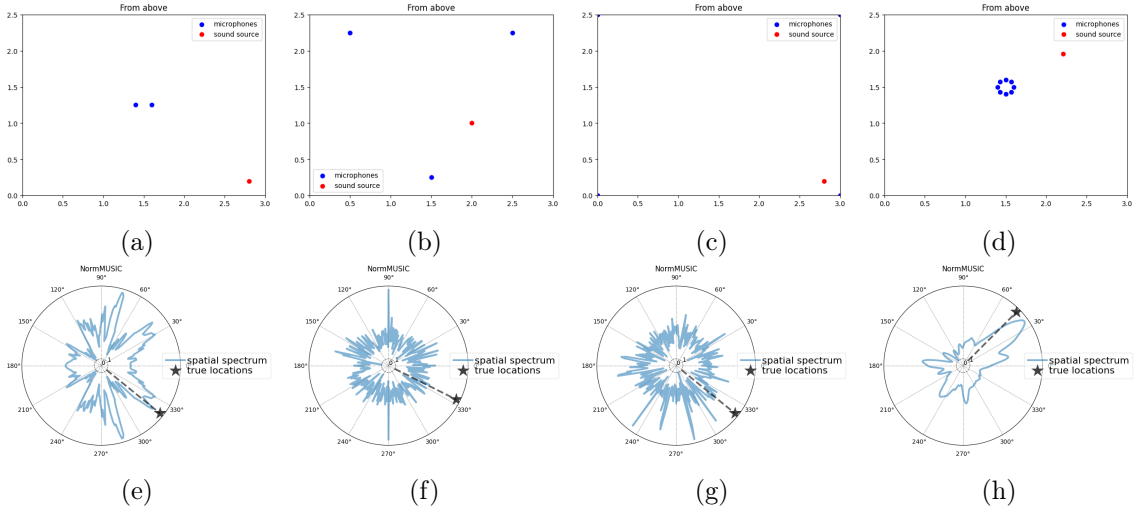


Figure 6.40: **NormMUSIC with 1 speaker:** azimuth for experiments (in degrees):
a) Binaural configuration: Real: 321.08, Recovered: 286, Error: 35.07
b) Triaural configuration: Real: -26.56, Recovered: 90, Error: 116.56
c) Tetra-aural configuration: Real: -38.92, Recovered: 284, Error: 37.07
d) Circular Array configuration: Real: 45, Recovered: 39, Error: 6.

2 Speaker Figure 6.41 presents the results of Sound Source Localization experiments using the NormMUSIC algorithm for a scenario involving two speakers. The objective is to estimate the azimuth angles (horizontal direction) of both sound sources, measured in degrees. The results are shown for four different microphone configurations:

1. Binaural Configuration: In this configuration, the real azimuth angles for the two speakers were [45, 270] degrees. The algorithm recovered angles of [157, 203] degrees, resulting in errors of [112, 67] degrees for the respective speakers.

2. Triaural Configuration: For the triaural configuration, the real azimuth angles for the two speakers were also [45, 270] degrees. The algorithm estimated angles of [90, 270] degrees, leading to errors of [45, 0] degrees for the respective speakers.

3. Tetra-Aural Configuration: Similarly, in the tetra-aural configuration, the real azimuth angles for both speakers were [45, 270] degrees. The algorithm recovered angles of [90, 270] degrees, resulting in errors of [45, 0] degrees.

4. Circular Array Configuration: In the circular array configuration, the real azimuth angles for the two speakers remained [45, 270] degrees. The algorithm estimated angles of [42, 271] degrees, leading to small errors of [3, 1] degrees for the respective speakers.

These results provide insights into the performance of the NormMUSIC algorithm for sound source localization when dealing with two speakers. The accuracy of azimuth estimation varies based on the microphone configuration, and the errors are reported for each speaker separately. This information is valuable for optimizing microphone placement and algorithm selection in scenarios involving multiple sound sources.

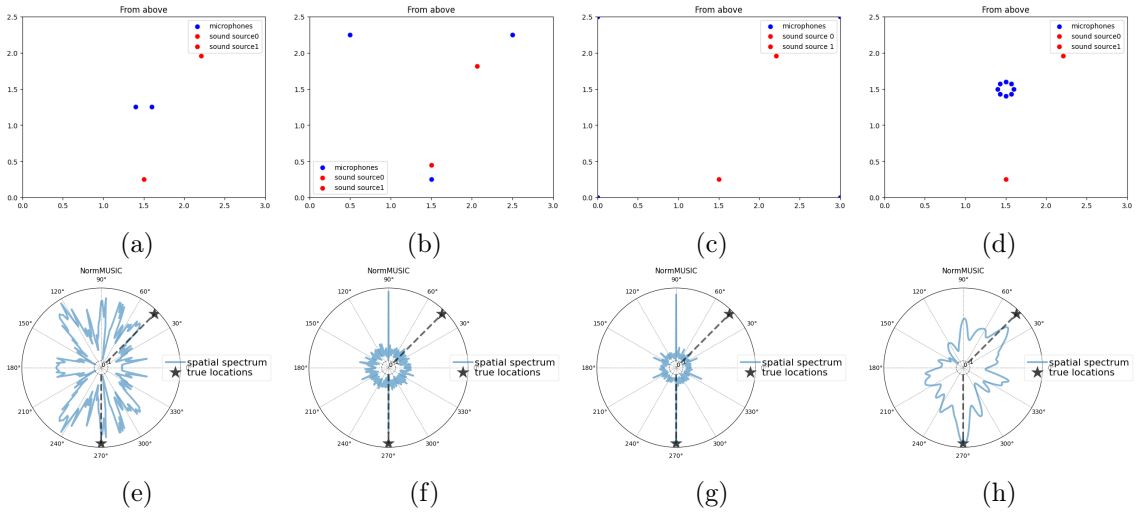


Figure 6.41: **NormMUSIC with 2 speakers:** azimuth for experiments (in degrees):
a) Binaural configuration: Real: [45, 270], Recovered: [157, 203.], Error: [112, 67]
b) Triaural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0]
c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0]
d) Circular Array configuration: Real: [45, 270], Recovered: [42, 271], Error:[3,1]

TOPS: Test of Orthogonality of Projected Subspaces

This technique [296] estimates DOAs by measuring the orthogonal relation between the signal and the noise subspaces of multiple frequency components of the sources. TOPS can be used with arbitrary shaped one-dimensional (1-D) or two-dimensional (2-D) arrays. Unlike other coherent wideband methods, such as the coherent signal subspace method (CSSM) and WAVES, the new method does not require any preprocessing for initial values. The performance of those wideband techniques and incoherent MUSIC is compared with that of the new method through computer simulations. The simulations show that this new technique performs better than others in mid signal-to-noise ratio (SNR) ranges, while coherent methods work best in low SNR and incoherent methods work best in high SNR. Thus, TOPS fills a gap between coherent and incoherent methods.

Experiments on different microphone configurations with 1 Speaker : Figure 6.42 displays the results of Sound Source Localization experiments using the TOPS algorithm for a scenario involving a single speaker. The objective is to estimate the azimuth angles (horizontal direction) of the sound source, measured in degrees. The results are presented for four different microphone configurations: Binaural Configuration: In the binaural configuration, the real azimuth angle was 321.08 degrees, while the algorithm estimated an angle of 241 degrees. This resulted in an error of 80.07 degrees. Triaural Configuration: For the triaural configuration, the real azimuth angle was -26.56 degrees, but the algorithm estimated an angle of 235 degrees, resulting in an error of 98.43 degrees. Tetra-Aural Configuration: In the tetra-aural configuration, the real azimuth angle was -38.92 degrees, and the algorithm estimated an angle of 84 degrees. The error in this case was 122.92 degrees. Circular Array Configuration: In the circular array configuration, the real azimuth angle was 45 degrees, and the algorithm estimated an azimuth angle of 38 degrees, resulting in a small error of 7 degrees.

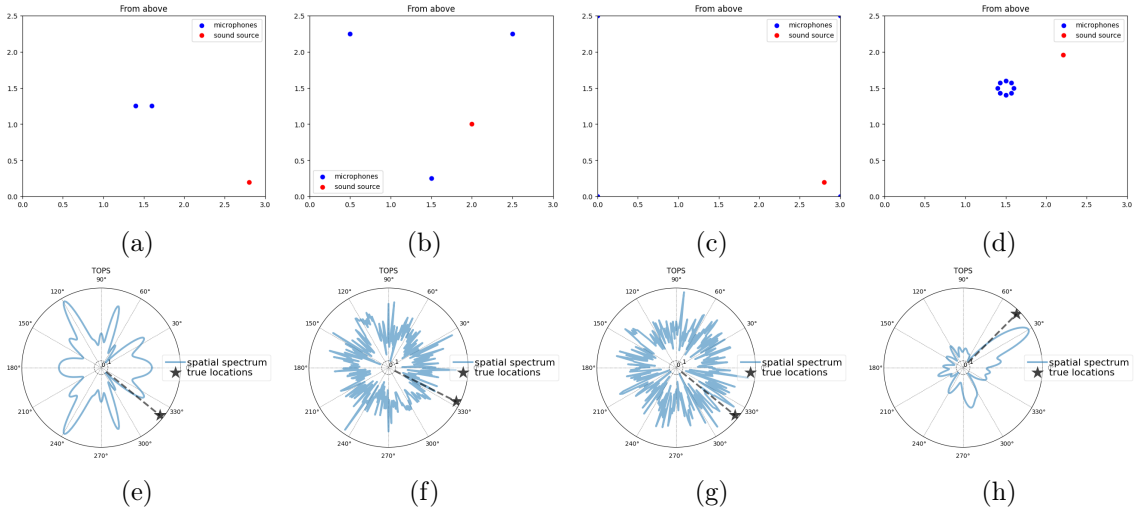


Figure 6.42: **TOPS with 1 speaker:** azimuth for experiments (in degrees):
a) Binaural configuration: Real: 321.08, Recovered: 241, Error: 80.07
b) Triaural configuration: Real: -26.56, Recovered: 235, Error: 98.43
c) Tetra-aural configuration: Real: -38.92, Recovered: 84, Error: 122.92
d) Circular Array configuration: Real: 45, Recovered: 38, Error:7

2 Speaker Figure 6.43 presents the results of Sound Source Localization experiments using the TOPS algorithm for a scenario involving two speakers. The objective is to estimate the azimuth angles (horizontal direction) of the two sound sources, measured in degrees. The results are presented for four different microphone configurations: Binaural Configuration: Unfortunately, an error occurred during the execution of the experiment for the binaural configuration. The error message indicates that an index -1 is out of bounds for axis 0 with size 0. This suggests that the experiment could not be completed successfully for this configuration. Triaural Configuration: In the triaural configuration, the real azimuth angles for the two speakers were [45, 270] degrees. The algorithm estimated angles of [90, 147] degrees, resulting in errors of [45, 123] degrees for the respective speakers. Tetra-Aural Configuration: Similarly, in the tetra-aural configuration, the real azimuth angles for both speakers were [45, 270] degrees. The algorithm recovered angles of [90, 270] degrees, resulting in errors of [45, 0] degrees. Circular Array Configuration: In the circular array configuration, the real azimuth angles for the two speakers remained [45, 270] degrees. The algorithm estimated angles of [28, 268] degrees, resulting in errors of [17, 2] degrees for the respective speakers.

These results provide insights into the performance of the TOPS algorithm for sound source localization when dealing with two speakers. The accuracy of azimuth estimation varies based on the microphone configuration, and errors are reported for each speaker separately. Unfortunately, the experiment for the binaural configuration encountered an error during execution and could not be completed.

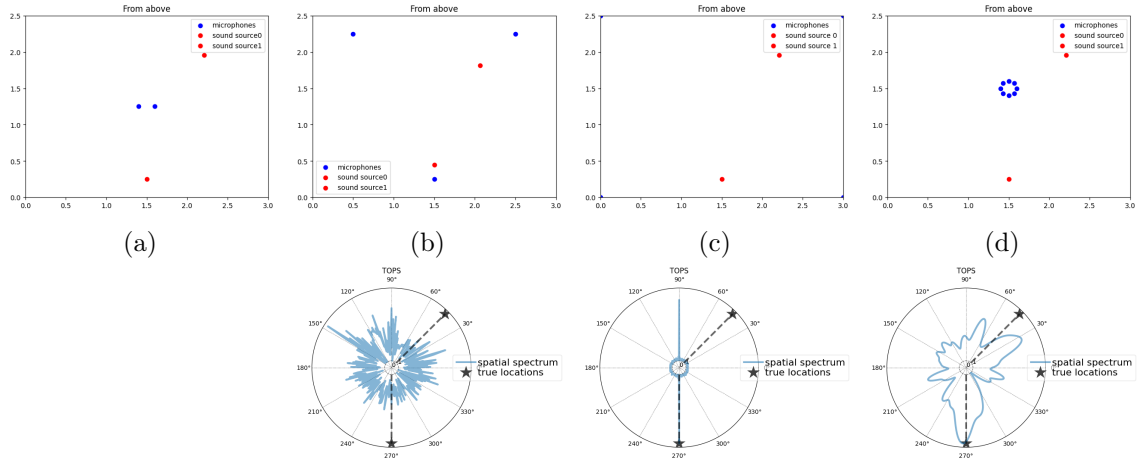


Figure 6.43: **TOPS with 2 speakers** azimuth for experiments (in degrees):
a) Binaural configuration: Error during execution -> index -1 is out of bounds for axis 0 with size 0
b) Triaural configuration: Real: [45, 270], Recovered: [90, 147], Error:[45, 123]
c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error:[45. 0.]
d) Circular Array configuration: Real: [45, 270], Recovered: [28, 268], Error:[17, 2]

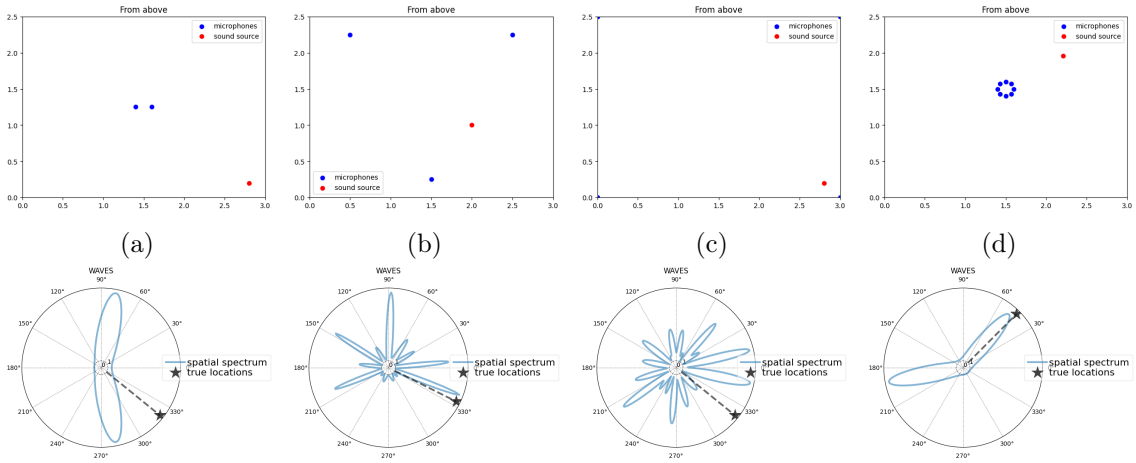
WAVES: Weighted Average of Signal Subspaces

Existing algorithms for wideband direction finding are mainly based on local approximations of the Gaussian log-likelihood around the true directions of arrival (DOAs), assuming negligible array calibration errors. Suboptimal and costly algorithms, such as classical or sequential beamforming, are required to initialize a local search that eventually furnishes DOA estimates. This multistage process may be nonrobust in the presence of even small errors in prior guesses about angles and number of sources generated by inherent limitations of the preprocessing and may lead to catastrophic errors in practical applications. This strategy combines a robust near-optimal data-adaptive statistic, called the weighted average of signal subspaces (WAVES) [70], with an enhanced design of focusing matrices to ensure a statistically robust preprocessing of wideband data. The overall sensitivity of WAVES to various error sources, such as imperfect array focusing, is also reduced with respect to traditional CSSM algorithms, as demonstrated by extensive Monte Carlo simulations.

1 Speaker Figure 6.44 displays the results of Sound Source Localization experiments using the WAVES algorithm. The objective is to estimate the azimuth angles (horizontal direction) of sound sources, measured in degrees. The results are presented for four different microphone configurations: In the binaural configuration, the real azimuth angle was 321.08 degrees, while the algorithm estimated an angle of 281 degrees. This resulted in an error of 40.07 degrees. For the triaural configuration, the real azimuth angle was -26.56 degrees, but the algorithm estimated an angle of 339 degrees, resulting in a small error of 5.56 degrees. In the tetra-aural configuration, the real azimuth angle was -38.92 degrees, and the algorithm estimated an angle of 348 degrees. The error in this case was 26.92 degrees. In the circular array configuration, the real azimuth angle was 45 degrees, and the algorithm estimated an azimuth angle of 49 degrees, resulting in a small error of 4 degrees.

These results provide insights into the performance of the WAVES algorithm for sound source localization in different microphone setups. The accuracy of azimuth estimation varies based on the microphone configuration, with some configurations exhibiting higher errors than others. These findings are valuable for optimizing microphone placement and algorithm selection in sound source localization applications.

2 Speaker Figure 6.45 presents the results of Sound Source Localization experiments using the WAVES algorithm. The objective is to estimate the azimuth angles (horizontal direction) of sound sources, measured in degrees. The results are presented for four different microphone configurations: Unfortunately, an error occurred during the execution of the experiment for the binaural configuration. The experiment encountered an error, and the azimuth angles for speakers could not be successfully estimated. In the triaural configuration, the real azimuth angles for the two speakers were [45, 270] degrees. The WAVES algorithm estimated angles of [196, 357] degrees, resulting in errors of [151, 87] degrees for the respective speakers. The tetra-aural configuration, it appears that the experiment encountered an error related to a singular matrix. Consequently, the azimuth angles for speakers could not be estimated. In the circular array configuration, the real azimuth angles for the two speakers were [45, 270] degrees. The WAVES algorithm estimated angles of [41,



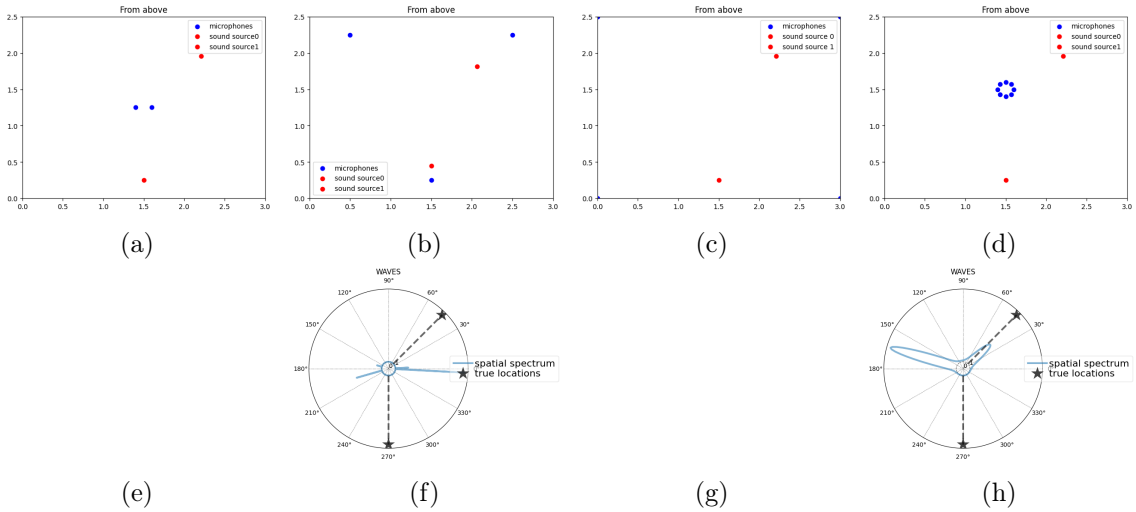
(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.44: **WAVES** values of azimuth for configuration experiments in degrees:
a) Binaural configuration: Real: 321.08, Recovered: 281, Error: 40.07
b) Triaural configuration: Real: -26.56, Recovered: 339, Error: 5.56
c) Tetra-aural configuration: Real: -38.92, Recovered: 348, Error: 26.92
d) Circular Array configuration: Real: 45, Recovered: 49, Error:4

164] degrees, resulting in errors of [4, 106] degrees for the respective speakers.

These results highlight the performance of the WAVES algorithm for localizing speakers in different microphone configurations. While some configurations yielded accurate estimations, others encountered errors or exhibited higher errors. These findings provide insights into the algorithm's limitations and areas for potential improvement in sound source localization tasks.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.45: **WAVES** speakers' azimuth for experiments (in degrees):
a) Binaural configuration: Error
b) Triaural configuration: Real : [45, 270], Recovered: [196, 357], Error: [151, 87]
c) Tetra-aural configuration: Error -> Singular Matrix
d) Circular Array configuration: Real : [45, 270.], Recovered: [41, 164], Error: [4, 106]

SRP-PHAT: Steered-Response Power-Phase Transform

The Steered-Response Power (SRP) map has been extensively employed in this regard. It involves applying delay-and-sum beamformers towards each potential grid position and measuring the energy originating from these directions. The PHAT version of SRP, known to be more resilient to reverberation, is widely adopted. In practical terms, the PHAT-based SRP map can be computed by averaging the GCC-PHAT values computed across all microphone pairs (DiBiase et al., 2001) [72]:

$$P(x) = \sum_{m_1=1}^{M} \sum_{m_2=m_1+1}^{M} r_{1,2}(\tau_{m_1,m_2}(x)) \tag{6.2}$$

Here, $\tau_{m_1,m_2}(x)$ denotes the delay between microphones m_1 and m_2 corresponding to spatial position x

Experiments conducted consist of applying this algorithm to 1 speaker and 2 speakers configuration.

1 Speaker Figure 6.46 presents the results of Sound Source Localization experiments using the SRP-PHAT algorithm. The objective is to estimate the azimuth angles (horizontal direction) of sound sources, measured in degrees. The results are presented for four different microphone configurations: In the binaural configuration, the real azimuth angle was 321.08 degrees, while the SRP-PHAT algorithm estimated an azimuth angle of 272 degrees. This resulted in an error of 49.07 degrees. For the triaural configuration, the real azimuth angle was -26.56 degrees, but the SRP-PHAT algorithm estimated an azimuth angle of 90 degrees. This led to an error of 116.56 degrees. In the tetra-aural configuration, the real azimuth angle was -38.92 degrees, and the SRP-PHAT algorithm estimated an azimuth angle of 306 degrees. The error in this case was 15.07 degrees. In the circular array configuration, the real azimuth angle was 45 degrees, and the SRP-PHAT algorithm estimated an azimuth angle of 39 degrees. This resulted in a relatively small error of 6 degrees.

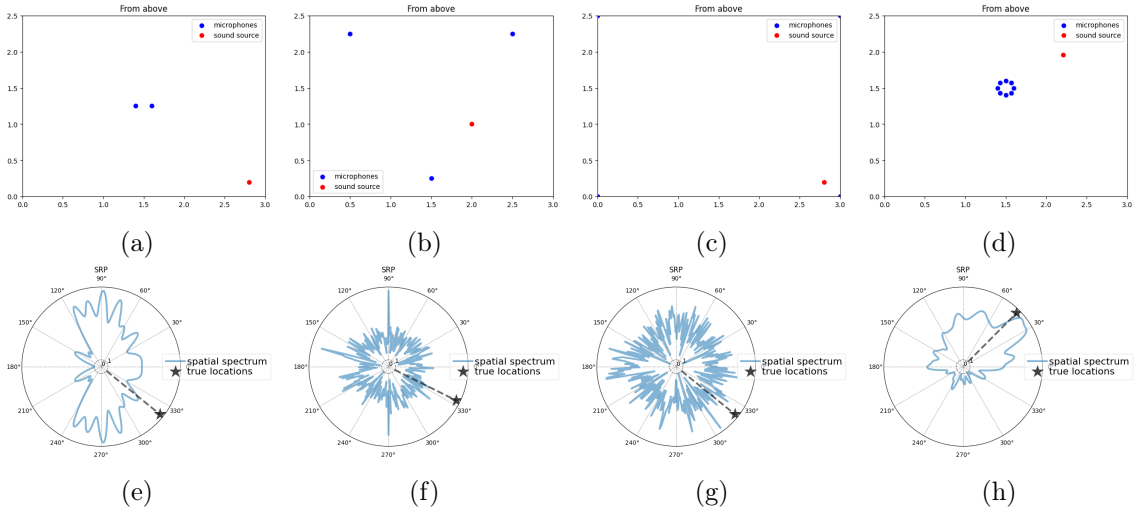


Figure 6.46: **SRP-PHAT** speaker azimuth for experiments (in degrees):
a) Binaural configuration: Real: 321.08, Recovered: 272, Error: 49.07
b) Triaural configuration: Real: -26.56, Recovered: 90, Error: 116.56
c) Tetra-aural configuration: Real: -38.92, Recovered: 306, Error: 15.07
d) Circular Array configuration: Real: 45, Recovered: 39, Error: 6

2 Speaker Figure 6.47 presents the results of Sound Source Localization experiments using the SRP-PHAT algorithm. The objective is to estimate the azimuth angles (horizontal direction) of sound sources, measured in degrees. The results are presented for four different microphone configurations: In the binaural configuration, the real azimuth angles were [45, 270] degrees for two different sources. The SRP-PHAT algorithm estimated azimuth angles of [90, 270] degrees for these sources. The error for the first source was 45 degrees, while there was no error (0 degrees) for the second source. For the triaural configuration, the real azimuth angles were [45, 270] degrees for two different sources. The SRP-PHAT algorithm estimated azimuth angles of [90, 270] degrees for these sources. The error for both sources was 45 degrees. In the tetra-aural configuration, the real azimuth angles were [45, 270] degrees for two different sources. The SRP-PHAT algorithm estimated azimuth angles of [90, 270] degrees for these sources. The error for both sources was 45 degrees. In the circular array configuration, the real azimuth angles were [45, 270] degrees for two different sources. The SRP-PHAT algorithm estimated azimuth angles of [88, 271] degrees for these sources. The error for the first source was 43 degrees, while the error for the second source was 1 degree.

These results demonstrate the performance of the SRP-PHAT algorithm in localizing speakers for various microphone configurations. In some cases, the algorithm was able to accurately estimate the azimuth angles with minimal error, while in others, there was a slight deviation from the real angles. The findings contribute to our understanding of the algorithm's capabilities in sound source localization tasks.

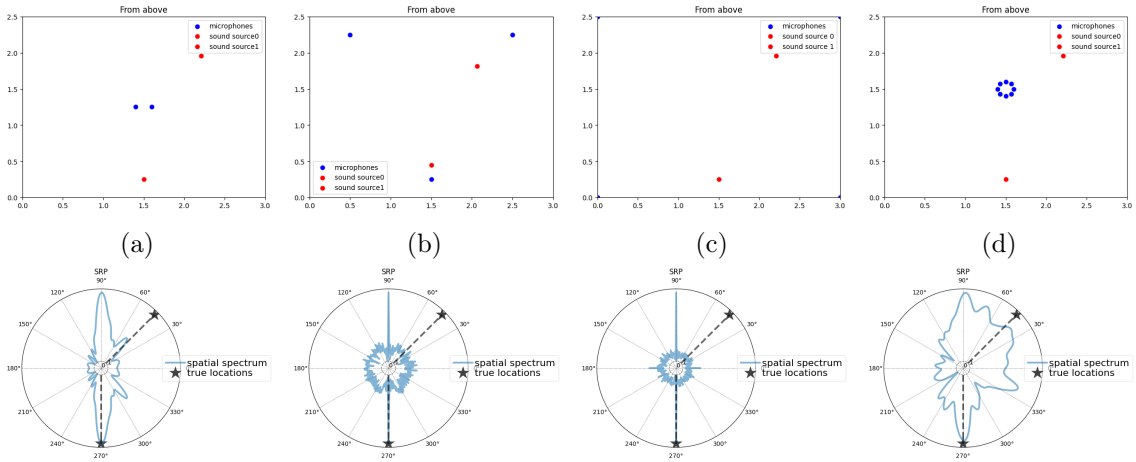


Figure 6.47: **SRP-PHAT** speakers' azimuth for experiments (in degrees):
a) Binaural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0]
b) Triaural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0]
c) Tetra-aural configuration: Real: [45, 270], Recovered: [90, 270], Error: [45, 0]
d) Circular Array configuration: Real: [45, 270] Recovered: [88, 271], Error: [43, 1]

GCC-PHAT: General Cross Correlation on Phase Transform

The generalized cross-correlation (CC) with phase transform (GCC-PHAT) is one of the most employed methods when dealing with a 2-microphone array [127]. To estimate TDOA, the delay between the cross-correlation between two signals should be maximum. Phase transform GCC increases its robustness. Let be x_i and x_j be two signals:

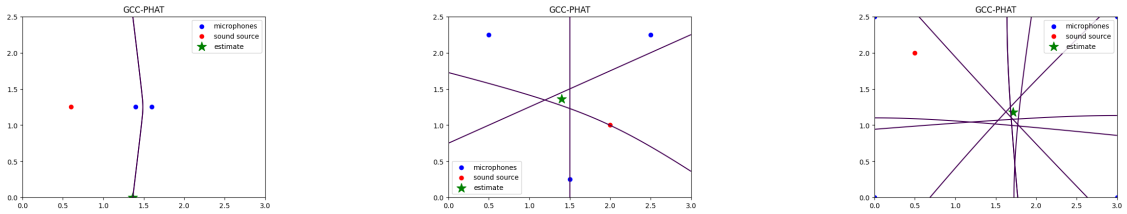
$$\hat{G}_{PHAT}(f) = \frac{X_i(f)[X_j(f)]^*}{|X_i(f)[X_j(f)]^*|} \tag{6.3}$$

where $X_i(f)$ and $X_j(f)$ are Fourier transforms of two signals and $[]^*$ is complex conjugate. The TDOA for two microphones is given as

$$\hat{d}_{PHAT}(i,j) = {}^{argmax}_{d}\ (\hat{R}_{PHAT}(d)) \tag{6.4}$$

where $\hat{R}_{PHAT}(d)$ is the inverse Fourier transform.

1 Speaker This figure presents the results of experiments conducted using the GCC-PHAT algorithm for sound source localization. The goal of these experiments was to estimate the Cartesian coordinates (x, y) of sound sources in meters, representing the positions of sound sources in a 2D plane. The results are provided for three different microphone configurations: In the binaural configuration, the ground truth coordinates of the sound source were [0.6, 1.25] meters. The GCC-PHAT algorithm estimated the coordinates as [1.36, 0]. This estimation resulted in an error of [-0.76, 1.25] meters. For the triaural configuration, the ground truth coordinates were [2.0, 1.0] meters. The GCC-PHAT algorithm estimated the coordinates as [1.39, 1.36] meters. The estimation error was [0.61, 0.36] meters. In the tetra-aural configuration, the ground truth coordinates of the sound source were [0.5, 2.0] meters. The GCC-PHAT algorithm estimated the coordinates as [1.71, 1.17] meters. This estimation resulted in an error of [-1.21, 0.82] meters. These results provide valuable insights into the accuracy and performance of the GCC-PHAT algorithm for sound source localization across different microphone configurations.



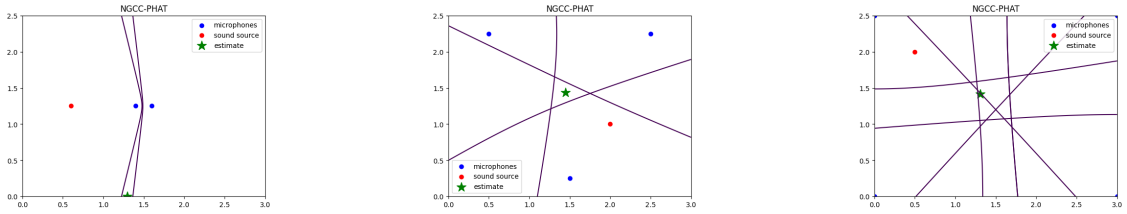
(a) (b) (c)

Figure 6.48: GCC-PHAT: Values of [x,y] in cartesian axes per configuration in meters:
a) Binaural configuration: Groundtruth: [0.6 1.25], GCC estimate: [1.36 0.]
b) Triaural configuration: Groundtruth: [2. 1.], GCC estimate: [1.39 1.36]
c) Tetra-aural configuration: Groundtruth: [0.5 2.], GCC estimate: [1.71 1.17], GCC Error: [-1.21 0.82]

Neural GCC-PHAT

The following figure 6.49 will show you experiment done on different microphone configurations with 1 and 2 speakers at the same moment:

1 Speaker Figure 6.49 presents the results of GCC-PHAT experiments, which involve estimating the Cartesian coordinates (x, y) of sound sources in meters. These coordinates represent the positions of sound sources in a 2D plane. The results are presented for three different microphone configurations: In the binaural configuration, the ground truth coordinates of the sound source were [0.6, 1.25] meters. The GCC-PHAT algorithm estimated the coordinates as [1.36, 0] meters. This estimation resulted in an error of [-0.76, 1.25] meters. For the triaural configuration, the ground truth coordinates were [2.0, 1.0] meters. The GCC-PHAT algorithm estimated the coordinates as [1.39, 1.36] meters. The estimation error was [0.61, 0.36] meters. In the tetra-aural configuration, the ground truth coordinates of the sound source were [0.5, 2.0] meters. The GCC-PHAT algorithm estimated the coordinates as [1.71, 1.17] meters. This estimation resulted in an error of [-1.21, 0.82] meters. These results provide insights into the accuracy of GCC-PHAT in estimating the spatial positions of sound sources in different microphone configurations. In some cases, the estimated coordinates closely matched the ground truth, while in others, there were notable deviations.



(a) (b) (c)

Figure 6.49: NGCC-PHAT: position of [x,y] in cartesian axes and for different configuration in meters:
a) Binaural configuration: Groundtruth: [0.6 1.25], NGCC estimate: [1.29 0]
b) Triaural configuration: Groundtruth position: [2. 1.], NGCC estimate: [1.44461964 1.43318727]
c) Tetra-aural configuration: Groundtruth: [0.5 2.], GCC estimate: [1.31010009 1.41599011], GCC Error: [-0.81010009 0.58400989]

3.6 Results

3.7 Single Speaker

In this section, we present the results of our experiments focused on Direction of Arrival (DOA) estimation for a single speaker source. The table below (Table 6.2) summarizes the azimuthal errors in degrees obtained across various microphone configurations and algorithms.

Table 6.1: **Single Speaker**: DOA Estimation Error Results: Speaker azimuth for experiments in degrees

System	***Binaural***	***Triaural***	***Tetra-Aural***	***Circular Arr Mics***
TOPS	80.07	98.43	122.92	7
SRP-PHAT	49.07	116.56	15.07	6
WAVES	40.07	5.56	26.92	4
MUSIC	18.07	63.43	3.07	4
NormMUSIC	35.07	116.56	37.07	6
NGCC-PHAT	[0.6 0.4]	[0.6 0.3]	[-0.8 0.5]	-

Table 6.2: **Multiple Speakers (2 Speaker)**: DOA Estimation Error Results: Speaker azimuth for experiments in degrees

System	***Binaural***	***Triaural***	***Tetra-Aural***	***Circular Arr Mics***
TOPS	-	[45, 123]	[45, 0.]	[17, 2]
SRP-PHAT	[45, 0]	[45, 0]	[45, 0]	[43, 1]
WAVES	-	[196, 357]	[151, 87]	[4, 106]
MUSIC	[15, 60]	[136, 52]	[45, 0]	[2, 4]
NormMUSIC	[112, 67]	[45, 0]	[45, 0]	[3,1]
NGCC-PHAT	-	[-0.2, 0.01], [0.01, -0.01]	-	-
GCC-PHAT	-	[-0.20 0.00], [0.04 -0.03]	-	-

The results of the experiments conducted with various Sound Source Localization algorithms have shown that the microphone configuration that consistently performed the best across all algorithms is the one with a circular array of microphones.

In the case of the multispeaker experiment with NGCC-PHAT, promising results have also been obtained, and further exploration is underway to better understand the effectiveness of microphone configurations in different environments. This research aims to improve the accuracy of speaker localization in various scenarios.

4 On going works and Future Works

Currently, there are experiments involving fine-tuning and training of the NGCC-PHAT model using various microphone configurations considered in this study. The ongoing experiments with the NGCC-PHAT model are based on the model available in the author's public repository.

If the performance improves with individual training, it may be worth considering further investigation by adding other types of background noise that more closely

simulate real-world contexts for simulations. However, the NGCC-PHAT model is designed for the estimation of a single source. To use the model for multi-source estimation, a ReSepFormer has been applied to separate the recorded sources, assuming that they maintain phase properties once separated. This allows the application of the individual NGCC-PHAT on each separate track to localize the individual source.

Currently, we are in the process of creating simulations on the LibriSpeech dataset, using the same type of environment to test the algorithms on larger datasets that can provide a more representative sample. The aim is to create a new Sound Source Localization dataset based on LibriSpeech and pyroomacoustics, along with a new separation and localization method, the performance of which can be tested.

Once we have obtained the results of the tests of various algorithms on this dataset that is currently being created, the thesis will be updated as the completion of the work, and there is potential for publishing a contribution based on these findings.

Here (`https://github.com/valeriopuglisi/librispeech-ssl-datasets`) you can find the repository about this work with the code optimized to create LibriSpeech Sound Source Localization Dataset and preliminary results of DOA Algorithms Mean Error on Created Dataset. Actually the Dataset is just for single speaker but soon it will be extended as multispeaker Sound Source Localization Dataset with relative tests.

Chapter 7

Conclusions

This doctoral thesis represents an extensive exploration of deep learning-based audio analysis, a rapidly evolving field that holds immense potential for extracting and comprehending valuable insights from audio signals. Throughout this research, we embarked on a comprehensive journey, investigating various techniques, methodologies, and approaches in pursuit of a deeper understanding of audio data and its practical applications.

One of the fundamental lessons gleaned from this study is the paramount importance of conducting an exhaustive review of the state-of-the-art literature when tackling a new problem. Prior to diving into research, conducting an extensive literature review provides essential context, reveals existing methodologies, and highlights potential gaps in knowledge. This initial phase not only lays the foundation for informed research but also serves as a compass, guiding researchers toward the most promising avenues of exploration.

Moreover, the significance of reproducibility in scientific experimentation cannot be overstated. It is crucial that researchers rigorously document their methodologies, share their code, and make their experiments reproducible. The research community greatly benefits from transparent and reproducible experiments, as they serve as benchmarks for future investigations. It is through the dissemination of code and experimental details that the collective knowledge of the scientific community expands, enabling others to build upon existing work and contribute to the advancement of the field.

This work serves as a foundational stepping stone for future scientific discoveries. The "Deep Audio Analyzer" framework developed throughout this research stands as a testament to the advancements in audio analysis. This framework seamlessly integrates state-of-the-art models from Hugging Face with a simple click, democratizing access to cutting-edge technologies. Additionally, its unique capability to combine models for specific tasks across different datasets without the need for extensive coding makes it a valuable tool, particularly in forensic contexts.

In conclusion, this thesis marks a significant stride forward in the realm of deep learning-based audio analysis. It underscores the importance of beginning any research endeavour with a thorough review of the state-of-the-art literature, recognizing the invaluable insights and methodologies that have been developed by the scientific community. Furthermore, it highlights the indispensable role of reproducibility in

advancing scientific knowledge, emphasizing the need for open access to code and experimental details. As the journey continues, I eagerly anticipate future contributions and discoveries that will further enrich the field of audio analysis and deep learning. The "Deep Audio Analyzer" framework, with its ability to harness state-of-the-art models and simplify complex tasks, is poised to play a pivotal role in shaping the future of scientific exploration in audio analysis.

Appendix A

Deep Learning Models used in Deep Audio Analyzer

Deep Audio Analyzer integrates models developed through SpeechBrain and featured in the HugginFace platform.

1 Automatic Speech Recognition

1.1 Wav2Vec & Wav2Vec2

Wav2Vec

[236]

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

[14] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed and Michael Auli show for the first time that learning powerful representations from speech audio alone followed by fine-tuning on transcribed speech can outperform the best semi-supervised methods while being conceptually simpler. wav2vec 2.0 masks the speech input in the latent space and solves a contrastive task defined over a quantization of the latent representations which are jointly learned. Experiments using all labeled data of Librispeech achieve 1.8/3.3 WER on the clean/other test sets. When lowering the amount of labeled data to one hour, wav2vec 2.0 outperforms the previous state of the art on the 100 hour subset while using 100 times less labeled data. Using just ten minutes of labeled data and pre-training on 53k hours of unlabeled data still achieves 4.8/8.2 WER. This demonstrates the feasibility of speech recognition with limited amounts of labeled data.

- **wav2vec 2.0 with CTC/Attention trained on CommonVoice Italian (No LM)** This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on CommonVoice (Italian Language) within SpeechBrain.
  The performance of the model is the following:

  Release Test WER GPUs

  03-06-21 9.86 2xV100 32GB

This ASR system is composed of 2 different but linked blocks:

1- Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions (train.tsv) of CommonVoice (EN).

2- Acoustic model (wav2vec2.0 + CTC/Attention). A pretrained wav2vec 2.0 model (facebook/wav2vec2-large-it-voxpopuli) is combined with two DNN layers and finetuned on CommonVoice En. The obtained final acoustic representation is given to the CTC and attention decoders.

The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

- **wav2vec 2.0 with CTC/Attention trained on CommonVoice Kinyarwanda (No LM)** This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on CommonVoice (Kinyarwanda Language) within SpeechBrain.
  The performance of the model is the following:
  Release Test WER GPUs

  03-06-21 18.91 2xV100 32GB Pipeline description This ASR system is composed of 2 different but linked blocks:

  1 - Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions (train.tsv) of CommonVoice (RW).

  2 - Acoustic model (wav2vec2.0 + CTC/Attention). A pretrained wav2vec 2.0 model (wav2vec2-large-xlsr-53) is combined with two DNN layers and finetuned on CommonVoice En. The obtained final acoustic representation is given to the CTC and attention decoders.

- **wav2vec 2.0 with CTC/Attention trained on CommonVoice French (No LM)** This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on CommonVoice (French Language) within SpeechBrain.
  This ASR system is composed of 2 different but linked blocks: 1- Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions (train.tsv) of CommonVoice (FR).
  2 - Acoustic model (wav2vec2.0 + CTC). A pretrained wav2vec 2.0 model (LeBenchmark/wav2vec2-FR-7K-large) is combined with two DNN layers and finetuned on CommonVoice FR. The obtained final acoustic representation is given to the CTC greedy decoder.
  The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

- **wav2vec 2.0 with CTC trained on LibriSpeech** This ASR system is composed of 2 different but linked blocks:

  1 - Tokenizer (unigram) that transforms words into characters and trained with the train transcriptions (EN).

  2 - Acoustic model (wav2vec2.0 + CTC). A pretrained wav2vec 2.0 model (wav2vec2-large-960h-lv60-self) is combined with two DNN layers and finetuned on LibriSpeech. The obtained final acoustic representation is given to the CTC.

The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

- **wav2vec 2.0 with CTC trained on CommonVoice English (No LM)**

  This ASR system is composed of 2 different but linked blocks:

  1 - Tokenizer (unigram) that transforms words into characters and trained with the train transcriptions (EN).

  2 - Acoustic model (wav2vec2.0 + CTC). A pretrained wav2vec 2.0 model (wav2vec2-large-960h-lv60-self) is combined with two DNN layers and finetuned on LibriSpeech. The obtained final acoustic representation is given to the CTC. The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

1.2 CRDNN

- **CRDNN with CTC/Attention trained on CommonVoice 7.0 German (No LM)** This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on CommonVoice (German Language) within SpeechBrain.
  This ASR system is composed of 2 different but linked blocks:

  1 - Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions (train.tsv) of CommonVoice (DE).
  2 - Acoustic model (CRDNN + CTC/Attention). The CRDNN architecture is made of N blocks of convolutional neural networks with normalization and pooling on the frequency domain. Then, a bidirectional LSTM is connected to a final DNN to obtain the final acoustic representation that is given to the CTC and attention decoders.

  The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

- **CRDNN with CTC/Attention trained on CommonVoice French (No LM)** This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on CommonVoice (French Language) within SpeechBrain. This ASR system is composed of 2 different but linked blocks:

  1- Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions (train.tsv) of CommonVoice (FR).

  2 - Acoustic model (CRDNN + CTC/Attention). The CRDNN architecture is made of N blocks of convolutional neural networks with normalization and pooling on the frequency domain. Then, a bidirectional LSTM is connected to a final DNN to obtain the final acoustic representation that is given to the CTC and attention decoders.

- **CRDNN with CTC/Attention trained on CommonVoice Italian (No LM)** This repository provides all the necessary tools to perform automatic

speech recognition from an end-to-end system pretrained on CommonVoice (IT) within SpeechBrain.
The performance of the model is the following:

Release Test CER Test WER GPUs

07-03-21 5.40 16.61 2xV100 16GB

This ASR system is composed of 2 different but linked blocks:
1 - Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions (train.tsv) of CommonVoice (IT).
2 - Acoustic model (CRDNN + CTC/Attention). The CRDNN architecture is made of N blocks of convolutional neural networks with normalization and pooling on the frequency domain. Then, a bidirectional LSTM is connected to a final DNN to obtain the final acoustic representation that is given to the CTC and attention decoders.

The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

- **CRDNN with CTC/Attention and RNNLM trained on LibriSpeech** This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on LibriSpeech (EN) within SpeechBrain. For a better experience we encourage you to learn more about SpeechBrain. The performance of the model is the following:

  Release Test WER GPUs 20-05-22 3.09 1xV100 32GB

  This ASR system is composed with 3 different but linked blocks:

  1 - Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions of LibriSpeech.

  2- Neural language model (RNNLM) trained on the full 10M words dataset.

  3- Acoustic model (CRDNN + CTC/Attention). The CRDNN architecture is made of N blocks of convolutional neural networks with normalisation and pooling on the frequency domain. Then, a bidirectional LSTM is connected to a final DNN to obtain the final acoustic representation that is given to the CTC and attention decoders.
  The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

1.3 Conformer for KsponSpeech (with Transformer LM)

This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on KsponSpeech (Kr) within SpeechBrain.
This ASR system is composed of 3 different but linked blocks:

1 - Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions of KsponSpeech.
2 - Neural language model (Transformer LM) trained on the train transcriptions of

KsponSpeech
3 - Acoustic model made of a conformer encoder and a joint decoder with CTC + transformer. Hence, the decoding also incorporates the CTC probabilities.
The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling transcribe_file if needed.

1.4 Transformer for AISHELL (Mandarin Chinese)

This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on AISHELL (Mandarin Chinese) within SpeechBrain. For a better experience, we encourage you to learn more about SpeechBrain.
The performance of the model is the following: Release Dev CER Test CER GPUs Full Results 05-03-21 5.60 6.04 2xV100 32GB

https://drive.google.com/drive/folders/1zlTBib0XEwWeyhaXDXnkqtPsIBI18Uzs?usp=sharingG
Drive

This ASR system is composed of 2 different but linked blocks: 1 - Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions of LibriSpeech. 2 - Acoustic model made of a transformer encoder and a joint decoder with CTC + transformer. Hence, the decoding also incorporates the CTC probabilities.

1.5 Transformer for AISHELL + wav2vec2 (Mandarin Chinese)

This repository provides all the necessary tools to perform automatic speech recognition from an end-to-end system pretrained on AISHELL +wav2vec2 (Mandarin Chinese) within SpeechBrain. For a better experience, we encourage you to learn more about SpeechBrain. The performance of the model is the following: Release Dev CER Test CER GPUs Full Results 05-03-21 5.19 5.58 2xV100 32GB Google Drive

This ASR system is composed of 2 different but linked blocks: 1 - Tokenizer (unigram) that transforms words into subword units and trained with the train transcriptions of LibriSpeech. 2 - Acoustic model made of a wav2vec2 encoder and a joint decoder with CTC + transformer. Hence, the decoding also incorporates the CTC probabilities.

2 Emotion Recognition

2.1 Emotion Recognition with wav2vec2 base on IEMOCAP

This repository provides all the necessary tools to perform emotion recognition with a fine-tuned wav2vec2 (base) model using SpeechBrain. It is trained on IEMOCAP training data.
For a better experience, we encourage you to learn more about SpeechBrain. The model performance on IEMOCAP test set is:

Release Accuracy(%)

19-10-21 78.7 (Avg: 75.3)

This system is composed of an wav2vec2 model. It is a combination of convolutional and residual blocks[14]. The embeddings are extracted using attentive statistical pooling. The system is trained with Additive Margin Softmax Loss. Speaker Verification is performed using cosine distance between speaker embeddings. The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling classify_file if needed.

2.2 ECAPA-TDNN

Current speaker verification techniques rely on a neural network to extract speaker representations. The successful x-vector architecture is a Time Delay Neural Network (TDNN) that applies statistics pooling to project variable-length utterances into fixed-length speaker characterizing embeddings. In [68], the authors proposed multiple enhancements to this architecture based on recent trends in the related fields of face verification and computer vision. Firstly, the initial frame layers can be restructured into 1-dimensional Res2Net modules with impactful skip connections. Similarly to SE-ResNet, we introduce Squeeze-and-Excitation blocks in these modules to explicitly model channel interdependencies. The SE block expands the temporal context of the frame layer by rescaling the channels according to global properties of the recording. Secondly, neural networks are known to learn hierarchical features, with each layer operating on a different level of complexity. To leverage this complementary information, we aggregate and propagate features of different hierarchical levels. Finally, we improve the statistics pooling module with channel-dependent frame attention. This enables the network to focus on different subsets of frames during each of the channel's statistics estimation. The proposed ECAPA-TDNN architecture significantly outperforms state-of-the-art TDNN based systems on the VoxCeleb test sets and the 2019 VoxCeleb Speaker Recognition Challenge.

3 Speech Enhancement

3.1 MetricGAN+: An Improved Version of MetricGAN for Speech Enhancement

The discrepancy between the cost function used for training a speech enhancement model and human auditory perception usually makes the quality of enhanced speech unsatisfactory. Objective evaluation metrics which consider human perception can hence serve as a bridge to reduce the gap. In [80] MetricGAN was previously and designed to optimize objective metrics by connecting the metric with a discriminator. Because only the scores of the target evaluation functions are needed during training, the metrics can even be non-differentiable. In study [81] , researchers propose a MetricGAN+ in which three training techniques incorporating domain-knowledge of speech processing are proposed. With these techniques, experimental results on the VoiceBank-DEMAND dataset show that MetricGAN+ can increase PESQ score by 0.3 compared to the previous MetricGAN and achieve state-of-the-art results (PESQ score = 3.15).

3.2 SepFormer: Attention is All You Need in Speech Separation

[249] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, Jianyuan Zhong Recurrent Neural Networks (RNNs) have long been the dominant architecture in sequence-to-sequence learning. RNNs, however, are inherently sequential models that do not allow parallelization of their computations. Transformers are emerging as a natural alternative to standard RNNs, replacing recurrent computations with a multi-head attention mechanism. In this paper, we propose the SepFormer, a novel RNN-free Transformer-based neural network for speech separation. The SepFormer learns short and long-term dependencies with a multi-scale approach that employs transformers. The proposed model achieves state-of-the-art (SOTA) performance on the standard WSJ0-2/3mix [106] datasets. It reaches an SI-SNRi of 22.3 dB on WSJ0-2mix and an SI-SNRi of 19.5 dB on WSJ0-3mix. The SepFormer inherits the parallelization advantages of Transformers and achieves a competitive performance even when downsampling the encoded representation by a factor of 8. It is thus significantly faster and it is less memory-demanding than the latest speech separation systems with comparable performance.

- **SepFormer trained on WHAM!** : This repository provides all the necessary tools to perform audio source separation with a SepFormer model, implemented with SpeechBrain, and pretrained on WHAM! dataset, which is basically a version of WSJ0-Mix dataset with environmental noise. For a better experience we encourage you to learn more about SpeechBrain. The model performance is 16.3 dB SI-SNRi on the test set of WHAM! dataset.

- **SepFormer trained on WHAMR! (16k sampling frequency)**: This repository provides all the necessary tools to perform audio source separation with a SepFormer model, implemented with SpeechBrain, and pretrained on WHAMR! dataset with 16k sampling frequency, which is basically a version of WSJ0-Mix dataset with environmental noise and reverberation in 16k. The given model performance is 13.5 dB SI-SNRi on the test set of WHAMR! dataset.

- **SepFormer trained on WHAMR!(8k sampling frequency)** : This repository provides all the necessary tools to perform audio source separation with a SepFormer model, implemented with SpeechBrain, and pretrained on WHAMR! dataset, which is basically a version of WSJ0-Mix dataset with environmental noise and reverberation. For a better experience we encourage you to learn more about SpeechBrain. The model performance is 13.7 dB SI-SNRi on the test set of WHAMR! dataset.

- **SepFormer trained on WSJ0-2Mix** : This repository provides all the necessary tools to perform audio source separation with a SepFormer model, implemented with SpeechBrain, and pretrained on WSJ0-2Mix dataset. For a better experience we encourage you to learn more about SpeechBrain. The model performance is 22.4 dB on the test set of WSJ0-2Mix dataset.

- **SepFormer trained on WSJ0-3Mix** : This repository provides all the necessary tools to perform audio source separation with a SepFormer model, implemented with SpeechBrain, and pretrained on WSJ0-3Mix dataset. For a better experience we encourage you to learn more about SpeechBrain. The model performance is 19.8 dB SI-SNRi on the test set of WSJ0-3Mix dataset.

- **SepFormer trained on WHAM! for speech enhancement (8k sampling frequency)**: This repository provides all the necessary tools to perform speech enhancement (denoising) with a SepFormer model, implemented with SpeechBrain, and pretrained on WHAM! dataset with 8k sampling frequency, which is basically a version of WSJ0-Mix dataset with environmental noise and reverberation in 8k. For a better experience we encourage you to learn more about SpeechBrain. The given model performance is 14.35 dB SI-SNR on the test set of WHAMR! dataset.

- **SepFormer trained on WHAMR! for speech enhancement (8k sampling frequency)**: This repository provides all the necessary tools to perform speech enhancement (denoising + dereverberation) with a SepFormer model, implemented with SpeechBrain, and pretrained on WHAMR! dataset with 8k sampling frequency, which is basically a version of WSJ0-Mix dataset with environmental noise and reverberation in 8k. For a better experience we encourage you to learn more about SpeechBrain. The given model performance is 10.59 dB SI-SNR on the test set of WHAMR! dataset.

4 Voice Activity Detection

4.1 Voice Activity Detection with a (small) CRDNN model trained on Libriparty

This repository provides all the necessary tools to perform voice activity detection with SpeechBrain using a model pretrained on Libriparty. This system is composed of a CRDNN that outputs posteriors probabilities with a value close to one for speech frames and close to zero for non-speech segments. A threshold is applied on top of the posteriors to detect candidate speech boundaries.

Depending on the active options, these boundaries can be post-processed (e.g, merging close segments, removing short segments, etc) to further improve the performance. See more details below.

5 Speaker Verification

5.1 Speaker Verification with ECAPA-TDNN embeddings on Voxceleb

This repository provides all the necessary tools to perform speaker verification with a pretrained ECAPA-TDNN model using SpeechBrain. The system can be used to extract speaker embeddings as well. It is trained on Voxceleb 1+ Voxceleb2 training data.

For a better experience, we encourage you to learn more about SpeechBrain. The model performance on Voxceleb1-test set(Cleaned) is:

Release EER(%) minDCF

05-03-21 0.69 0.08258

This system is composed of an ECAPA-TDNN model. It is a combination of convolutional and residual blocks. The embeddings are extracted using attentive statistical pooling. The system is trained with Additive Margin Softmax Loss. Speaker

Verification is performed using cosine distance between speaker embeddings.

5.2 Speaker Verification with xvector embeddings on Voxceleb

This repository provides all the necessary tools to extract speaker embeddings with a pretrained TDNN model using SpeechBrain. The system is trained on Voxceleb 1+ Voxceleb2 training data.

For a better experience, we encourage you to learn more about SpeechBrain. The given model performance on Voxceleb1-test set (Cleaned) is:
Release EER(
05-03-21 3.2

This system is composed of a TDNN model coupled with statistical pooling. The system is trained with Categorical Cross-Entropy Loss.

6 Language Identification

- **Language Identification from Speech Recordings with ECAPA embeddings on CommonLanguage** This repository provides all the necessary tools to perform language identification from speeech recordinfs with SpeechBrain. The system uses a model pretrained on the CommonLanguage dataset (45 languages). You can download the dataset here The provided system can recognize the following 45 languages from short speech recordings:

  Arabic, Basque, Breton, Catalan, Chinese_China, Chinese_Hongkong, Chinese_Taiwan, Chuvash, Czech, Dhivehi, Dutch, English, Esperanto, Estonian, French, Frisian, Georgian, German, Greek, Hakha_Chin, Indonesian, Interlingua, Italian, Japanese, Kabyle, Kinyarwanda, Kyrgyz, Latvian, Maltese, Mangolian, Persian, Polish, Portuguese, Romanian, Romansh_Sursilvan, Russian, Sakha, Slovenian, Spanish, Swedish, Tamil, Tatar, Turkish, Ukranian, Welsh
  The given model performance on the test set is:

  Release Accuracy (%)

  30-06-21 85.0

  This system is composed of a ECAPA model coupled with statistical pooling. A classifier, trained with Categorical Cross-Entropy Loss, is applied on top of that.

  The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling classify_file if needed. Make sure your input tensor is compliant with the expected sampling rate if you use encode_batch and classify_batch.

- **VoxLingua107 ECAPA-TDNN Spoken Language Identification Model** This is a spoken language recognition model trained on the VoxLingua107 dataset using SpeechBrain. The model uses the ECAPA-TDNN architecture that has previously been used for speaker recognition. However, it uses more fully connected hidden layers after the embedding layer, and cross-entropy loss

was used for training. We observed that this improved the performance of extracted utterance embeddings for downstream tasks.
The system is trained with recordings sampled at 16kHz (single channel). The code will automatically normalize your audio (i.e., resampling + mono channel selection) when calling classify_file if needed.
The model can classify a speech utterance according to the language spoken. It covers 107 different languages (Abkhazian, Afrikaans, Amharic, Arabic, Assamese, Azerbaijani, Bashkir, Belarusian, Bulgarian, Bengali, Tibetan, Breton, Bosnian, Catalan, Cebuano, Czech, Welsh, Danish, German, Greek, English, Esperanto, Spanish, Estonian, Basque, Persian, Finnish, Faroese, French, Galician, Guarani, Gujarati, Manx, Hausa, Hawaiian, Hindi, Croatian, Haitian, Hungarian, Armenian, Interlingua, Indonesian, Icelandic, Italian, Hebrew, Japanese, Javanese, Georgian, Kazakh, Central Khmer, Kannada, Korean, Latin, Luxembourgish, Lingala, Lao, Lithuanian, Latvian, Malagasy, Maori, Macedonian, Malayalam, Mongolian, Marathi, Malay, Maltese, Burmese, Nepali, Dutch, Norwegian Nynorsk, Norwegian, Occitan, Panjabi, Polish, Pushto, Portuguese, Romanian, Russian, Sanskrit, Scots, Sindhi, Sinhala, Slovak, Slovenian, Shona, Somali, Albanian, Serbian, Sundanese, Swedish, Swahili, Tamil, Telugu, Tajik, Thai, Turkmen, Tagalog, Turkish, Tatar, Ukrainian, Urdu, Uzbek, Vietnamese, Waray, Yiddish, Yoruba, Mandarin Chinese).

Appendix B

Projects outside the research field

1 S.A.T.U.R.N

The Saturn project, which overall aims at improving the productivity of semiconductor production processes, includes three main lines of activity, articulated in 14 Realization Objectives (OR). The Department of Mathematics and Computer Science was entrusted with the OR 13 - Technology and applications of artificial intelligence as a support to the production system. The final goal of this OR is the definition and validation of algorithmic solutions for scheduling optimization. Currently the workers in the production department are divided into three shifts (6-14; 14-22; 22-6). They stick to a dispatching document, which shows the queue of batches to be processed on specific machines. Dispatching gets this information from a simulator (seen as a black box), started three times a day, at the start of each shift. In turn, the simulator receives this information from various sources.

1.1 Input

- Photography of the state of all lots in the machines;
- Static rules are inserted by industry experts, which allow to maintain the integrity of the production flow (which batches can be processed in which machines).

The simulator formats the data contained in various files with the ".csv" extension.

The union of these files provides the list of the various products that must be processed, in particular, there is the list of the various operations that concern them with the relative events to be carried out. Also, you are aware of the qualified machines that can carry out those particular events.

1.2 Output

- Provides choices (not strictly used in all areas) on which batches must be processed in which machine based on a maximization of the handled;
- These choices can be bypassed, using a priority field by the experts to allow one or more batches to be processed before others, (for example to compensate for customer changes).

Handled refers to the number of batches that have been processed during a given shift. Currently, maximizing the amount of traffic implies first choosing fast operations

at the expense of those that require more time. This creates "bubbles" within the production process, that is, some areas will be blocked, with no batches to process, while other areas will be overloaded. The department head being assessed for busy cannot afford to remain stationary for an entire shift, so he explicitly requests lots from other areas in order to create a busy one. All these choices cause cascading changes in all other areas, making the whole system chaotic and unpredictable.

1.3 Activities

RI 13.1 Comparison and setup of scheduling algorithms The objective of this activity is the study and validation of previous works regarding the simulation of manufacturing processes of semiconductor components and in particular on the definition of dispatching optimization algorithms, with the aim of evaluating a based development that takes into account results and limitations of previous attempts. The production process must take into account a large number of variables and parameters that can change relatively quickly.

In a first phase, the study will focus on more general methods for scheduling management, starting from operative research with CP-SAT problem model and moving on other types of approaches like neural networks and specifically the Reinforcement Learning (RL) paradigm. In a second phase, more specific solutions relating to the scheduling of chip production will be investigated and changes will be proposed to adapt to the problem faced. These include work in [Park, J. Huh, J. Kim and J. Park, "A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities," in IEEE Transactions on Automation Science and Engineering. doi: 10.1109 / TASE.2019.2956762] in which a decentralized model is used to agents learning through RL or working in [H. Kim, D. Lim and S. Lee, "Deep Learning-Based Dynamic Scheduling for Semiconductor Manufacturing With High Uncertainty of Automated Material Handling System Capability," in IEEE Transactions on Semiconductor Manufacturing, vol. 33, no. 1, pp. 13-22, Feb. 2020. doi: 10.1109 / TSM.2020.2965293] in which the different production conditions are modeled and through deep-learning the allocation of resources that maximizes production.

RI 13.2 Profiling of the FAB Given the operating complexity of the FAB, and its specificity, this activity has the objective of profiling the FAB present in the STMicroelectronics headquarters in Catania, which is necessary for the development of a scheduling and dispatching system optimized for the specific FAB. For this purpose, the members of the research team will hold training and discussion meetings with the specialized staff at the STM headquarters in Catania. During these meetings, the different process phases necessary for the production of the different technologies will be explained in detail. It will also show the WIP (Work In Progress) models used in the FAB, the formal representation of the scheduling and processing phases, as well as the specific terminology of the sector. In this activity, the production targets will be presented in quantitative terms. This formalization is essential for the mathematical definition of optimization methods of smart scheduling processes.

RI 13.3 Prototype definition Based on the results of the previous activities, some algorithmic solutions for scheduling optimization will be defined during this phase. Subsequently, they will proceed with the benchmarking of these solutions, using real dispatching data provided by STM, and a software that simulates in detail

a FAB production environment. In particular, FAB-dispatching interaction systems will be tested (in a virtual environment) with the ability to extract timely reports on performance according to the choices made by the scheduling algorithm.

Catania, November 30, 2023

Dott. Valerio Francesco Puglisi

Tutor: Prof. S. Battiato

Bibliography

[1] U.s. v. mckeever.

[2] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[3] Adavanne, S., Politis, A., Nikunen, J., and Virtanen, T. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing 13*, 1 (2018), 34–48.

[4] Adavanne, S., Politis, A., and Virtanen, T. Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network. In *2018 26th European Signal Processing Conference (EUSIPCO)* (2018), IEEE, pp. 1462–1466.

[5] Adavanne, S., Politis, A., and Virtanen, T. Localization, detection and tracking of multiple moving sound sources with a convolutional recurrent neural network. *arXiv preprint arXiv:1904.12769* (2019).

[6] Adavanne, S., Politis, A., and Virtanen, T. Differentiable tracking-based training of deep learning sound source localizers. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* (2021), IEEE, pp. 211–215.

[7] Allen, J. B., Jeng, P. S., and Levitt, H. Evaluation of human middle ear function via an acoustic power assessment. *Journal of Rehabilitation Research & Development 42* (2005).

[8] Allen, J. B., and Rabiner, L. R. A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE 65*, 11 (1977), 1558–1564.

[9] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning* (2016), PMLR, pp. 173–182.

[10] Arberet, S., Gribonval, R., and Bimbot, F. A robust method to count and locate audio sources in a multichannel underdetermined mixture. *IEEE Transactions on Signal Processing 58*, 1 (2009), 121–133.

[11] Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., and Weber,

G. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670* (2019).

[12] Argentieri, S., Danes, P., and Souères, P. A survey on sound source localization in robotics: From binaural to array processing methods. *Computer Speech & Language 34*, 1 (2015), 87–112.

[13] Assael, Y. M., Shillingford, B., Whiteson, S., and De Freitas, N. Lipnet: End-to-end sentence-level lipreading. *arXiv preprint arXiv:1611.01599* (2016).

[14] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems 33* (2020), 12449–12460.

[15] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., and Bengio, Y. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2016), IEEE, pp. 4945–4949.

[16] Bai, S., Kolter, J. Z., and Koltun, V. Trellis networks for sequence modeling. *arXiv preprint arXiv:1810.06682* (2018).

[17] Bai, Z., and Zhang, X.-L. Speaker recognition based on deep learning: An overview. *Neural Networks 140* (2021), 65–99.

[18] Basten, T., de Bree, H., and Sadasivan, S. Acoustic eyes: A novel sound source localization and monitoring technique with 3d sound probes.

[19] Battiato, S., Giudice, O., and Paratore, A. Multimedia forensics: discovering the history of multimedia contents. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016* (2016), pp. 5–16.

[20] Benesty, J., Chen, J., and Huang, Y. *Microphone array signal processing*, vol. 1. Springer Science & Business Media, 2008.

[21] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence 35*, 8 (2013), 1798–1828.

[22] Bess, F. H., and Humes, L. Audiology: the fundamentals. *(No Title)* (1995).

[23] Bianchi, L., Antonacci, F., Sarti, A., and Tubaro, S. The ray space transform: A new framework for wave field processing. *IEEE Transactions on Signal Processing 64*, 21 (2016), 5696–5706.

[24] Bianco, M. J., Gannot, S., Fernandez-Grande, E., and Gerstoft, P. Semi-supervised source localization in reverberant environments with deep generative modeling. *IEEE Access 9* (2021), 84956–84970.

[25] Bianco, M. J., Gannot, S., and Gerstoft, P. Semi-supervised source localization with deep generative modeling. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)* (2020), IEEE, pp. 1–6.

[26] Bohlender, A., Spriet, A., Tirry, W., and Madhu, N. Exploiting temporal context in cnn based multisource doa estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 29* (2021), 1594–1608.

[27] Boigne, J., Liyanage, B., and Östrem, T. Recognizing more emotions with less data using self-supervised transfer learning. *arXiv preprint arXiv:2011.05585* (2020).

[28] Bologni, G., Heusdens, R., and Martinez, J. Acoustic reflectors localization from stereo recordings using neural networks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 1–5.

[29] Bolt, R. H., Cooper, F. S., David Jr, E. E., Denes, P. B., Pickett, J. M., and Stevens, K. N. Identification of a speaker by speech spectrograms: How do scientists view its reliability for use as legal evidence? *Science 166*, 3903 (1969), 338–343.

[30] Bolt, R. H., Cooper, F. S., David Jr, E. E., Denes, P. B., Pickett, J. M., and Stevens, K. N. Speaker identification by speech spectrograms: a scientists' view of its reliability for legal purposes. *The Journal of the Acoustical Society of America 47*, 2B (1970), 597–612.

[31] Bolt, R. H., Cooper, F. S., David Jr, E. E., Denes, P. B., Pickett, J. M., and Stevens, K. N. Speaker identification by speech spectrograms: some further observations. *The Journal of the Acoustical Society of America 54*, 2 (1973), 531–534.

[32] Bolt, R. H., Cooper, F. S., David Jr, E. E., Denes, P. B., Pickett, J. M., and Stevens, K. N. *On the Theory and Practice of Voice Identification*. The National Academies Press, Washington, DC, 1979.

[33] Brown, J. C. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America 89*, 1 (1991), 425–434.

[34] Bu, H., Du, J., Na, X., Wu, B., and Zheng, H. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)* (2017), IEEE, pp. 1–5.

[35] Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation 42* (2008), 335–359.

[36] Cao, Y., Iqbal, T., Kong, Q., An, F., Wang, W., and Plumbley, M. D. An improved event-independent network for polyphonic sound event localization and detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 885–889.

[37] Cao, Y., Kong, Q., Iqbal, T., An, F., Wang, W., and Plumbley, M. D. Polyphonic sound event detection and localization using a two-stage strategy. *arXiv preprint arXiv:1905.00268* (2019).

[38] Cerna, M., and Harvey, A. F. The fundamentals of fft-based signal analysis and measurement. Tech. rep., Application Note 041, National Instruments, 2000.

[39] Chakrabarty, S., and Habets, E. A. Broadband doa estimation using convolutional neural networks trained with noise signals. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* (2017), IEEE, pp. 136–140.

[40] Chakrabarty, S., and Habets, E. A. Multi-speaker localization using convolutional neural network trained with noise. *arXiv preprint arXiv:1712.04276* (2017).

[41] Chakrabarty, S., and Habets, E. A. Multi-speaker doa estimation using deep convolutional networks trained with noise signals. *IEEE Journal of Selected Topics in Signal Processing 13*, 1 (2019), 8–21.

[42] Chan, W., Jaitly, N., Le, Q., and Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016), pp. 4960–4964.

[43] Chang, S.-Y., Li, B., Simko, G., Sainath, T. N., Tripathi, A., van den Oord, A., and Vinyals, O. Temporal modeling using dilated convolution and gating for voice-activity-detection. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2018), IEEE, pp. 5549–5553.

[44] Chazan, S. E., Hammer, H., Hazan, G., Goldberger, J., and Gannot, S. Multi-microphone speaker separation based on deep doa estimation. In *2019 27th European Signal Processing Conference (EUSIPCO)* (2019), IEEE, pp. 1–5.

[45] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[46] Chollet, F. *Deep learning with Python*. Simon and Schuster, 2021.

[47] Chorowski, J., and Jaitly, N. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695* (2016).

[48] Chung, J. S., Nagrani, A., and Zisserman, A. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622* (2018).

[49] Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. Lip reading sentences in the wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 3444–3453.

[50] Chytas, S. P., and Potamianos, G. Hierarchical detection of sound events and their localization using convolutional neural networks with adaptive thresholds.

[51] Cohen, T., Weiler, M., Kicanaoglu, B., and Welling, M. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning* (2019), PMLR, pp. 1321–1330.

[52] Comanducci, L., Borra, F., Bestagini, P., Antonacci, F., Tubaro, S., and Sarti, A. Source localization using distributed microphones in reverberant environments based on deep learning and ray space transform. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 28* (2020), 2238–2251.

[53] Comminiello, D., Lella, M., Scardapane, S., and Uncini, A. Quaternion convolutional neural networks for detection and localization of 3d sound events. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 8533–8537.

[54] Community, H. F. The ai community building the future. build, train and deploy state of the art models powered by the reference open source in machine learning., 2021.

[55] Conrad, E., Misenar, S., and Feldman, J. Chapter 1 - domain 1: Access control. In *Eleventh Hour CISSP (Second Edition)*, E. Conrad, S. Misenar, and J. Feldman, Eds., second edition ed. Syngress, Boston, 2014, pp. 1–21.

[56] Cordourier, H., Lopez Meyer, P., Huang, J., Del Hoyo Ontiveros, J., and Lu, H. Gcc-phat cross-correlation audio features for simultaneous sound event localization and detection (seld) on multiple rooms.

[57] Cosentino, J., Pariente, M., Cornell, S., Deleforge, A., and Vincent, E. Librimix: An open-source dataset for generalizable speech separation, 2020.

[58] Dahl, G. E., Yu, D., Deng, L., and Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing 20*, 1 (2012), 30–42.

[59] Daniel, J., and Kitić, S. Time domain velocity vector for retracing the multipath propagation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 421–425.

[60] Deleforge, A., Forbes, F., and Horaud, R. Variational em for binaural sound-source separation and localization. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), IEEE, pp. 76–80.

[61] Deleforge, A., and Horaud, R. 2d sound-source localization on the binaural manifold. In *2012 IEEE International Workshop on Machine Learning for Signal Processing* (2012), IEEE, pp. 1–6.

[62] Deleforge, A., Horaud, R., Schechner, Y. Y., and Girin, L. Co-localization of audio sources in images using binaural features and locally-linear regression. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 23*, 4 (2015), 718–731.

[63] Deng, K., Yang, Z., Watanabe, S., Higuchi, Y., Cheng, G., and Zhang, P. Improving non-autoregressive end-to-end speech recognition with pre-trained acoustic and language models. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), IEEE, pp. 8522–8526.

[64] Deng, L., Li, J., Huang, J. T., Yao, K., Yu, D., Seide, F., Seltzer, M. L., Zweig, G., He, X., Williams, J., Gong, Y., and Acero, A. Recent advances in deep learning for speech research at microsoft. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), 8604–8608.

[65] Deng, L., and Li, X. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing 21*, 5 (2013), 1060–1089.

[66] Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed, A.-r., and Hinton, G. Binary coding of speech spectrograms using a deep auto-encoder. In *Interspeech 2010* (September 2010), International Speech Communication Association.

[67] Deng, L., and Yu, D. Deep learning: Methods and applications. Tech. Rep. MSR-TR-2014-21, Microsoft, May 2014.

[68] Desplanques, B., Thienpondt, J., and Demuynck, K. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143* (2020).

[69] Detlefsen, N. S., Borovec, J., Schock, J., Jha, A. H., Koker, T., Di Liello, L., Stancl, D., Quan, C., Grechkin, M., and Falcon, W. Torchmetrics-measuring reproducibility in pytorch. *Journal of Open Source Software 7*, 70 (2022), 4101.

[70] Di Claudio, E. D., and Parisi, R. Waves: Weighted average of signal subspaces for robust wideband direction finding. *IEEE Transactions on Signal Processing 49*, 10 (2001), 2179–2191.

[71] Diaz-Guerra, D., Miguel, A., and Beltran, J. R. Robust sound source tracking using srp-phat and 3d convolutional neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 29* (2020), 300–311.

[72] DiBiase, J. H., Silverman, H. F., and Brandstein, M. S. Robust localization in reverberant rooms. In *Microphone arrays: signal processing techniques and applications*. Springer, 2001, pp. 157–180.

[73] Dmochowski, J. P., Benesty, J., and Affes, S. Broadband music: Opportunities and challenges for multiple source localization. In *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (2007), IEEE, pp. 18–21.

[74] Dorfan, Y., and Gannot, S. Tree-based recursive expectation-maximization algorithm for localization of acoustic sources. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 23*, 10 (2015), 1692–1703.

[75] Dwivedi, D., Ganguly, A., and Haragopal, V. Contrast between simple and complex classification algorithms. In *Statistical Modeling in Machine Learning*. Elsevier, 2023, pp. 93–110.

[76] Ellis, D. Chroma feature analysis and synthesis. *Resources of laboratory for the recognition and organization of speech and Audio-LabROSA 5* (2007).

[77] Emmanuel, P., Parrish, N., and Horton, M. Multi-scale network for sound event localization and detection. *Tech. report of DCASE Challenge* (2021).

[78] Ewert, S. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Proc. ISMIR* (2011).

[79] Fu, Q. *Beyond Audio Quality: Understanding and Improving Voice Communication With Low-Resource Deep Learning*. PhD thesis, Vanderbilt University, 2023.

[80] Fu, S.-W., Liao, C.-F., Tsao, Y., and Lin, S.-D. Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement. In *International Conference on Machine Learning* (2019), PMLR, pp. 2031–2041.

[81] Fu, S.-W., Yu, C., Hsieh, T.-A., Plantinga, P., Ravanelli, M., Lu, X., and Tsao, Y. Metricgan+: An improved version of metricgan for speech enhancement. *arXiv preprint arXiv:2104.03538* (2021).

[82] Furui, S. Chapter 7 - speaker recognition in smart environments. In *Human-Centric Interfaces for Ambient Intelligence*, H. Aghajan, R. L.-C. Delgado, and J. C. Augusto, Eds. Academic Press, Oxford, 2010, pp. 163–184.

[83] Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., and Pallett, D. S. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n 93* (1993), 27403.

[84] Geisler, C. D. *From sound to synapse: physiology of the mammalian ear*. Oxford University Press, USA, 1998.

[85] Giudice, O., Guarnera, L., Paratore, A. B., Farinella, G. M., and Battiato, S. Siamese ballistics neural network. In *2019 IEEE International Conference on Image Processing (ICIP)* (2019), IEEE, pp. 4045–4049.

[86] Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

[87] Grandini, M., Bagli, E., and Visani, G. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756* (2020).

[88] Graves, A., and Jaitly, N. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning* (2014), PMLR, pp. 1764–1772.

[89] Grondin, F., Glass, J., Sobieraj, I., and Plumbley, M. D. Sound event localization and detection using crnn on pairs of microphones. *arXiv preprint arXiv:1910.10049* (2019).

[90] Grumiaux, P.-A., Kitić, S., Girin, L., and Guérin, A. Improved feature extraction for crnn-based multiple sound source localization. In *2021 29th European Signal Processing Conference (EUSIPCO)* (2021), IEEE, pp. 231–235.

[91] Grumiaux, P.-A., Kitić, S., Girin, L., and Guérin, A. A survey of sound source localization with deep learning methods. *The Journal of the Acoustical Society of America 152*, 1 (2022), 107–151.

[92] Grumiaux, P.-A., Kitić, S., Srivastava, P., Girin, L., and Guérin, A. Saladnet: Self-attentive multisource localization in the ambisonics domain. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* (2021), IEEE, pp. 336–340.

[93] Guarnera, F., Allegra, D., Giudice, O., Stanco, F., and Battiato, S. A new study on wood fibers textures: documents authentication through lbp fingerprint. In *2019 IEEE International Conference on Image Processing (ICIP)* (2019), IEEE, pp. 4594–4598.

[94] Guirguis, K., Schorn, C., Guntoro, A., Abdulatif, S., and Yang, B. Seld-tcn: Sound event localization & detection via temporal convolutional networks. In *2020 28th European Signal Processing Conference (EUSIPCO)* (2021), IEEE, pp. 16–20.

[95] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100* (2020).

[96] Hammer, H., Chazan, S. E., Goldberger, J., and Gannot, S. Dynamically localizing multiple speakers based on the time-frequency domain. *EURASIP Journal on Audio, Speech, and Music Processing 2021*, 1 (2021), 16.

[97] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567* (2014).

[98] Hao, Y., Küçük, A., Ganguly, A., and Panahi, I. M. Spectral flux-based convolutional neural network architecture for speech source localization and its real-time implementation. *IEEE Access 8* (2020), 197047–197058.

[99] Harris, C. M. Absorption of sound in air versus humidity and temperature. *The Journal of the Acoustical Society of America 40*, 1 (1966), 148–159.

[100] Hartmann, W. M. *Principles of musical acoustics*. Springer, 2013.

[101] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[102] He, W. Joint localization and classification of multiple sound sources using a multi-task neural network, he, weipeng, motlicek, petr and odobez, jean-marc, idiap-rr-17-2018.

[103] He, W., Motlicek, P., and Odobez, J.-M. Deep neural networks for multiple speaker detection and localization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (2018), IEEE, pp. 74–79.

[104] He, W., Motlicek, P., and Odobez, J.-M. Neural network adaptation and data augmentation for multi-speaker direction-of-arrival estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 29* (2021), 1303–1317.

[105] He, Y., Trigoni, N., and Markham, A. Sounddet: Polyphonic moving sound event detection and localization from raw waveform. In *International Conference on Machine Learning* (2021), PMLR, pp. 4160–4170.

[106] Hershey, J. R., Chen, Z., Le Roux, J., and Watanabe, S. Deep clustering: Discriminative embeddings for segmentation and separation. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016).

[107] Hickling, R., Wei, W., and Raspet, R. Finding the direction of a sound source using a vector sound-intensity probe. *The Journal of the Acoustical Society of America 94*, 4 (1993), 2408–2412.

[108] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine 29*, 6 (2012), 82–97.

[109] Hirvonen, T. Classification of spatial audio location and content using convolutional neural networks. In *Audio Engineering Society Convention 138* (2015), Audio Engineering Society.

[110] Hogg, A. O., Neo, V. W., Weiss, S., Evers, C., and Naylor, P. A. A polynomial eigenvalue decomposition music approach for broadband sound source localization. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* (2021), IEEE, pp. 326–330.

[111] Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 7132–7141.

[112] Huang, D., and Perez, R. F. Sseldnet: a fully end-to-end sample-level framework for sound event localization and detection. *DCASE* (2021).

[113] Huang, Y., Wu, X., and Qu, T. A time-domain unsupervised learning based sound source localization method. In *2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP)* (2020), IEEE, pp. 26–32.

[114] Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., and Rudnicky, A. I. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE international conference on acoustics speech and signal processing proceedings* (2006), vol. 1, IEEE, pp. I–I.

[115] Jarrett, D. P., Habets, E. A., and Naylor, P. A. 3d source localization in the spherical harmonic domain using a pseudointensity vector. In *2010 18th European Signal Processing Conference* (2010), IEEE, pp. 442–446.

[116] Jenrungrot, T., Jayaram, V., Seitz, S., and Kemelmacher-Shlizerman, I. The cone of silence: Speech separation by localization. *Advances in Neural Information Processing Systems 33* (2020), 20925–20938.

[117] Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., and Cai, L.-H. Music type classification by spectral contrast feature. In *Proceedings. IEEE international conference on multimedia and expo* (2002), vol. 1, IEEE, pp. 113–116.

[118] Jurafsky, D., and Martin, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st ed. Prentice Hall PTR, USA, 2000.

[119] Kapka, S., and Lewandowski, M. Sound source detection, localization and classification using consecutive ensemble of crnn models. *arXiv preprint arXiv:1908.00766* (2019).

[120] Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., and Alhussain, T. Speech emotion recognition using deep learning techniques: A review. *IEEE Access 7* (2019), 117327–117345.

[121] Kim, J., and Hahn, M. Voice activity detection using an adaptive context attention model. *IEEE Signal Processing Letters 25*, 8 (2018), 1181–1185.

[122] Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[123] Kim, Y., and Ling, H. Direction of arrival estimation of humans with a small sensor array using an artificial neural network. *Progress In Electromagnetics Research B 27* (2011), 127–149.

[124] Kingma, D. P., and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[125] Kinsler, L. E., Frey, A. R., Coppens, A. B., and Sanders, J. V. *Fundamentals of acoustics*. John wiley & sons, 2000.

[126] Klapuri, A., and Davy, M. Signal processing methods for music transcription.

[127] Knapp, C., and Carter, G. The generalized correlation method for estimation of time delay. *IEEE transactions on acoustics, speech, and signal processing 24*, 4 (1976), 320–327.

[128] Koenig, B. E. Spectrographic voice identification: A forensic survey. *The Journal of the Acoustical Society of America 79*, 6 (1986), 2088–2090.

[129] Koenig, B. E. Enhancement of forensic audio recordings. *Journal of the Audio Engineering Society 36*, 11 (1988), 884–894.

[130] Komatsu, T., Togami, M., and Takahashi, T. Sound event localization and detection using convolutional recurrent neural networks and gated linear units. In *2020 28th European Signal Processing Conference (EUSIPCO)* (2021), IEEE, pp. 41–45.

[131] Kong, Q., Cao, Y., Iqbal, T., Xu, Y., Wang, W., and Plumbley, M. D. Cross-task learning for audio tagging, sound event detection and spatial localization: Dcase 2019 baseline systems. *arXiv preprint arXiv:1904.03476* (2019).

[132] Krause, D., and Kowalczyk, K. Arborescent neural network architectures for sound event detection and localization. *Detection Classification Acoust. Scenes Events Challenge, Tech. Rep* (2019).

[133] Krause, D., Politis, A., and Kowalczyk, K. Comparison of convolution types in cnn-based feature extraction for sound source localization. In *2020 28th European Signal Processing Conference (EUSIPCO)* (2021), IEEE, pp. 820–824.

[134] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25* (2012).

[135] Kuchaiev, O., Li, J., Nguyen, H., Hrinchuk, O., Leary, R., Ginsburg, B., Kriman, S., Beliaev, S., Lavrukhin, V., Cook, J., et al. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577* (2019).

[136] Kujawski, A., Herold, G., and Sarradj, E. A deep learning method for grid-free localization and quantification of sound sources. *The Journal of the Acoustical Society of America 146*, 3 (2019), EL225–EL231.

[137] Landschoot, C. R., and Xiang, N. Model-based bayesian direction of arrival analysis for sound sources using a spherical microphone array. *The Journal of the Acoustical Society of America 146*, 6 (2019), 4936–4946.

[138] Le Moing, G., Vinayavekhin, P., Agravante, D. J., Inoue, T., Vongkulbhisal, J., Munawar, A., and Tachibana, R. Data-efficient framework for real-world multiple sound source 2d localization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 3425–3429.

[139] Le Moing, G., Vinayavekhin, P., Inoue, T., Vongkulbhisal, J., Munawar, A., Tachibana, R., and Agravante, D. J. Learning multiple sound source 2d localization. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)* (2019), IEEE, pp. 1–6.

[140] Le Roux, J., Wisdom, S., Erdogan, H., and Hershey, J. R. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 626–630.

[141] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 156–165.

[142] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature 521*, 7553 (2015), 436–444.

[143] Lee, J., Park, J., Kim, K. L., and Nam, J. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789* (2017).

[144] Lee, S.-H., Hwang, J.-W., Seo, S.-B., and Park, H.-M. Sound event localization and detection using cross-modal attention and parameter sharing for dcase2021 challenge. *DCASE2021 Challenge, Tech. Rep.* (2021).

[145] Leung, S., and Ren, Y. Spectrum combination and convolutional recurrent neural networks for joint localization and detection of sound events. *Detection Classification Acoust. Scenes Events Challenge, Tech. Rep* (2019).

[146] Li, Q., Zhang, X., and Li, H. Online direction of arrival estimation based on deep learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 2616–2620.

[147] Li, X., Girin, L., Horaud, R., and Gannot, S. Multiple-speaker localization based on direct-path features and likelihood maximization with spatial sparsity regularization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 25*, 10 (2017), 1997–2012.

[148] Li, X., Horaud, R., Girin, L., and Gannot, S. Voice activity detection based on statistical likelihood ratio with adaptive thresholding. In *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)* (2016), IEEE, pp. 1–5.

[149] Library, N. P., and (2015), M. History of the white house tapes.

[150] Lin, Y., and Wang, Z. A report on sound event localization and detection. *Detection Classification Acoust. Scenes Events Challenge, Tech. Rep* (2019).

[151] Liu, Y., Sun, G., Qiu, Y., Zhang, L., Chhatkuli, A., and Van Gool, L. Transformer in convolutional neural networks. *arXiv preprint arXiv:2106.03180* (2021).

[152] Lu, Z. Sound event detection and localization based on cnn and lstm. *Detection Classification Acoust. Scenes Events Challenge, Tech. Rep* (2019).

[153] Luo, Y., Chen, Z., and Yoshioka, T. Dual-path rnn: Efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), pp. 46–50.

[154] Luo, Y., and Mesgarani, N. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 27*, 8 (2019), 1256–1266.

[155] M., T., Adeel, A., and Hussain, A. A survey on techniques for enhancing speech. *International Journal of Computer Applications 179* (02 2018), 1–14.

[156] Ma, N., Brown, G., and May, T. Exploiting deep neural networks and head movements for binaural localisation of multiple speakers in reverberant conditions. In *Interspeech* (2015), vol. 2015, International Speech Communication Association, pp. 160–164.

[157] Ma, W., and Liu, X. Phased microphone array for sound source localization with deep learning. *Aerospace Systems 2*, 2 (2019), 71–81.

[158] Maas, A., Le, Q. V., O'neil, T. M., Vinyals, O., Nguyen, P., and Ng, A. Y. Recurrent neural networks for noise reduction in robust asr.

[159] Mabande, E., Sun, H., Kowalczyk, K., and Kellermann, W. Comparison of subspace-based and steered beamformer-based reflection localization methods. In *2011 19th European Signal Processing Conference* (2011), IEEE, pp. 146–150.

[160] Maciejewski, M., Wichern, G., McQuinn, E., and Roux, J. L. Whamr!: Noisy and reverberant single-channel speech separation.

[161] Mack, W., Bharadwaj, U., Chakrabarty, S., and Habets, E. A. Signal-aware broadband doa estimation using attention mechanisms. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 4930–4934.

[162] Maher, R. C. Audio forensic examination. *IEEE Signal Processing Magazine 26*, 2 (2009), 84–94.

[163] Maher, R. C. Overview of audio forensics. In *Intelligent multimedia analysis for security applications*. Springer, 2010, pp. 127–144.

[164] Maher, R. C. *Principles of forensic audio analysis*, vol. 34. Springer, 2018.

[165] Mandel, M. I., Weiss, R. J., and Ellis, D. P. Model-based expectation-maximization source separation and localization. *IEEE Transactions on Audio, Speech, and Language Processing 18*, 2 (2009), 382–394.

[166] May, T., Van De Par, S., and Kohlrausch, A. A probabilistic model for robust localization based on a binaural auditory front-end. *IEEE Transactions on audio, speech, and language processing 19*, 1 (2010), 1–13.

[167] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (2015), vol. 8, pp. 18–25.

[168] Mehrish, A., Majumder, N., Bharadwaj, R., Mihalcea, R., and Poria, S. A review of deep learning techniques for speech processing. *Information Fusion* (2023), 101869.

[169] Mermelstein, P. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence 116* (1976), 374–388.

[170] Metz, C. E. Basic principles of roc analysis. *Seminars in nuclear medicine 8 4* (1978), 283–98.

[171] Miller, F. P., Vandome, A. F., and McBrewster, J. *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press, 2009.

[172] Mirsamadi, S., Barsoum, E., and Zhang, C. Automatic speech emotion recognition using recurrent neural networks with local attention. In *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)* (2017), IEEE, pp. 2227–2231.

[173] Moore, B. C. *An introduction to the psychology of hearing*. Brill, 2012.

[174] Morris, A., Maier, V., and Green, P. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition.

[175] Morris, A. C., Maier, V., and Green, P. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. In *Eighth International Conference on Spoken Language Processing* (2004).

[176] Müller, M. Short-time fourier transform and chroma features. *Lab Course, Friedrich-Alexander-Universität Erlangen-Nürnberg* (2015).

[177] Müller, M., and Zalkow, F. Fmp notebooks: Educational material for teaching and learning fundamentals of music processing. In *ISMIR* (2019), pp. 573–580.

[178] Naranjo-Alcazar, J., Perez-Castanos, S., Ferrandis, J., Zuccarello, P., and Cobos, M. Sound event localization and detection using squeeze-excitation residual cnns. *arXiv preprint arXiv:2006.14436* (2020).

[179] Nehorai, A., and Paldi, E. Acoustic vector-sensor array processing. *IEEE Transactions on signal processing 42*, 9 (1994), 2481–2491.

[180] Nguyen, P., Tran, D., Huang, X., and Sharma, D. Automatic classification of speaker characteristics. In *International Conference on Communications and Electronics 2010* (2010), pp. 147–152.

[181] Nguyen, T. N. T., Gan, W.-S., Ranjan, R., and Jones, D. L. Robust source counting and doa estimation using spatial pseudo-spectrum and convolutional neural network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 28* (2020), 2626–2637.

[182] Nguyen, T. N. T., Jones, D. L., and Gan, W.-S. Ensemble of sequence matching networks for dynamic sound event localization, detection, and tracking. In *DCASE* (2020), pp. 120–124.

[183] Nguyen, T. N. T., Jones, D. L., and Gan, W.-S. A sequence matching network for polyphonic sound event localization and detection. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 71–75.

[184] Nguyen, T. N. T., Nguyen, N. K., Phan, H., Pham, L., Ooi, K., Jones, D. L., and Gan, W.-S. A general network architecture for sound event localization and detection using transfer learning and recurrent neural network. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 935–939.

[185] Noh, K., Jeong-Hwan, C., Dongyeop, J., and Joon-Hyuk, C. Three-stage approach for sound event localization and detection. *Detection Classification Acoust. Scenes Events Challenge, Tech. Rep* (2019).

[186] Opochinsky, R., Chechik, G., and Gannot, S. Deep ranking-based doa tracking algorithm. In *2021 29th European Signal Processing Conference (EUSIPCO)* (2021), IEEE, pp. 1020–1024.

[187] Opochinsky, R., Laufer-Goldshtein, B., Gannot, S., and Chechik, G. Deep ranking-based sound source localization. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* (2019), IEEE, pp. 283–287.

[188] Oppenheim, A. V. Speech spectrograms using the fast fourier transform. *IEEE spectrum 7*, 8 (1970), 57–62.

[189] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038* (2019).

[190] Pak, J., and Shin, J. W. Sound localization based on phase difference enhancement using deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 27*, 8 (2019), 1335–1345.

[191] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), pp. 5206–5210.

[192] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2015), IEEE, pp. 5206–5210.

[193] Pariente, M., Cornell, S., Cosentino, J., Sivasankaran, S., Tzinis, E., Heitkaemper, J., Olvera, M., Stöter, F.-R., Hu, M., Martín-Doñas, J. M., et al. Asteroid: the pytorch-based audio source separation toolkit for researchers. *arXiv preprint arXiv:2005.04132* (2020).

[194] Park, S. Trellisnet-based architecture for sound event localization and detection with reassembly learning.

[195] Park, S., Jeong, Y., and Lee, T. Many-to-many audio spectrogram tansformer: Transformer for sound event localization and detection. In *DCASE* (2021), pp. 105–109.

[196] Park, S., Suh, S., and Jeong, Y. Sound event localization and detection with various loss functions. In *Proceedings of the Acoustic Scenes and Events 2020 Workshop (DCASE2020)* (2020), pp. 2–4.

[197] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems 32* (2019).

[198] Patel, S., Zawodniok, M., and Benesty, J. Dcase 2020 task 3: A single stage fully convolutional neural network for sound source localization and detection. *DCASE2020 Challenge* (2020).

[199] Paul, D. B., and Baker, J. M. The design for the wall street journal-based csr corpus. In *Proceedings of the Workshop on Speech and Natural Language* (USA, 1992), HLT '91, Association for Computational Linguistics, p. 357–362.

[200] Pepino, L., Riera, P., and Ferrer, L. Emotion recognition from speech using wav2vec 2.0 embeddings. *arXiv preprint arXiv:2104.03502* (2021).

[201] Perotin, L., Serizel, R., Vincent, E., and Guérin, A. Crnn-based joint azimuth and elevation localization with the ambisonics intensity vector. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)* (2018), IEEE, pp. 241–245.

[202] Perotin, L., Serizel, R., Vincent, E., and Guérin, A. Crnn-based multiple doa estimation using acoustic intensity features for ambisonics recordings. *IEEE Journal of Selected Topics in Signal Processing 13*, 1 (2019), 22–33.

[203] Phan, H., Pham, L., Koch, P., Duong, N. Q., McLoughlin, I., and Mertins, A. Audio event detection and localization with multitask regression network. In *Proceedings of the Acoustic Scenes and Events 2020 Workshop (DCASE2020)* (2020), pp. 2–4.

[204] Pickles, J. An introduction to the physiology of hearing. In *An Introduction to the Physiology of Hearing*. Brill, 2013.

[205] Politis, A., Mesaros, A., Adavanne, S., Heittola, T., and Virtanen, T. Overview and evaluation of sound event localization and detection in dcase 2019. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 29* (2020), 684–698.

[206] Pratik, P., Jee, W. J., Nagisetty, S., Mars, R., and Lim, C. Sound event localization and detection using crnn architecture with mixup for model generalization.

[207] Pu, Z., Bai, J., and Chen, J. Dcase 2021 task 3: Seld system based on resnet and random segment augmentation technical report.

[208] Pujol, H., Bavu, E., and Garcia, A. Source localization in reverberant rooms using deep learning and microphone arrays. In *23rd International Congress on Acoustics (ICA 2019 Aachen)* (2019).

[209] Pujol, H., Bavu, E., and Garcia, A. Beamlearning: An end-to-end deep learning approach for the angular localization of sound sources using raw multichannel acoustic pressure data. *The Journal of the Acoustical Society of America 149*, 6 (2021), 4248–4263.

[210] Quatieri, T. F. *Discrete-time speech signal processing: principles and practice*. Pearson Education India, 2002.

[211] Raangs, R., and Druyvesteyn, E. Sound source localization using sound intensity measured by a three dimensional pu-probe. In *Audio Engineering Society Convention 112* (2002), Audio Engineering Society.

[212] Ranjan, R., Jayabalan, S., Nguyen, T. N. T., and Gan, W. S. Sound event detection and direction of arrival estimation using residual net and recurrent neural networks.

[213] Ravanelli, M., Parcollet, T., and Bengio, Y. The pytorch-kaldi speech recognition toolkit. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 6465–6469.

[214] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. LibriParty synthetic dataset.

[215] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

[216] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. Voice Activity Detection with a (small) CRDNN model trained on Libriparty.

[217] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., et al. Speechbrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624* (2021).

[218] Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning* (2014), PMLR, pp. 1278–1286.

[219] Rho, D., Lee, S., Park, J., Kim, T., Chang, J., and Ko, J. A combination of various neural networks for sound event localization and detection. Tech. rep., Technical Report, DCASE 2021 Challenge, 2021.

[220] Rickard, S., and Yilmaz, O. On the approximate w-disjoint orthogonality of speech. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing* (2002), vol. 1, IEEE, pp. I–529.

[221] Rix, A., Beerends, J., Hollier, M., and Hekstra, A. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)* (2001), vol. 2, pp. 749–752 vol.2.

[222] Roden, R., Moritz, N., Gerlach, S., Weinzierl, S., and Goetze, S. *On sound source localization of speech signals using deep neural networks*. Technische Universität Berlin, 2019.

[223] Roman, N., and Wang, D. Binaural tracking of multiple moving sources. *IEEE Transactions on Audio, Speech, and Language Processing 16*, 4 (2008), 728–739.

[224] Ronchini, F., Arteaga, D., and Pérez-López, A. Sound event localization and detection based on crnn using rectangular filters and channel rotation data augmentation. *arXiv preprint arXiv:2010.06422* (2020).

[225] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (2015), Springer, pp. 234–241.

[226] Roy, R., and Kailath, T. Esprit-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing 37*, 7 (1989), 984–995.

[227] Rybach, D., Hahn, S., Lehnen, P., Nolden, D., Sundermeyer, M., Tüske, Z., Wiesler, S., Schlüter, R., and Ney, H. Rasr-the rwth aachen university open source speech recognition toolkit. In *Proc. ieee automatic speech recognition and understanding workshop* (2011).

[228] Rybka, J., and Janicki, A. Comparison of speaker dependent and speaker independent emotion recognition. *International Journal of Applied Mathematics and Computer Science 23* (12 2013).

[229] Salvati, D., Drioli, C., and Foresti, G. L. Incoherent frequency fusion for broadband steered response power algorithms in noisy environments. *IEEE Signal Processing Letters 21*, 5 (2014), 581–585.

[230] Salvati, D., Drioli, C., and Foresti, G. L. Exploiting cnns for improving acoustic source localization in noisy and reverberant conditions. *IEEE Transactions on Emerging Topics in Computational Intelligence 2*, 2 (2018), 103–116.

[231] Sampathkumar, A., and Kowerko, D. Sound event detection and localization using crnn models. Tech. rep., Technical Report, DCASE 2020 Challenge, 2020.

[232] Sarma, M., Ghahremani, P., Povey, D., Goel, N. K., Sarma, K. K., and Dehak, N. Emotion identification from raw speech signals using dnns. In *Interspeech* (2018), pp. 3097–3101.

[233] Satt, A., Rozenberg, S., and Hoory, R. Efficient emotion recognition from speech using deep learning on spectrograms. In *Interspeech* (2017), pp. 1089–1093.

[234] Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks 61* (2015), 85–117.

[235] Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation 34*, 3 (1986), 276–280.

[236] Schneider, S., Baevski, A., Collobert, R., and Auli, M. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862* (2019).

[237] Schwartz, O., and Gannot, S. Speaker tracking using recursive em algorithms. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 22*, 2 (2013), 392–402.

[238] Schymura, C., Bönninghoff, B., Ochiai, T., Delcroix, M., Kinoshita, K., Nakatani, T., Araki, S., and Kolossa, D. Pilot: Introducing transformers for probabilistic sound event localization. *arXiv preprint arXiv:2106.03903* (2021).

[239] Schymura, C., Ochiai, T., Delcroix, M., Kinoshita, K., Nakatani, T., Araki, S., and Kolossa, D. Exploiting attention-based sequence-to-sequence architectures for sound event localization. In *2020 28th European Signal Processing Conference (EUSIPCO)* (2021), IEEE, pp. 231–235.

[240] Sehgal, A., and Kehtarnavaz, N. A convolutional neural network smartphone app for real-time voice activity detection. *IEEE Access 6* (2018), 9017–9026.

[241] Shen, J., Nguyen, P., Wu, Y., Chen, Z., Chen, M. X., Jia, Y., Kannan, A., Sainath, T., Cao, Y., Chiu, C.-C., et al. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *arXiv preprint arXiv:1902.08295* (2019).

[242] Shillingford, B., Assael, Y., Hoffman, M. W., Paine, T., Hughes, C., Prabhu, U., Liao, H., Sak, H., Rao, K., Bennett, L., et al. Large-scale visual speech recognition. *arXiv preprint arXiv:1807.05162* (2018).

[243] Shimada, K., Koyama, Y., Takahashi, N., Takahashi, S., and Mitsufuji, Y. Accdoa: Activity-coupled cartesian direction of arrival representation for sound event localization and detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 915–919.

[244] Shimada, K., Takahashi, N., Takahashi, S., and Mitsufuji, Y. Sound event localization and detection using activity-coupled cartesian doa vector and rd3net. *arXiv preprint arXiv:2006.12014* (2020).

[245] Singla, R., Tiwari, S., and Sharma, R. A sequential system for sound event detection and localization using crnn. *DCASE, Tokyo, Japan, Tech. Rep. DCASE2020-Single-56 1* (2020), 1.

[246] Sinisetty, G., Ruban, P., Dymov, O., and Ravanelli, M. Commonlanguage, June 2021.

[247] Sivasankaran, S., Vincent, E., and Fohr, D. Keyword-based speaker localization: Localizing a target speaker in a multi-speaker environment. In *Interspeech 2018-19th Annual Conference of the International Speech Communication Association* (2018).

[248] Song, J.-m. Localization and detection for moving sound sources using consecutive ensembles of 2d-crnn.

[249] Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., and Zhong, J. Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 21–25.

[250] Sudarsanam, P., Politis, A., and Drossos, K. Assessment of self-attention on learned features for sound event localization and detection. *arXiv preprint arXiv:2107.09388* (2021).

[251] Sun, X., Zhu, X., Hu, Y., Chen, Y., Qiu, W., Tang, Y., He, L., and Xu, M. Sound event localization and detection based on crnn using adaptive hybrid convolution and multi-scale feature extractor. *DCASE* (2021).

[252] Sundar, H., Wang, W., Sun, M., and Wang, C. Raw waveform based end-to-end deep convolutional network for spatial localization of multiple acoustic sources. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 4642–4646.

[253] Takahashi, N., Goswami, N., and Mitsufuji, Y. Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation. In *2018 16th International workshop on acoustic signal enhancement (IWAENC)* (2018), IEEE, pp. 106–110.

[254] Takeda, R., and Komatani, K. Discriminative multiple sound source localization based on deep neural networks using independent location model. In *2016 IEEE Spoken Language Technology Workshop (SLT)* (2016), IEEE, pp. 603–609.

[255] Takeda, R., and Komatani, K. Unsupervised adaptation of deep neural networks for sound source localization using entropy minimization. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 2217–2221.

[256] Takeda, R., Kudo, Y., Takashima, K., Kitamura, Y., and Komatani, K. Unsupervised adaptation of neural networks for discriminative sound source localization with eliminative constraint. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 3514–3518.

[257] Tervo, S. Direction estimation based on sound intensity vectors. In *2009 17th European Signal Processing Conference* (2009), IEEE, pp. 700–704.

[258] Thuillier, E., Gamper, H., and Tashev, I. J. Spatial audio feature discovery with convolutional neural networks. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2018), IEEE, pp. 6797–6801.

[259] Tian, C. Multiple crnn for seld. *parameters 488211*, 508257 (2020), 490326.

[260] Tosi, O., Oyer, H., Lashbrook, W., Pedrey, C., Nicol, J., and Nash, E. Experiment on voice identification. *The Journal of the Audio Engineering Society 51*, 6 (1972), 2030–2043.

[261] Tsuzuki, H., Kugler, M., Kuroyanagi, S., and Iwata, A. An approach for sound source localization by complex-valued neural network. *IEICE TRANSACTIONS on Information and Systems 96*, 10 (2013), 2257–2265.

[262] Tüske, Z., Golik, P., Schlüter, R., and Ney, H. Acoustic modeling with deep neural networks using raw time signal for lvcsr. In *Fifteenth annual conference of the international speech communication association* (2014).

[263] Van Veen, B. D., and Buckley, K. M. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine 5*, 2 (1988), 4–24.

[264] Varanasi, V., Gupta, H., and Hegde, R. M. A deep learning framework for robust doa estimation using spherical harmonic decomposition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 28* (2020), 1248–1259.

[265] Varzandeh, R., Adiloğlu, K., Doclo, S., and Hohmann, V. Exploiting periodicity features for joint detection and doa estimation of speech sources using convolutional neural networks. In *ICASSP 2020-2020 IEEE International conference on acoustics, speech and signal processing (ICASSP)* (2020), IEEE, pp. 566–570.

[266] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

[267] Veaux, C., Yamagishi, J., and King, S. The voice bank corpus: Design, collection and data analysis of a large regional accent speech database. In *2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)* (2013), pp. 1–4.

[268] Vecchiotti, P., Ma, N., Squartini, S., and Brown, G. J. End-to-end binaural sound localisation from the raw waveform. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), IEEE, pp. 451–455.

[269] Vecchiotti, P., Pepe, G., Principi, E., and Squartini, S. Detection of activity and position of speakers by using deep neural networks and acoustic data augmentation. *Expert Systems with Applications 134* (2019), 53–65.

[270] Vecchiotti, P., Principi, E., Squartini, S., and Piazza, F. Deep neural networks for joint voice activity detection and speaker localization. In *2018 26th European Signal Processing Conference (EUSIPCO)* (2018), IEEE, pp. 1567–1571.

[271] Vera-Diaz, J. M., Pizarro, D., and Macias-Guarasa, J. Towards end-to-end acoustic localization using deep learning: From audio signals to source position coordinates. *Sensors 18*, 10 (2018), 3418.

[272] Vera-Diaz, J. M., Pizarro, D., and Macias-Guarasa, J. Acoustic source localization with deep generalized cross correlations. *Signal Processing 187* (2021), 108169.

[273] Vera-Diaz, J. M., Pizarro, D., and Macias-Guarasa, J. Towards domain independence in cnn-based acoustic localization using deep cross correlations. In *2020 28th European Signal Processing Conference (EUSIPCO)* (2021), IEEE, pp. 226–230.

[274] Vesperini, F., Vecchiotti, P., Principi, E., Squartini, S., and Piazza, F. A neural network based algorithm for speaker localization in a multi-room environment. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)* (2016), IEEE, pp. 1–6.

[275] Vincent, E., Gribonval, R., and Fevotte, C. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing 14*, 4 (2006), 1462–1469.

[276] Vo, B., Mallick, M., Bar-Shalom, Y., Coraluppi, S., Osborne, R., Mahler, R., and Vo, B. Multitarget tracking wiley encyclopedia of electrical and electronics engineering. *Wiley Encyclopedia of Electrical and Electronics Engineering* (2015), 1–15.

[277] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing 37*, 3 (1989), 328–339.

[278] Wang, D., and Chen, J. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 26*, 10 (2018), 1702–1726.

[279] Wang, Q., Wu, H., Jing, Z., Ma, F., Fang, Y., Wang, Y., Chen, T., Pan, J., Du, J., and Lee, C.-H. The ustc-iflytek system for sound event localization and detection of dcase2020 challenge. *IEEE AASP Chall. Detect. Classif. Acoust. Scenes Events* (2020).

[280] Wang, Y., Chen, T., Xu, H., Ding, S., Lv, H., Shao, Y., Peng, N., Xie, L., Watanabe, S., and Khudanpur, S. Espresso: A fast end-to-end neural speech recognition toolkit. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2019), IEEE, pp. 136–143.

[281] Wang, Z.-Q., Zhang, X., and Wang, D. Robust speaker localization guided by deep learning-based time-frequency masking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 27*, 1 (2018), 178–188.

[282] Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Soplin, N. E. Y., Heymann, J., Wiesner, M., Chen, N., et al. Espnet: End-to-end speech processing toolkit. *arXiv preprint arXiv:1804.00015* (2018).

[283] Wichern, G., Antognini, J., Flynn, M., Zhu, L. R., McQuinn, E., Crow, D., Manilow, E., and Roux, J. L. WHAM!: extending speech separation to noisy environments. In *Proc. Interspeech* (2019), pp. 1368–1372.

[284] Woodard, J., and Nelson, J. An information theoretic measure of speech recognition performance.

[285] Woodruff, J., and Wang, D. Binaural localization of multiple sources in reverberant and noisy environments. *IEEE Transactions on Audio, Speech, and Language Processing 20*, 5 (2012), 1503–1512.

[286] Wu, Y., Ayyalasomayajula, R., Bianco, M. J., Bharadia, D., and Gerstoft, P. Sound source localization based on multi-task learning and image translation network. *The Journal of the Acoustical Society of America 150*, 5 (2021), 3374–3386.

[287] Wu, Y., Ayyalasomayajula, R., Bianco, M. J., Bharadia, D., and Gerstoft, P. Sslide: Sound source localization for indoors based on deep learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 4680–4684.

[288] Xiao, X., Zhao, S., Zhong, X., Jones, D. L., Chng, E. S., and Li, H. A learning-based approach to direction of arrival estimation in noisy and reverberant environments. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), IEEE, pp. 2814–2818.

[289] Xu, B., Sun, G., Yu, R., and Yang, Z. High-accuracy tdoa-based localization without time synchronization. *IEEE Transactions on Parallel and Distributed Systems 24*, 8 (2012), 1567–1576.

[290] Xue, W., Tong, Y., Zhang, C., Ding, G., He, X., and Zhou, B. Sound event localization and detection based on multiple doa beamforming and multi-task learning. In *INTERSPEECH* (2020), pp. 5091–5095.

[291] Xue, W., Ying, T., Chao, Z., and Guohong, D. Multi-beam and multi-task learning for joint sound event detection and localization. *DCASE 2019 Detection and Classification of Acoustic Scenes and Events 2019 Challenge* (2019).

[292] Yalta, N., Nakadai, K., and Ogata, T. Sound source localization using deep learning models. *Journal of Robotics and Mechatronics 29*, 1 (2017), 37–48.

[293] Yalta, N., Sumiyoshi, Y., and Kawaguchi, Y. The hitachi dcase 2021 task 3 system: Handling directive interference with self attention layers. Tech. rep., Technical Report, DCASE 2021 Challenge, 2021.

[294] Yasuda, M., Koizumi, Y., Saito, S., Uematsu, H., and Imoto, K. Sound event localization based on sound intensity vector refined by dnn-based denoising and source separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 651–655.

[295] Yiwere, M., and Rhee, E. J. Distance estimation and localization of sound sources in reverberant conditions using deep neural networks. *Int. J. Appl. Eng. Res 12*, 22 (2017), 12384–12389.

[296] Yoon, Y.-S., Kaplan, L. M., and McClellan, J. H. Tops: New doa estimator for wideband signals. *IEEE Transactions on Signal processing 54*, 6 (2006), 1977–1989.

[297] Youssef, K., Argentieri, S., and Zarader, J.-L. A learning-based approach to robust binaural sound localization. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2013), IEEE, pp. 2927–2932.

[298] Yu, D., and Deng, L. *Automatic speech recognition*, vol. 1. Springer, 2016.

[299] Yu, D., Deng, L., and Dahl, G. E. Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition. In *NIPS 2010 workshop on Deep Learning and Unsupervised Feature Learning* (December 2010).

[300] Zakariah, M., Khan, M. K., and Malik, H. Digital multimedia audio forensics: past, present and future. *Multimedia tools and applications 77* (2018), 1009–1040.

[301] Zeghidour, N., and Grangier, D. Wavesplit: End-to-end speech separation by speaker clustering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 29* (2021), 2840–2849.

[302] Zeyer, A., Alkhouli, T., and Ney, H. Returnn as a generic flexible neural toolkit with application to translation and speech recognition. *arXiv preprint arXiv:1805.05225* (2018).

[303] Zhang, J., Ding, W., and He, L. Data augmentation and prior knowledge-based regularization for sound event localization and detection. *DCASE 2019 Detection and Classification of Acoustic Scenes and Events 2019 Challenge* (2019).

[304] Zhang, W., Zhou, Y., and Qian, Y. Robust doa estimation based on convolutional neural network and time-frequency masking. In *INTERSPEECH* (2019), pp. 2703–2707.

[305] Zhang, Y., Wang, S., Li, Z., Guo, K., Chen, S., and Pang, Y. Data augmentation and class-based ensembled cnn-conformer networks for sound event localization and detection. Tech. rep., Technical Report of DCASE Challenge. 2021. Available online: http://dcase ..., 2021.