Enhancing Speaker Diarization: Improved VAD & Transformer-Based Speaker Counting

Submitted in partial fulfillment of the requirements of
BITS F422T Thesis

By

Kanishk Yadav

Under the supervision of

Prof. Chng Eng Siong

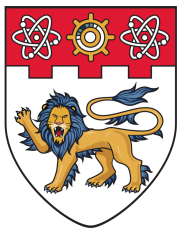
College of Computing and Data Science

NTU, Singapore

Prof. Aritra Mukherjee

Department of Computer Science

BITS Pilani, India



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

August 2024

Acknowledgement

I extend my sincere gratitude to my supervisor, Prof. Chng Eng Siong, for their guidance and unwavering support as my project progressed. I consider myself fortunate to have had the privilege of working under his mentorship, and I hold deep appreciation for his continuous encouragement and feedback. Furthermore, I express my heartfelt appreciation to my co-supervisor, Prof. Aritra Mukherjee, for his support and encouragement throughout this project. Their presence provided valuable academic oversight, contributing to the overall direction of this thesis.

I also take this opportunity to extend my sincere gratitude to all the members of the Speech Lab at the Nanyang Technological University, Singapore for their valuable contributions and insights during the weekly tele-conference(s). Collaborating and engaging with such a highly talented and dedicated group of researchers has enriched my research experience.

Certificate

This is to certify that this thesis entitled 'Enhancing Speaker Diarization: Improved VAD & Transformer-Based Speaker Counting' submitted by Kanishk Yadav, ID No 2019B2A71452H in partial fulfillment of the requirement of BITS F422T Thesis embodies the original work done by him under our supervision.

Prof. Chng Eng Siong

College of Computing and Data Science

NTU, Singapore

Assistant Prof. Aritra Mukherjee

Department of Computer Science

BITS Pilani, India

List of Abbreviations & Symbols

1. **AISG**: AI Singapore
2. **ASR**: Automatic Speech Recognition
3. **CNN**: Convolutional Neural Network
4. **DER**: Detection Error Rate
5. **DIHARD III**: (A specific dataset used in the thesis)
6. **FF**: Feed-Forward
7. **FN**: False Negative
8. **FP**: False Positive
9. **GRU**: Gated Recurrent Unit
10. **LSTM**: Long Short-Term Memory
11. **OSD**: Overlapped Speech Detection
12. **ROC-AUC**: Receiver Operating Characteristic - Area Under Curve
13. **RTF**: Real-Time Factor
14. **TN**: True Negative
15. **TP**: True Positive
16. **VAD**: Voice Activity Detection
17. **WavLM**: (A specific model used in the thesis)

Contents

Abstract

This thesis presents a comprehensive investigation into Voice Activity Detection (VAD) and Overlapped Speech Detection (OSD) as foundational components for robust VAD and speaker counting systems in diverse acoustic environments. The research systematically explores the performance of VAD using the Pyannote.audio toolkit, analyzing the impact of chunk length and stride on accuracy and computational efficiency, and highlighting domain-specific challenges in real-world audio. Building upon this, the study focuses on OSD for speaker counting, employing the WavLM model to classify audio segments by the number of active speakers (0, 1, or equal or greater than 2).

A core contribution involves evaluating various fine-tuning strategies for the WavLM model on both synthetic (LibriSpeech, SparseLibriMix) and real-world (DIHARD III) datasets. Direct fine-tuning on the DIHARD III dataset yielded an impressive overall accuracy of 0.87, significantly outperforming models trained solely on synthetic data. Furthermore, the thesis delves into sequential fine-tuning, demonstrating that an initial fine-tuning on LibriSpeech followed by DIHARD III can achieve comparable performance on the target DIHARD III dataset (0.87 accuracy). However, this strategy revealed a trade-off, leading to a slight decrease in accuracy on the original LibriSpeech dataset (0.86), particularly impacting the detection of silence, suggesting a form of catastrophic forgetting or feature re-prioritization towards the last fine-tuned domain. Conversely, fine-tuning on DIHARD III then LibriSpeech resulted in a performance drop on DIHARD III while maintaining higher accuracy on LibriSpeech.

The findings underscore the critical importance of domain-specific training data for optimal performance in speaker counting and provide valuable insights into the complexities and trade-offs associated with sequential model adaptation across disparate speech datasets. This research contributes to a deeper understanding of robust multi-speaker audio analysis and lays a foundation for future advancements in the field.

Chapter 1

Introduction

1.1 UNDERSTANDING SPEECH: THE FOUNDATION OF COMMUNICATION

Speech, a fundamental form of human communication, is a complex and dynamic interplay of linguistic, physical, and psychological elements. It is the primary means through which we convey thoughts, emotions, and intentions to others. In its raw form, speech is a continuous acoustic signal with variations in frequency, amplitude, and timing. However, these acoustic patterns encapsulate rich information, making speech an invaluable resource in various applications like telecommunication, voice-activated systems, and multimedia processing.

Despite its ubiquity and simplicity, processing and interpreting speech is a challenging task, especially for machines. This complexity arises from the inherent variability of speech. Factors such as accent, speed, pitch, and environmental noise contribute to the unique nature of each spoken word or phrase. These variances present a substantial challenge in developing robust and reliable speech processing systems.

Deep learning, a subset of artificial intelligence, has shown great promise in tackling these complexities. By leveraging large datasets and powerful computational resources, deep learning models can learn to recognize and interpret diverse speech patterns. However, the application of deep learning in speech processing is not without difficulties. The need for vast and diverse training datasets, computational expense, and the challenge of modelling temporal dependencies in speech signals are some of the key obstacles.

1.2 SPEAKER DIARIZATION

The crucial process of finding out *Who spoke when?* in an audio is precisely the goal of Speaker Diarization. This process involves identifying and segregating speech segments based on the speaker's identity, effectively partitioning an audio stream into homogenous segments that correspond to individual speakers. The term "Diarization" comes from the idea of creating a 'diary' or a log of who spoke at each point in time in the audio.

The concept of 'end-to-end' in Speaker Diarization refers to the ability of the system to take raw audio input and directly output diarization results, encompassing the entire pipeline from initial audio analysis to final speaker attribution. This approach contrasts with traditional multi-stage processing, where outputs from one stage become inputs for the next.

The key components of a Speaker Diarization Pipeline are as follows:

1. Voice Activity Detection
2. Speech Segmentation
3. Speaker Classification
4. Clustering
5. Re-segmentation

The above modules are discussed briefly in chapter 2.

1.3 CHALLENGES IN SPEAKER DIARIZATION

Speaker Diarization, while a powerful tool in speech processing, faces several significant challenges that impact its effectiveness and efficiency. These challenges occur from the inherent complexity of audio signals and the variability of human speech. Understanding these challenges is crucial for advancing the field and developing more robust diarization systems. Some of the primary challenges include:

1. **Variability in Speech Characteristics**: Human speech varies greatly due to factors like accent, pitch, speed, and tone. Each speaker has a unique vocal profile, and these profiles can change depending on the speaker's emotional state, health, or environment. This variability makes it difficult for diarization systems to consistently and accurately identify and distinguish speakers.

2. **Overlapping Speech**: In many real-world scenarios, such as meetings or conversations in noisy environments, speakers often talk over each other. This overlapping speech presents a significant challenge for diarization systems, which may struggle to separate and correctly attribute speech segments to the correct speakers.

3. **Background Noise and Acoustic Variability**: Background noise and varying acoustic conditions (like echoes in large rooms or sound distortions in poor quality recordings) can significantly impact the quality of the audio signal, making it challenging for diarization systems to isolate and identify individual speakers accurately.

4. **Segmentation Errors**: Dividing the continuous speech stream into distinct segments is a critical step in diarization. Errors in segmentation, such as incorrectly splitting a single speaker's speech or merging speech from different speakers, can lead to inaccuracies in the final diarization output.

5. **Scalability and Computational Efficiency**: As the number of speakers in an audio recording increases, the complexity of diarization also increases. Ensuring that diarization systems are scalable and can process audio efficiently without sacrificing accuracy is a significant challenge, especially for real-time applications.

6. **Limited Annotated Data**: Training effective diarization systems, particularly deep learning-based models, requires large amounts of annotated data. However, such data can be scarce, expensive to produce, and may not cover all possible variations in speech and acoustic environments.

7. **Domain Adaptation**: Diarization systems often struggle to maintain high performance when applied to audio data from domains different from the ones they were trained on. This includes adapting to different types of conversations, recording qualities, and environmental conditions.

Chapter 2

Speaker Diarization Pipeline

2.1 COMPONENTS OF THE PIPELINE

As stated in the previous introductory chapter, a speaker diarization pipeline can be explained using the following steps:

1. **Voice Activity Detection**: The pipeline's initial stage, during which the system separates speech from non-speech segments (such as background noise, music, or silence). It is essential for minimizing the volume of data that must be handled in subsequent phases.

2. **Speech Segmentation**: In this step, the continuous voice stream is divided into more manageable chunks. To identify the speaker, these portions can be examined in more detail. Segmentation merely produces manageable sections for examination; it does not always imply speaker change.

3. **Speaker Classification**: Classifying the speech segments by speaker identification comes next once the speech has been divided into segments. This entails taking characteristics (such as pitch, tone, speaking tempo, etc.) out of the speech signal and applying them to differentiate between several speakers.

4. **Clustering**: Segments that are identified as coming from the same speaker are grouped together in this stage. For clustering, a variety of techniques can be

applied, from straightforward distance metrics to intricate machine learning models.

5. **Re-segmentation**: Some diarization systems include a re-segmentation step, which refines the initial segmentation and clustering results. This can involve adjusting segment boundaries or reclassifying segments based on additional information obtained during the clustering process.

2.2 APPLICATIONS OF SPEAKER DIARIZATION

Speaker diarization is pivotal in business and academia, where it transforms meeting transcriptions and analyses. It accurately attributes each speaker's words in meetings, lectures, or seminars, aiding in documentation and enhancing communication efficiency. Especially useful in multi-participant discussions, speaker diarization ensures each voice is recognized, streamlining review and follow-up processes.

In media and entertainment, diarization significantly assists in transcribing shows and interviews, improving accessibility through precise subtitling and closed captioning. This is especially beneficial for hearing-impaired viewers. It also helps viewers easily follow who is speaking in programs with multiple speakers, thereby enhancing the viewing experience. Additionally, in e-learning, speaker diarization makes online classes more accessible and interactive by providing clear, speaker-annotated transcripts, thereby improving the learning experience.

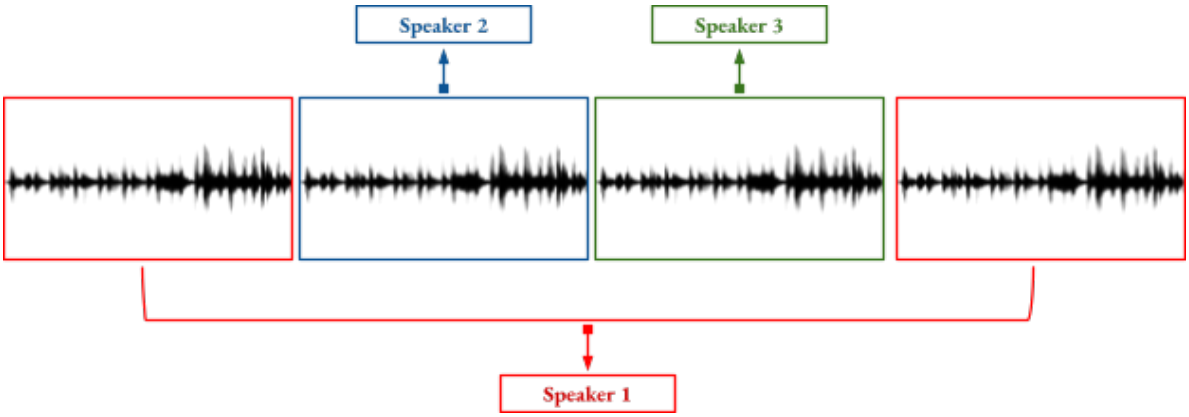


Figure 2

2.3 EVALUATION METRICS

Evaluating the performance of a Speaker Diarization system, specifically the Voice Activity Detection (VAD) component, is crucial to determine its effectiveness and accuracy. The VAD is responsible for distinguishing speech from non-speech segments in the audio. Here are key evaluation metrics, along with their formulas, used in assessing VAD performance:

1. **Accuracy (Acc)**: This measures the overall proportion of correctly identified segments (both speech and non-speech).

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

   Here, TP (True Positive) is the correctly identified speech segments, TN (True Negative) is the correctly identified non-speech segments, FP (False Positive) is the non-speech segments incorrectly identified as speech, and FN (False Negative) is the speech segments incorrectly identified as non-speech.

2. **Precision (P)**: This metric calculates the proportion of correctly identified speech segments among all segments identified as speech.

$$Precision = \frac{TP}{TP + FP}$$

3. **Recall (R) or Sensitivity**: This measures the proportion of actual speech segments that were correctly identified as speech.

$$Recall = \frac{TP}{TP + FN}$$

4. **F1-Score**: The F1-score is the harmonic mean of precision and recall, providing a balance between them.

$$F1\text{-}Score = \frac{2 \times P \times R}{P + R}$$

5. **Detection Error Rate (DER)**: While not specific to VAD but to speaker diarization, DER is important. It measures the total error based on missed speech, false alarms, and speaker confusion.

$$Detection\ Error\ Rate = \frac{\text{Missed speech} + \text{False alarm speech} + \text{Speaker error}}{\text{Total Speech}}$$

Each of these metrics offers a different perspective on the performance of the VAD component within a speaker diarization system. A combination of these metrics is often used to get a **comprehensive evaluation** of the system's performance.

6. **Receiver Operating Characteristic-Area Under Curve (ROC-AUC)**: In the context of VAD in diarization, it's used to measure how well the system distinguishes between speech and non-speech. The ROC curve is a plot of True Positive Rate (TPR) against False Positive Rate (FPR) at different thresholds. The AUC is the area under this ROC curve.

$$AUC = \int_{x=0}^{1} ROC(x)\, dx$$

7. **Real-Time Factor (RTF)**: RTF is used to measure the computational efficiency of a diarization system, indicating how fast the system processes audio compared to the actual duration of the audio.

Chapter 3

Voice Activity Detection

3.1 ONLINE AND OFFLINE VAD

Voice Activity Detection (VAD) systems can be categorized into two types based on their operational settings and use cases: Online VAD and Offline VAD.

Online VAD, also known as *real-time VAD*, is designed to work in real-time environments. It processes audio streams as they are being recorded or transmitted, making decisions about the presence of speech almost instantaneously. This type of VAD is crucial in applications where immediate response or interaction is required.

Characteristics:

1. **Low Latency**: Processes audio in small chunks to ensure minimal delay between audio input and VAD output.
2. **Incremental Processing**: Analyses audio data as it arrives without the need for the entire audio to be available beforehand.
3. **Resource Efficiency**: Often optimized for lower computational and memory usage to be suitable for embedded or real-time systems.

Applications:

1. Telecommunication systems like mobile phones and VoIP.
2. Real-time speech recognition systems, such as voice-activated virtual assistants.
3. Hearing aids and other assistive listening devices.

Offline VAD is used when the entire audio recording is available beforehand. It's typically applied in post-processing scenarios where there is no constraint on the processing time, and higher accuracy might be prioritized over immediate response.

Characteristics:

1. **High Accuracy**: Can utilize more complex algorithms and consider the entire context of the audio file, potentially leading to higher accuracy.
2. **Comprehensive Analysis**: Processes the complete audio recording, allowing for re-analysis and adjustment of decisions based on the full context.
3. **Flexibility in Resource Usage**: Since real-time processing is not a constraint, it can afford to use more computational resources and sophisticated models.

Applications:

1. Audio file transcription and analysis, such as in meeting recordings or podcasts.
2. Forensic analysis and archival searches where precision is more important than real-time processing.
3. Research purposes, where detailed analysis of speech patterns is required.

In summary, the choice between online and offline VAD depends on the specific requirements of the application, particularly in terms of the need for real-time processing versus the desire for higher accuracy and comprehensive analysis. Online VAD prioritizes immediacy and efficiency, suitable for real-time interactions, while offline VAD focuses on accuracy and thorough analysis, applicable in scenarios where time is not a critical factor.

3.2 CHUNK, STRIDE AND FRAME

Let's try to understand this with an example. Assume that the sampling rate (frame rate) is 16000 Hertz. This means that each second of audio contains 16000 frames.

1. **Frame**: A frame is a single sample of audio. At 16000 Hertz, each frame is 1/16000th of a second
2. **Chunk (Analysis Window)**: A chunk is a larger segment of audio consisting of multiple consecutive frames. For example, a chunk of 1 second would contain 16000 frames.
3. **Stride**: This is analogous to the term used in deep learning. It refers to the step size between processing of consecutive chunks. It determines how much the window (starting point) moves for the new chunk.

3.3 ILLUSTRATION

Chunk Size = *1 second* (16000 frames)
Stride = *0.5 second* (8000 frames)

Now, let's process a *3-second audio clip*.

First Chunk:

1. Start at the beginning (0 seconds).
2. The chunk covers 0 to 1 second.
3. In terms of frames, it includes frames 0 to 15999.

Second Chunk:

1. Start at 0.5 seconds (due to a stride of 0.5 seconds).
2. The chunk covers 0.5 to 1.5 seconds.
3. In terms of frames, it includes frames 8000 to 23999.

Subsequent Chunks:
This pattern continues, with each new chunk starting 8000 frames (0.5 seconds) after the start of the previous chunk.

Last Chunk:

1. The final chunk might not be a full second if the total audio duration isn't a multiple of the stride.
2. In this example, the last chunk would start at 2.5 seconds and go until the end of the audio at 3 second

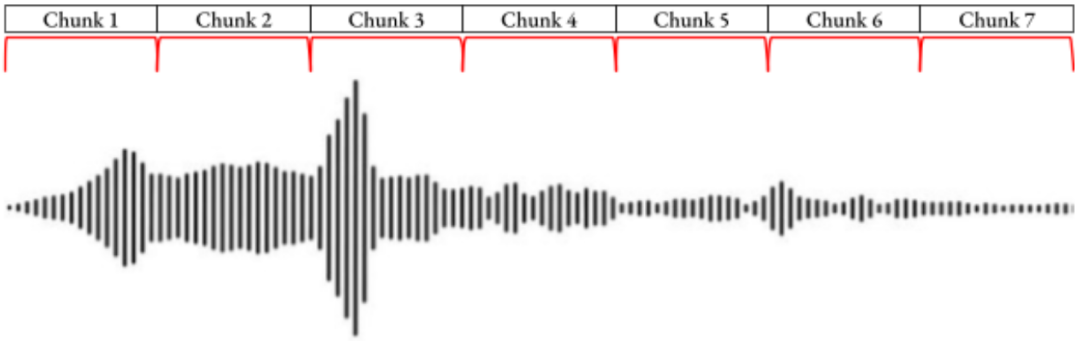


Figure 3

Chapter 4

Implementing Voice Activity Detection

4.1 DIHARD-III DATASET

The DIHARD III dataset is a challenging and diverse collection of audio recordings used for the evaluation of speaker diarization systems. It is the third edition in the DIHARD series of diarization challenges.

1. **Purpose and Scope**: DIHARD III is designed to test diarization systems under 'hard' conditions where traditional diarization approaches often struggle. It aims to advance research in speaker diarization by providing a diverse set of challenging real-world audio environments.

2. **Dataset Diversity**: The dataset includes a wide range of recording conditions and environments. This includes clean studio recordings, noisy field recordings, multi-party meetings, telephone conversations, and more. The variety in acoustic conditions and interaction styles poses significant challenges for diarization systems.

3. **Data Sources**: The audio sources for DIHARD III are diverse, including podcasts, audiobooks, clinical interviews, meetings, and telephone conversations. This variety helps in evaluating the robustness of diarization systems across different domains.

4. **Audio Characteristics**: The recordings vary significantly in terms of length, the number of speakers, speech styles, background noise, and channel quality. Some tracks contain overlapping speech, which adds complexity to the diarization task.

5. **Annotation and Evaluation**: The dataset is accompanied by rich annotations, including speaker labels with precise time markings. The challenge provides a standard evaluation protocol and metrics, ensuring consistency and comparability across different diarization systems.

6. **Challenge Tracks**: DIHARD III may include multiple tracks focused on different aspects of diarization, such as core diarization, diarization from unsegmented audio, or diarization in a specific domain like broadcast or conversational telephone speech.

Table 1: *Overview of DIHARD III datasets. The* **Part.** *column indicates the partition (core or full), while the* **% speech** *and* **% overlap** *columns indicate, respectively, the percentage of speech/overlapped speech in the partition.*

Set	Part.	# rec	# hours	% speech	% overlap
Dev	Core	181	23.94	78.43	10.04
	Full	254	34.15	79.81	10.70
Eval	Core	184	22.73	77.35	8.75
	Full	259	33.01	79.11	9.35

Table 1

Core evaluation set: a "balanced" evaluation set in which the total duration of each domain is approximately equal.

Full evaluation set: a larger evaluation set that uses all available selections for each domain; it is a proper superset of the core evaluation set.

Domains (environments): The domains span a range of recording conditions and interaction types, including read audiobooks, meeting speech, clinical interviews, web videos, and conversational telephone speech.

4.2 PYANNOTE ARCHITECTURE

The Pyannote.audio toolkit is designed for speaker diarization and related tasks, leveraging the power of deep learning.

1. **Raw Audio Input**: The model takes a waveform as input, representing the raw audio signal.

2. **SincConv Layer**: The first layer is a 'SincConv' layer, which is a type of convolutional layer specially designed for processing raw audio. Instead of learning all the filter coefficients, it only learns the parameters of a band-pass filter. This is efficient for feature extraction directly from the waveform and is based on the Sinc function, which is ideal for such tasks.

3. **Convolutional Layers**: Following the SincConv layer, there are additional convolutional layers. These layers can capture local patterns within the signal, such as the presence of certain frequencies over short time frames, and they gradually build a hierarchical representation of the audio signal.

4. **Recurrent Layers**: The output from the convolutional layers is then passed to recurrent layers. These layers can be LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) layers, which are adept at handling sequential data. They can capture temporal dependencies and patterns over time, which is critical for tasks like speaker diarization where the sequence of audio and changes over time are important.

5. **Temporal Pooling**: This step aggregates the information across time, summarizing the entire sequence of features into a single representation. This is useful when the subsequent layers need to make decisions based on the entire audio segment rather than on individual time steps.

6. **Feed-Forward Layers**: Finally, the aggregated representation is passed through one or more feed-forward layers (also known as fully connected layers). These layers can combine features in complex ways and output a final decision or representation, such as a speaker embedding or a classification result.

The Pyannote architecture effectively combines the strengths of convolutional layers for feature extraction, recurrent layers for capturing temporal dependencies, and feed-forward layers for decision-making. It's a powerful setup for various audio analysis tasks, particularly for speaker diarization where both the content of speech and its temporal structure are important.

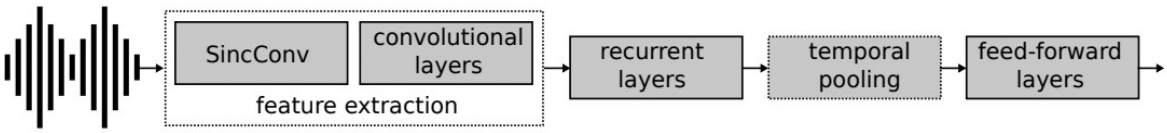


Figure 4

To increase the number of positive training examples for better generalization, Pyannote creates artificial audio segments by combining two different segments as shown below.

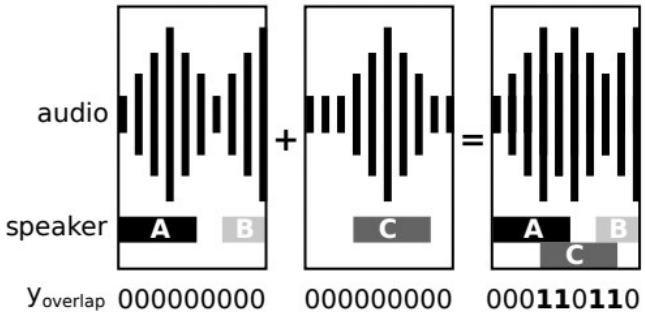


Figure 5

Chapter 5

Training & Testing

5.1 PRE-TRAINED MODELS

We initially begin with testing pre-trained Pyannote VAD models. The model is trained on the following parameters:

Chunk size (*window*)	*5 seconds*
Stride (*step*)	*4 seconds*
Batch size	*16*
Learning Rate	*1.0e-4*
Epochs	*50*
LSTM Block	*hidden layers: 3*
	size of each layer: 128
	dropout: 0.0
Linear Block	*hidden layers: 2*
	size of each layer: 128

Table 2

We performed inferencing on the model for various combinations of chunk size and stride at various thresholds. The table shown below contains the performance of these different combinations at various metrics. This was performed on the evaluation set of the DIHARD III challenge which contains 259 utterances with a total duration of 33.01 hours.

DIHARD III (evaluation set) *#hours:* ***33.01****, #utterances:* ***259***								
No.	**Model Size**	**Chunk Length**	**Stride**	**Accuracy**	**False Alarm**	**Missed Detection**	**ROC-AUC**	**Threshold**
1	Pyannote (2.4 Mb)	5 s	5 s	0.970	0.020	0.010	0.941	0.5
2		5 s	5 s	0.970	0.020	0.010	0.941	0.5
3		1 s	1 s	0.931	0.038	0.031	0.901	0.9
4		1 s	1 s	0.933	0.045	0.023	0.879	0.5
6		0.5 s	0.1 s	0.899	0.050	0.050	0.847	0.9
7		0.5 s	0.1 s	0.865	0.133	0.003	0.692	0.5

Table 3

Inference was also performed on a single noisy utterance containing a lot of background noise. The analysis for the same is shown in the table below.

Noisy utterance *#hours:* ***0.03****, #utterances:* ***1***								
No.	**Model Type**	**Chunk Length**	**Stride**	**Accuracy**	**False Alarm**	**Missed Detection**	**ROC-AUC**	**Threshold**
1	Pyannote	5 s	5 s	0.718	0.017	0.264	0.733	0.9
2	Pyannote	1 s	1 s	0.687	0.017	0.296	0.720	0.9
3	Pyannote	0.5 s	0.1 s	0.659	0.017	0.324	0.704	0.9

Table 4

To further analyze the domains for which the pre-trained model was performing poorly, we segmented the 259 utterances in the DIHARD III dataset and performed inference on the three domains: Meeting, Restaurant, and Web Video. For each of the three domains we used three different combinations of chunk size and stride. All the results are clearly tabulated in the table below.

DIHARD III (evaluation set: **full**)								
No.	**Domain**	**Chunk Length**	**Stride**	**Accuracy**	**False Alarm**	**Missed Detection**	**ROC-AUC**	**Threshold**
1	meeting	1 s	0.5 s	0.936	0.022	0.042	0.906	0.9
2	restaurant	1 s	0.5 s	0.859	0.030	0.111	0.801	
3	web video	1 s	0.5 s	0.870	0.071	0.059	0.832	
4	meeting	0.5 s	0.1 s	0.903	0.033	0.063	0.858	
5	restaurant	0.5 s	0.1 s	0.802	0.039	0.160	0.734	
6	web video	0.5 s	0.1 s	0.832	0.096	0.072	0.784	
7	meeting	0.3 s	0.1 s	0.864	0.044	0.092	0.808	
8	restaurant	0.3 s	0.1 s	0.758	0.044	0.198	0.693	
9	web video	0.3 s	0.1 s	0.807	0.110	0.083	0.752	

Table 5

To better understand SincNet block behavior, we experimented with different architecture settings, tuning the following hyperparameters:

1. Learning rate
2. LSTM block
    a. Hidden layer size
    b. # Layers
    c. Dropout
3. Linear Block
    a. Hidden layer size
    b. # Layers
    c. Dropout
4. Batch size
5. Evaluation loss

Table 6

DIHARD III (development set) *#utterances: 254,* ***chunk size: 5 s, stride size 4 s***									
			LSTM Block			**Linear Block**			
No.	**Epochs**	**Learning rate**	**Hidden layer size**	**#Layers**	**dropout**	**Hidden layer size**	**#Layers**	**Batch size**	**Evaluation Loss**
1	10	1.0e-4	128	3	0.0	128	2	16	0.16404
2		1.0e-4	128	2	0.0	128	3	16	0.16344
3		1.0e-4	256	2	0.0	128	2	16	0.16523
4		1.0e-3	256	3	0.0	128	2	16	0.16078
5		1.0e-3	256	2	0.3	128	2	8	0.15825
6		1.0e-3	256	3	0.3	128	2	8	0.15943

5.2 TRAINING PYANNOTE

The optimization of Voice Activity Detection (VAD) within the speaker diarization pipeline involved extensive training and evaluation of the pyannote.audio model. A crucial aspect of this optimization was the systematic exploration of *various chunk lengths and strides* during the training process. These parameters directly influence the temporal granularity of the VAD system, impacting both its real-time processing capabilities and its ability to accurately distinguish between speech and non-speech segments.

The primary objective of these experiments was to investigate the trade-offs between VAD performance (measured by metrics such as accuracy, false alarms, and missed detections) and the computational efficiency required for streaming applications. To this end, the Pyannote VAD model was trained and evaluated across a comprehensive set of chunk and stride combinations, ranging from larger segments providing more context to significantly smaller ones designed for finer-grained, low-latency processing.

The specific chunk and stride combinations used for training and evaluation on the DIHARD III (development set) are detailed below:

DIHARD III (development set)		
S.no	**Chunk (in seconds)**	**Stride (in seconds)**
1	3.0	2.0
2	1.5	1.0
3	0.75	0.5
4	0.1875	0.125

Table 7

Analysis 1

DIHARD III (evaluation set)								
No.	**Domain**	**Chunk Length**	**Stride**	**Accuracy**	**False Alarm**	**Missed Detection**	**ROC-AUC**	**Threshold**
1	All	3 s	2 s	**0.897**	**0.016**	**0.087**	**0.908**	0.9
2	restaurant			0.723	0.029	0.248	0.731	
3	web video			0.775	0.052	0.173	0.808	
4	meeting			0.834	0.008	0.159	0.874	

Analysis 2

DIHARD III (evaluation set)								
No.	**Domain**	**Chunk Length**	**Stride**	**Accuracy**	**False Alarm**	**Missed Detection**	**ROC-AUC**	**Threshold**
1	All	1.5 s	1 s	**0.882**	**0.012**	**0.105**	**0.903**	0.9
2	restaurant			0.692	0.017	0.291	0.751	
3	web video			0.767	0.042	0.191	0.807	
4	meeting			0.831	0.006	0.163	0.878	

Analysis 3

DIHARD III (evaluation set)								
No.	**Domain**	**Chunk Length**	**Stride**	**Accuracy**	**False Alarm**	**Missed Detection**	**ROC-AUC**	**Threshold**
1	All	0.75 s	0.5 s	**0.887**	**0.015**	**0.098**	**0.902**	0.9
2	restaurant			0.645	0.011	0.343	0.742	
3	web video			0.776	0.046	0.178	0.810	
4	meeting			0.836	0.007	0.158	0.880	

Analysis 4

DIHARD III (evaluation set)								
No.	**Domain**	**Chunk Length**	**Stride**	**Accuracy**	**False Alarm**	**Missed Detection**	**ROC - AUC**	**Threshold**
1	All	0.1875 s	0.25 s	**0.869**	**0.019**	**0.112**	**0.885**	0.9
2	restaurant			0.622	0.018	0.361	0.714	
3	web video			0.755	0.058	0.187	0.784	
4	meeting			0.801	0.012	0.188	0.848	

5.3 DIHARD III ACROSS VARYING CHUNK LENGTHS AND STRIDES

This comprehensive analysis examines the performance of the DIHARD III evaluation across four different configurations, varying in chunk length and stride. The consistent threshold of 0.9 allows for direct comparison of metrics: Accuracy, False Alarm, Missed Detection, and ROC-AUC.

Trends with Decreasing Chunk Length and Stride

As the *chunk length/stride* decrease from 3s/2s to 0.1875s/0.25s, a general trend of decreasing performance is observed, particularly in overall accuracy and ROC-AUC for the "All" domain.

1. **Accuracy:** Starts at 0.897 (3s/2s) and progressively decreases to 0.882 (1.5s/1s), 0.887 (0.75s/0.5s), and finally to 0.869 (0.1875s/0.25s). This indicates that longer audio chunks provide more context for better diarization performance.
2. **ROC-AUC:** Follows a similar downward trend, from 0.908 to 0.903, 0.902, and 0.885, reinforcing the finding that shorter chunks are less effective in distinguishing speakers.
3. **False Alarm:** Shows slight fluctuations, initially decreasing from 0.016 to 0.012, then slightly increasing to 0.015 and 0.019. While generally low, the slight increase with very short chunks might suggest increased sensitivity to non-speaker event.
4. **Missed Detection:** Generally, increases as chunk length decreases, from 0.087 to 0.105, 0.098, and 0.112. This suggests that shorter chunks make it harder to detect all speaker turns, leading to more missed segments.

Domain-Specific Performance

1. **Meeting Domain:** Consistently demonstrates the highest accuracy and lowest false alarm rates among the specific domains across all configurations. It shows remarkable resilience to decreasing chunk lengths, maintaining accuracy above 0.800 even at the shortest chunk (0.801 at 0.1875s/0.25s). This suggests that meeting audio, likely characterized by clearer speech and less background noise, is easier for the diarization system.

2. **Restaurant Domain:** Consistently performs the worst across all metrics and configurations. Its accuracy drops significantly with shorter chunks (from 0.723 to 0.622), and its missed detection rate is consistently the highest, often exceeding 0.300. This highlights the inherent challenges of restaurant audio, likely due to high background noise, overlapping speech, and reverberation.

3. **Web Video Domain:** Generally, performs better than *restaurant* but worse than *meeting*. Its performance also degrades with shorter chunks, particularly in accuracy. However, it maintains a relatively stable false alarm rate compared to *restaurant*.

5.4 CONCLUSION

This study demonstrates that chunk length and stride significantly impact the performance of VAD within speaker diarization systems. Longer chunks generally offer better performance due to the increased temporal context, as evidenced by higher accuracy and ROC-AUC scores and fewer missed detections. However, shorter chunks may still be necessary for low-latency or real-time applications, where responsiveness is critical.

Among evaluated domains, meeting audio consistently yields the best results, likely due to clearer speech and minimal noise. In contrast, restaurant recordings present significant challenges, underlining the importance of domain-specific tuning or advanced pre-processing strategies for highly noisy environments.

Ultimately, selecting optimal chunk and stride settings involves a careful trade-off between performance and latency, and should be tailored to the acoustic characteristics and application requirements of the target domain.

Chapter 6

Speaker Counting

6.1 OVERLAPPED SPEECH DETECTION

Overlapped Speech Detection (OSD) is a crucial component in speaker counting, aiming to identify segments of audio where multiple speakers are active simultaneously. Accurate OSD is fundamental for robust speaker diarization and counting systems, especially in complex real-world scenarios. This research utilizes an OSD system based on the AISG (Audio Intelligence and Speech Group) baseline, leveraging advanced deep learning models for classification. The primary goal is to classify audio segments into categories based on the number of active speakers: *0 speakers (silence)*, *1 speaker (monologue)*, or *2 speakers (overlapped speech)*.

6.2 DATASETS

The experimental evaluation of the OSD system was conducted using a combination of synthetic and real-world speech datasets to ensure comprehensive assessment under various conditions.

6.2.1 Libri2Mix

Libri2Mix is a synthetically generated dataset designed for speech separation and speaker counting tasks. It consists of mixtures created from two randomly selected clean audio files sourced from the LibriSpeech Dataset. Key characteristics include:

1. **Channel:** Mono-channel.
2. **Sampling Rate:** 16 kHz.
3. **Composition:** Mixtures of two distinct clean speech signals.

The dataset was partitioned for training, evaluation, and testing purposes as follows:

1. **Training Data:** 84 hours
2. **Evaluation Data:** 26 hours
3. **Test Data:** 25 hours

6.2.2 SparseLibriMix

SparseLibriMix extends the concept of mixed speech by introducing variable overlap percentages and optional noise. This dataset allows for the evaluation of OSD systems under more challenging conditions, mimicking diverse real-world acoustic environments. The variable overlap percentages range from 0% to 100% (specifically, [0, 0.2, 0.4, 0.6, 0.8, 1]), providing a spectrum of overlap scenarios. The test set for SparseLibriMix comprised 6 hours of audio.

6.2.3 Modified DIHARD III

The DIHARD III dataset, a prominent benchmark for speaker diarization, was adapted for this study. It was *modified into a Hugging Face object* format to facilitate seamless integration with the **WavLM** model. This modification involved chunking the audio and annotating each chunk with a speaker count label (0, 1, or 2). An example of the modified structure is shown below:


```
[
  {
    "audio_id": "DH_DEV_004",
    "input_values": [0.000213623046875, -3.0517578125e-05, -0.000213623046875, ...],
    "label": "speaker count for the chunk" (0/1/2),
    "chunk_filename": "DH_DEV_004_0.0_0.6"
  },
  {
    "audio_id": "DH_DEV_004",
    "input_values": [6.103515625e-05, 0.0001220703125, 0.000213623046875, ...],
    "label": "speaker count for the chunk" (0/1/2),
    "chunk_filename": "DH_DEV_004_0.6_1.2"
  }
]
```


The Modified DIHARD III dataset served as inference data, with a total duration of **67 hours**.

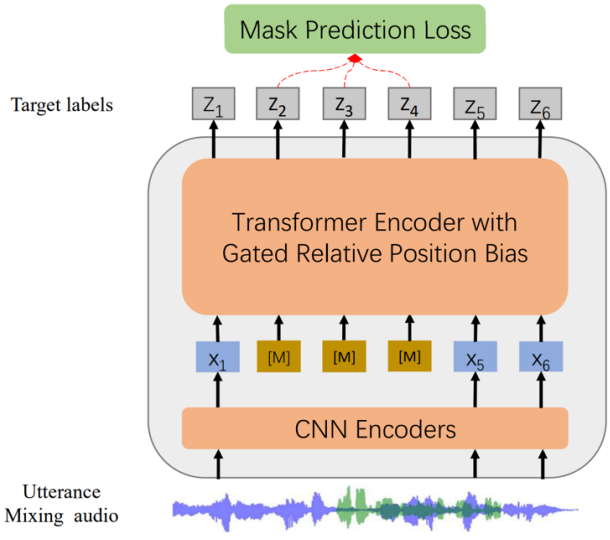
6.3 WAVLM MODEL: SEQUENTIAL TRAINING

The core of the OSD system is the WavLM model, a powerful self-supervised pre-trained model for speech processing.

WavLM employs a Transformer-based architecture, which is highly effective in capturing long-range dependencies in sequential data like audio. A key feature is the integration of a gated relative position bias, enhancing the model's ability to process sequential information by incorporating positional context. Furthermore, WavLM combines masked speech prediction with denoising objectives, making it versatile for both Automatic Speech Recognition (ASR) and non-ASR tasks. The model was pre-trained on an extensive dataset of 94k hours, including data from *Libri-Light, GigaSpeech, and VoxPopuli*, which contributes to its robust learning capabilities and adaptability across various speech applications.

The overall architecture involves:

1. **CNN Encoders:** Process raw audio input (waveform) to extract initial features.
2. **Feature Projection:** Applies Group Normalization, GELU Activation, and Dropout
3. **Masking (For Pre-training):** Includes Time masking and Channel masking.
4. **Transformer Encoder:** Comprises 12 layers and 12 heads, incorporating Self-Attention Layers, Gated Relative Position Bias, FF Layers, and Layer Norm with Dropout.
5. **Pooling:** Mean Pooling is applied to the output of the Transformer Encoder.
6. **Classification Head:** An output layer with a size of 3, corresponding to the classes: 0 Speaker, 1 Speaker, and 2 Speakers.



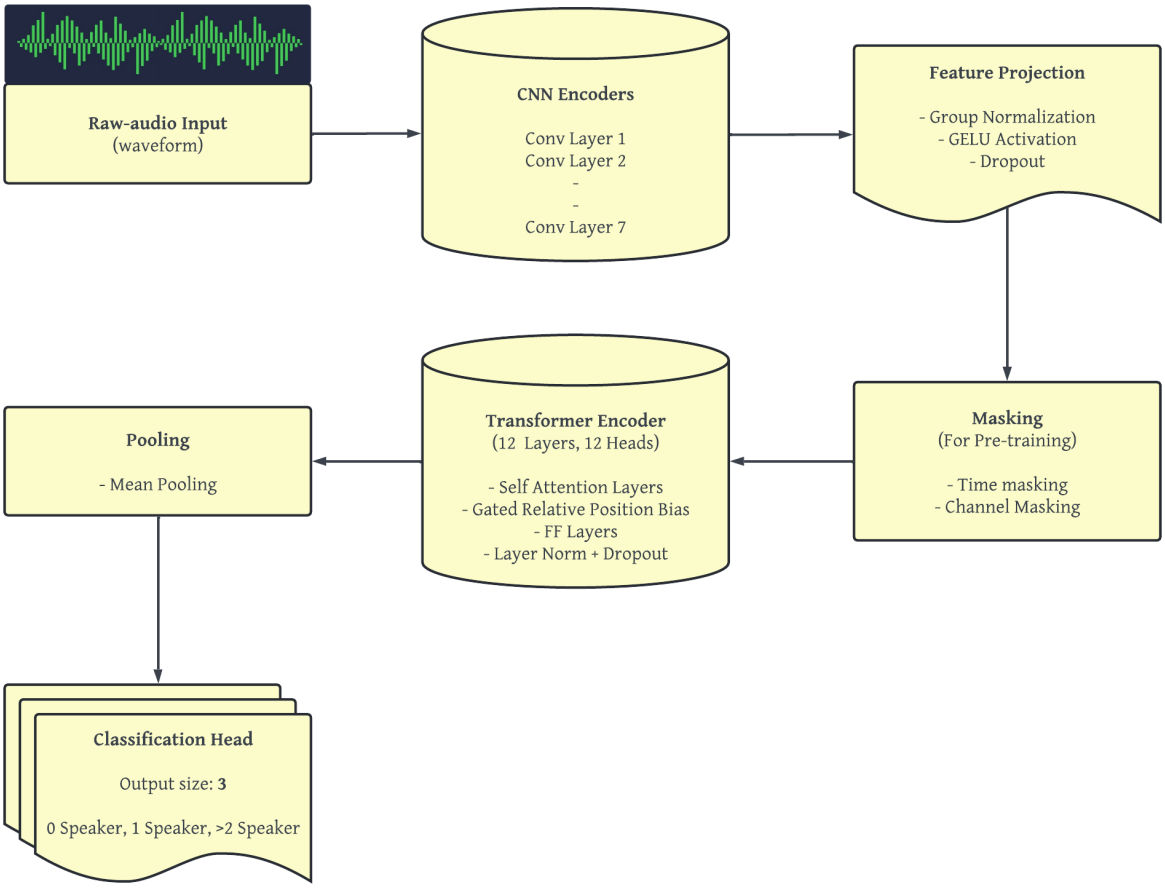


Figure 6 & 7: WavLM Architecture

The data processing pipeline for the OSD system involves several stages to prepare the audio data for WavLM model training and prediction:

1. **Hashing the Dataset:** Both Libri2Mix and SparseLibriMix datasets undergo a hashing process.
2. **Dataset Splitting:** The hashed dataset is then split into Training, Testing, and Validation sets.
3. **Audio Data Processing and Signal Segment Extraction:** Raw audio data is processed to extract relevant signal segments.
4. **Dataset Re-formatting:** The dataset is re-formatted with the newly chunked data.
5. **Further Chunking (Experimental):** Datasets are further split into smaller chunks, typically of size 600 milliseconds, for experimental purposes.
6. **WavLM Dataset Preparation:** The processed and chunked data forms the WavLM Dataset.
7. **Model Predictions:** The trained model is then used to make predictions on unseen data.

Various experimental configurations were explored to optimize the OSD system's performance and understand the impact of different parameters and training strategies.

S.No	Trained on	Tested (inference) on
1.	Pre-trained model (0.6 chunk)	DIHARD-III (0.6 chunk)
2.	Pre-trained model (0.6 chunk)	DIHARD-III (0.4 chunk)
3.	Fine-tuned (1) on LibriSpeech (0.6 chunk	DIHARD-III (0.6 chunk)
4.	Fine-tuned (1) on DIHARD-III (0.6 chunk)	DIHARD-III (0.6 chunk)

OSD Pipeline pre-trained on a chunk of size 0.6 s and inference performed on 0.6 s with DIHARD-III dataset

Inference performed with a chunk of 0.6 s using the **pre-trained** WavLM model, which is also trained on a chunk of 0.6 s

(0.6, 0.6)

Class (#speakers)	Precision	Recall	F1-Score	Support (#occurrences)
0	0.28	0.86	0.42	13987
1	0.81	0.82	0.81	147567
2	0.69	0.19	0.30	43450
Overall (Accuracy)			0.69	205004
Macro Avg	0.59	0.62	0.51	205004
Weighted Avg	0.74	0.69	0.68	205004

Confusion Matrix

Actual/Predicted	Class 0	Class 1	Class 2
Class 0	**12084**	1832	71
Class 1	23354	**120546**	3667
Class 2	7896	**27311**	8243

Experiments were conducted with different chunk sizes for audio processing. A primary chunk size of 0.6 seconds was consistently used for training and inference, with some evaluations also performed using a 0.4-second chunk size to assess its impact on performance, particularly when there was a mismatch between training and inference chunk sizes.

OSD Pipeline pre-trained on a chunk of size 0.6 s and inference performed on 0.4 s with DIHARD-III dataset

Inference performed with a chunk of 0.4 s using the **pre-trained** WavLM model, which is also trained on a chunk of 0.6 s

(0.6, 0.4)

Class (#speakers)	Precision	Recall	F1-Score	Support (#occurrences)
0	0.26	0.71	0.38	29843
1	0.83	0.77	0.80	224224
2	0.71	0.13	0.23	43094
Overall (Accuracy)			0.67	297161
Macro Avg	0.60	0.54	0.47	297161
Weighted Avg	0.76	0.67	0.67	297161

Confusion Matrix

Actual/Predicted	Class 0	Class 1	Class 2
Class 0	**21175**	8618	50
Class 1	49527	**172419**	2278
Class 2	11433	**25877**	5784

The pre-trained WavLM model underwent fine-tuning on specific datasets to adapt it to the speaker counting task. Two main fine-tuning approaches were investigated:

1. **Fine-tuning on Individual Datasets:**

    a. WavLM model fine-tuned solely on the LibriSpeech dataset (derived from Libri2Mix).
    b. WavLM model fine-tuned solely on the DIHARD III dataset.

OSD Pipeline (pre-trained) fine-tuned on the LibriSpeech dataset with a chunk of size 0.6 s and inference performed on 0.6 s with DIHARD-III dataset

Inference performed with a chunk of 0.6 s using the WavLM model, which is also fine-tuned on **LibriSpeech** dataset using chunk 0.6 s

(0.6, 0.6)

Class (#speakers)	Precision	Recall	F1-Score	Support (#occurrences)
0	0.28	0.89	0.43	14363
1	0.82	0.81	0.81	144056
2	0.66	0.18	0.28	39728
Overall (Accuracy)			**0.69**	198147
Macro Avg	0.59	0.62	0.61	198147
Weighted Avg	0.75	0.69	0.68	198147

Confusion Matrix

Actual/Predicted	Class 0	Class 1	Class 2
Class 0	**12764**	1499	100
Class 1	24295	**116146**	3615
Class 2	8254	**24322**	7152

OSD Pipeline (pre-trained) fine-tuned on the DIHARD-III dataset with a chunk of size 0.6 s and inference performed on 0.6 s with DIHARD-III dataset

Inference performed with a chunk of 0.6 s using the WavLM model, which is also fine-tuned on **DIHARD-III** dataset using chunk 0.6 s

(0.6, 0.6)

Class (#speakers)	Precision	Recall	F1-Score	Support (#occurrences)
0	0.83	0.78	0.80	14363
1	0.89	0.94	0.91	144056
2	0.77	0.65	0.70	39728
Overall (Accuracy)			**0.87**	198147
Macro Avg	0.83	0.79	0.81	198147
Weighted Avg	0.86	0.87	0.86	198147

Confusion Matrix

Actual/Predicted	Class 0	Class 1	Class 2
Class 0	**11189**	2940	234
Class 1	2022	**134769**	7265
Class 2	350	13751	**25627**

Significant reduction in misclassification and improved precision, recall, F1-score across all classes

2. **Sequential Fine-tuning on Combined Datasets:**

    a. **Strategy 1 (LibriSpeech then DIHARD III):** The pre-trained WavLM model was first fine-tuned on the LibriSpeech dataset, and then this fine-tuned model was further fine-tuned on the DIHARD III dataset.

    b. **Strategy 2 (DIHARD III then LibriSpeech):** Conversely, the pre-trained WavLM model was first fine-tuned on the DIHARD III dataset, and then this fine-tuned model was further fine-tuned on the LibriSpeech dataset.

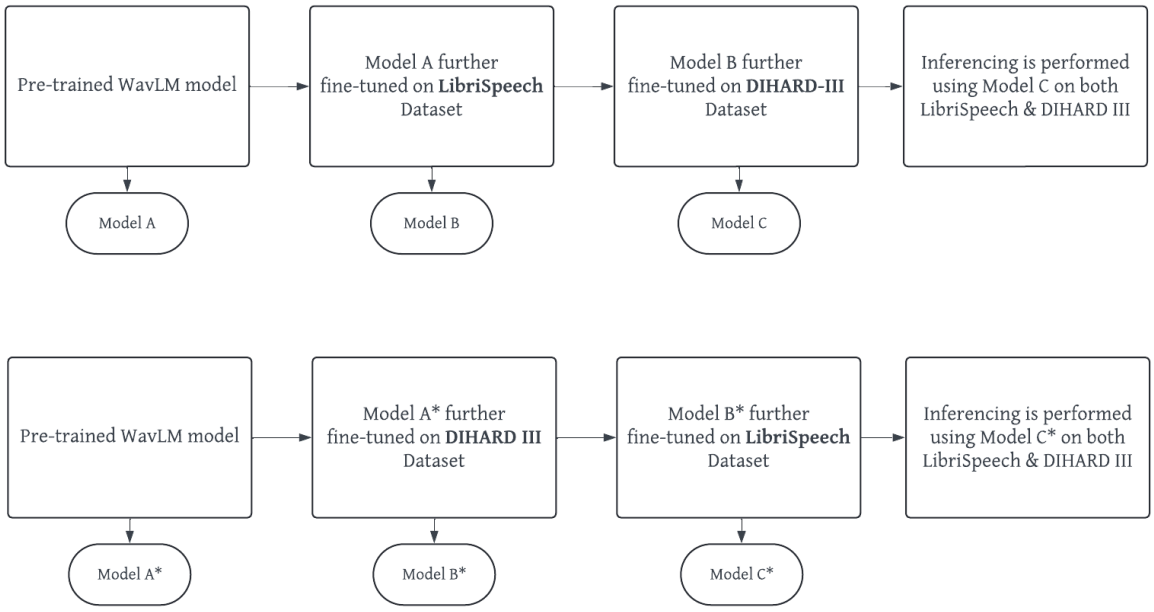


Figure 8

These sequential fine-tuning experiments aimed to understand the impact of the training data order on the model's generalization capabilities across different speech characteristics.

Throughout the training process, various hyperparameters were experimented with to optimize model performance. These included learning rates, batch sizes, and optimization algorithms. Specific details of the hyperparameter tuning process are beyond the scope of this chapter but contributed to achieving the reported results.

Investigation into different loss functions was also part of the experimental setup to determine their effect on the model's ability to accurately classify speaker counts. The *choice of loss function* plays a critical role in guiding the model's learning process and minimizing prediction errors.

Fine-tuned on	Tested on	Overall average accuracy	Average Precision	Average Recall	Average F1-score
LibriSpeech (AISG) (baseline)	LibriSpeech	**0.91**	**0.93**	**0.75**	**0.79**
	DIHARD III	**0.69**	**0.59**	**0.62**	**0.61**
DIHARD III	LibriSpeech	*Yet to perform inference (expecting poor performance)*			
	DIHARD III	0.87	0.83	0.79	0.81
LibriSpeech → DIHARD III **(achieved)**	LibriSpeech	**0.86**	**0.90**	**0.63**	**0.62**
	DIHARD III	**0.87**	**0.83**	**0.79**	**0.81**
DIHARD III → LibriSpeech	LibriSpeech	0.90	0.92	0.71	0.73
	DIHARD III	0.76	0.69	0.71	0.62

Average metrics denote the specific average metric for all classes: 0 speaker, 1 speaker and >2 speakers

6.4 RESULTS & CONCLUSION

When the WavLM model, sequentially fine-tuned on LibriSpeech and then DIHARD III, was evaluated on the DIHARD III dataset, it achieved an impressive overall accuracy of **0.87**. This performance is particularly significant as it matches the accuracy obtained by the model fine-tuned directly and exclusively on DIHARD III. This indicates that leveraging an initial fine-tuning phase on a large synthetic dataset like LibriSpeech, followed by adaptation to a more complex real-world dataset such as DIHARD III, can effectively prepare the model for the target domain without compromising its ultimate performance in that domain. The observed reduction in misclassification further underscores the success of this adaptation.

However, this advantage comes with a notable trade-off. While performing excellently on DIHARD III, the same sequentially fine-tuned model exhibited a slight decrease in overall accuracy to **0.86** when inferred back on the LibriSpeech dataset. This contrasts with the 0.91 accuracy achieved by a model fine-tuned solely on LibriSpeech.

In conclusion, the *sequential fine-tuning strategy* of starting with a broad synthetic dataset (LibriSpeech) and then specializing on a specific real-world dataset (DIHARD III) proves highly beneficial for optimizing performance within the target real-world domain. While this *approach yields excellent results* for the primary application, it highlights the inherent challenge of maintaining optimal performance across all previously encountered domains, particularly for less represented classes like silence, due to the nature of sequential learning.

Chapter 7

Conclusion

This thesis presented a comprehensive investigation into Voice Activity Detection (VAD) and Overlapped Speech Detection (OSD) as critical components for robust speaker counting systems. Through systematic experimentation with both Pyannote.audio for VAD and WavLM for OSD, this work has elucidated key factors influencing performance, including chunk length, stride, and fine-tuning strategies across diverse datasets.

The initial work on Voice Activity Detection (*discussed in Chapters 3-5*) demonstrated the significant impact of temporal parameters on VAD accuracy and efficiency. Longer audio chunks generally yielded superior performance, as evidenced by higher accuracy and ROC-AUC scores and fewer missed detections. However, a trade-off was observed with shorter chunks, which, while potentially necessary for low-latency or real-time applications, often led to a decrease in overall performance. Domain-specific analysis within the DIHARD III dataset revealed that meeting audio consistently presented fewer challenges for VAD systems, exhibiting higher accuracy and lower false alarm rates. Conversely, restaurant environments posed significant difficulties due to high background noise and reverberation, underscoring the need for domain-specific tuning or advanced pre-processing in such challenging acoustic conditions. This segment of the work provided crucial foundational understanding of audio segmentation and the inherent complexities of real-world acoustic environments, which directly inform the subsequent speaker counting efforts.

Building upon this foundation, the Speaker Counting component, primarily driven by Overlapped Speech Detection (*discussed Chapter 6 onwards*), focused on classifying audio segments by the number of active speakers (0, 1, or equal or great than 2). The WavLM model, a powerful self-supervised pre-trained architecture, was at the core of this system. Experiments on synthetic Libri2Mix and SparseLibriMix datasets showed high overall accuracies (around 0.91), though a persistent challenge was observed in accurately identifying 0-speaker segments (silence), which were frequently misclassified as 1-speaker segments. This highlights a common issue in OSD where silence can be ambiguous or contain subtle background noise interpreted as speech.

A major contribution of this thesis lies in the exploration of fine-tuning strategies for the WavLM model on real-world DIHARD III data. Direct fine-tuning on DIHARD III significantly improved performance on this challenging dataset, achieving an overall accuracy of 0.87, a substantial gain over models trained solely on synthetic data. This underscores the critical importance of training data that closely matches the target application domain.

Furthermore, the investigation into *sequential fine-tuning* provided valuable insights into model adaptation and generalization. The strategy of fine-tuning on LibriSpeech first, followed by DIHARD III, proved highly effective for performance on the DIHARD III dataset, matching the accuracy of direct DIHARD III fine-tuning (0.87). This demonstrates that an initial exposure to a large, diverse synthetic dataset can serve as a robust pre-adaptation step before specializing on a more complex, real-world target domain. However, a notable trade-off was observed: this sequential fine-tuning led to a slight decrease in overall accuracy (to 0.86) when the model was re-evaluated on the LibriSpeech dataset, particularly impacting the detection of silence. This phenomenon, often referred to as '*catastrophic forgetting*' indicates that the model's features became strongly biased towards the characteristics of the last fine-tuning dataset. Conversely, fine-tuning on DIHARD III first, then LibriSpeech, resulted in a significant performance drop on DIHARD III (to 0.76), while maintaining higher accuracy on LibriSpeech (0.90). This reinforces the concept that the final fine-tuning stage heavily dictates the model's specialized performance.

In conclusion, this research successfully developed and evaluated an OSD-based speaker counting system utilizing the WavLM model. It demonstrated that domain-specific fine-tuning is paramount for achieving high performance on challenging real-world datasets like DIHARD III. While sequential fine-tuning offers a promising avenue for leveraging diverse data sources, particularly when adapting from synthetic to real-world domains, it necessitates careful consideration of potential performance degradation on previously learned domains. The findings contribute to a deeper understanding of robust speaker counting in varied acoustic environments and provide a foundation for future advancements in multi-speaker audio analysis.

Bibliography

1. Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Zeng, M., & Wei, F. (2022). WAVLM: Large-Scale Self-Supervised Pre-Training for Full stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6), 1505-1518. https://doi.org/10.1109/jstsp.2022.3188113

2. Cornell, S., Omologo, M., Squartini, S., & Vincent, E. (2022). Overlapped Speech Detection and speaker counting using distant microphone arrays. *Computer Speech & Language*, 72, 101306. https://doi.org/10.1016/j.csl.2021.101306

3. CountNet: Estimating the number of concurrent speakers using supervised learning. (2019, February 1). *IEEE Journals & Magazine | IEEE Xplore*. https://ieeexplore.ieee.org/document/8506601

4. Fei Jia, Somshubhra Majumdar, and Boris Ginsburg. Marblenet: deep 1d time-channel separable convolutional neural network for voice activity detection. *arXiv preprint arXiv:2010.13886*, 2020.

5. H. Bredin, 'pyannote.audio: neural building blocks for speaker diarization,' *arXiv.org*, Nov. 04, 2019. https://arxiv.org/abs/1911.01255

6. H. Bredin, 'End-to-end speaker segmentation for overlap-aware resegmentation' *arXiv.org*, Apr. 08, 2021. https://arxiv.org/abs/2104.04045

7. Ryant, N., Singh, P., Krishnamohan, V., Varma, R., Church, K., Cieri, C., Du, J., Ganapathy, S., & Lieberman, M. (2020). The Third DIHARD Diarization Challenge. *arXiv.org/abs/2012.01477*

8. https://github.com/snakers4/silero-vad

9. Yu, F., Zhang, S., Fu, Y., Xie, L., Zheng, S., Du, Z., Huang, W., Guo, P., Yan, Z., Ma, B., Xu, X., & Bu, H. (2021). M2MeT: The ICASSP 2022 Multi-Channel Multi-Party Meeting Transcription Challenge. *arXiv.org/abs/2110.07393*