Otto-von-Guericke University Magdeburg

Faculty of Computer Science

Master Thesis

Pipeline for Automated Speaker-Attributed Video Transcription

Author:

Sunil Jaipal Ghadwal

June 7, 2024

Advisors:

Prof. Dr. Gunter Saake
Institute of Technical and Business Information Systems (ITI)
Otto-von-Guericke University Magdeburg

Dr.-Ing. David Broneske
Department Infrastructure and Methods
German Centre for Higher Education Research and Science Studies (DZHW)

M.Sc. Saijal Shahania
Department Infrastructure and Methods
German Centre for Higher Education Research and Science Studies (DZHW)

Ghadwal, Sunil Jaipal:
Pipeline for Automated Speaker-Attributed Video Transcription
Master Thesis, Otto-von-Guericke University Magdeburg, 2024.

Abstract

Speech-based applications that deal with conversations require not just the transcripts of words but also their respective speakers. Large speech models such as Whisper have achieved remarkable success in automatic speech recognition (ASR); however, they lack the ability to identify and distinguish speakers in a multi-speaker setting. Generally, they are coupled with speaker diarization (SD) tools to overcome this drawback. SD answers the question *"who spoke when?"* in a multi-speaker setting. Traditionally, SD systems have only focused on audio to perform diarization. However, acoustic data are inherently ambiguous as they generally contain mixed speech signals from several persons and noise from other sound sources. In the current digital age, video has become mainstream, and there is an abundance of video data. This thesis proposes to reduce the ambiguity issue of audio-based systems by utilizing the additional information available in the form of visual cues to generate speaker-attributed transcriptions. In this regard, a pipeline is designed that takes a video as input, performs audio-only diarization and video-based diarization independently, and then applies a rule-based algorithm using temporal overlap to assign speakers to Whisper-generated transcriptions. For the video-based diarization, an active speaker detection (ASD) system is used in conjunction with face recognition and clustering to perform diarization. Additionally, a dataset is constructed by manually annotating videos from YouTube with speaker-attributed transcriptions and face identifiers. Experiments showed that the addition of visual information improved the quality of speaker attribution when the videos had consistent frontal faces visible. It was also better at predicting the number of speakers for such videos. However, it struggled on videos where multiple variations, in terms of angle, distance from the camera, and partial occlusion, of the same faces were present in the videos.

Keywords: speaker diarization, active speaker detection, automatic speech recognition, face recognition

Acknowledgments

I would like to extend my deepest gratitude to my supervisors and mentors Prof. Dr. Gunter Saake, Dr.-Ing. David Broneske, and M.Sc. Saijal Shahania for their constant encouragement, expert guidance and continuous support, which has been crucial in bringing this thesis to completion.

Your dedication to the growth of the students and willingness to provide extensive support was a constant source of inspiration. Your expertise and mentorship have not only helped me with my thesis but also made a significant impact on my professional and personal growth.

Special thanks Dr.-Ing. David Broneske for his continuous feedback that ensured the quality of my thesis, and also to M.Sc. Saijal Shahania for her expert insights that helped shape the outlook of my thesis.

Thank you for all your contributions to my work and for making this thesis possible.

Contents

List of Figures

List of Tables

List of Algorithms

1. Introduction

In the current digital age, the amount of video data is continuously on the rise, primarily due to the continuous and rapid development of video capturing, storage and distribution technologies. Various domains, such as education, entertainment, business, and social media, are producing substantial amounts of video data that have provided an opportunity to extract valuable information from such a rich data source. This source was further augmented by the pandemic, as video became the go-to tool for communication in businesses, classrooms and other aspects of daily life.

This ever-growing data source, coupled with the advancements in deep learning, presents immense potential for developing and improving applications such as natural language processing (NLP), speech emotion detection, speech data mining, spoken document retrieval, summarization, semantic navigation, and others. A critical requirement for these applications is the transcripts of the spoken content in the videos. A comprehensive and accurate transcript is the foundation for the effective development of many of these downstream applications. The speech from these videos can be easily converted to text using automatic speech recognition (ASR) systems. The recent development of large speech models, such as Whisper [Radford et al., 2023], shows the significant advancements in the ability of the ASR systems to handle diverse accents, languages, and noisy environments. While these systems can effectively convert speech to text, they often lack the capability to capture the context of multi-speaker interactions. Identifying text segments from different speakers in a multi-speaker conversation is crucial for applications such as semantic navigation and speech data mining, where such differentiation can provide insights into conversational dynamics and speaker-specific contributions.

This limitation of ASR systems to fail to identify speakers can be overcome by combining them with a speaker diarization (SD) system. SD is the process of identifying and distinguishing between audio segments from different speakers. SD answers the question *"who spoke when?"* in a multi-speaker setting [Park et al., 2022]. The combination of ASR-SD will result in the sort of rich transcriptions that are crucial for various downstream tasks mentioned earlier.

SD has seen significant progress in the past few decades. The majority of this research has been focused towards the utilization of only audio to perform speaker diarization [Park et al., 2022] [Anguera et al., 2012]. While the audio-only approaches have achieved great success so far, they still have certain limitations due to the nature of the data source. Acoustic data are inherently ambiguous as they generally contain mixed speech signals from several persons along with noise from other sound sources. As a result, the quality and features of recorded audio could change depending on the recording environment, equipment quality, and in some cases, even a person's voice changes depending on the way they talk. It is difficult to account for such changes just based on the audio of the recordings.

Although these audio-based systems have achieved decent performance over the years, they have largely overlooked the potential of incorporating visual information from videos into these systems. Visual cues can help reduce ambiguity and provide additional context for distinguishing audio segments from different speakers in complex acoustic environments. Visual information regarding speakers can be extracted with the help of an active speaker detection (ASD) system. ASD is the task of identifying whether a person on screen is speaking. ASD systems are generally not concerned with the identity of the person speaking, and hence, by design, they do not have the capability of recognizing whether the person speaking in one frame is the same person speaking in another frame. However, this limitation can be overcome by coupling it with an external tool that can use features from the detected faces and group them based on similarity. Face recognition and clustering is one such approach to achieve this.

Goal of this Thesis

Ever since the introduction of Whisper [Radford et al., 2023], it has been gaining popularity as the go-to speech recognition model due to its versatility to cope with different voices and conditions without fine-tuning. Additionally, it is also capable of transcribing audio in a number of languages and can handle different accents effectively to achieve state-of-the-art results. However, as mentioned earlier, it does not have the capability to distinguish speakers of the transcriptions. In this regard, a speaker assignment framework is proposed that uses both audio and visual information to make decisions.

Specifically, the focus of this research is to enhance an ASD system by incorporating facial information to convert it into a video-based speaker diarization (VBSD) and then utilize the combination of this VBSD and an audio-only speaker diarization (AOSD) system to improve the speaker assignment of Whisper-generated transcripts, as compared to the use of only an AOSD. Essentially, this thesis tries to answer the following research questions:

- **RQ1** How effectively can an ASD system be used in conjunction with face clustering to create a VBSD system?
- **RQ2** How can the audio-only and video-based diarization systems be optimally combined to improve the speaker assignment?
- **RQ3** What are the major challenges concerning audio-visual characteristics that the combined approach still faces when performing speaker assignments?

Structure of this Thesis

The rest of the thesis is structured as follows:

- Chapter 2 introduces the theoretical and technical background necessary to implement the proposed system. It starts off with a brief discussion about the concepts of SD and ASD. Then, the introduction and design of pyannote.audio [Plaquet and Bredin, 2023] and TalkNet-ASD [Tao et al., 2021] are described in details. Lastly, the Whisper-ASR [Radford et al., 2023] and the face detection and recognition models are briefly introduced.
- Chapter 3 provides context regarding the progress made in the fields of SD and ASD in recent times. It establishes the research gap that was mentioned earlier in this chapter.
- Chapter 4 provides the design of the pipeline, including the implementation of individual models, data pre-processing, and the algorithm to generate the final output.
- Chapter 5 establishes the evaluation dataset and metric used to assess the performance of the proposed system.
- Chapter 6 discusses the evaluation results of the experiment and provides a breakdown of key factors affecting them.
- Chapter 7 concludes this thesis and outlines some intriguing directions for future research.

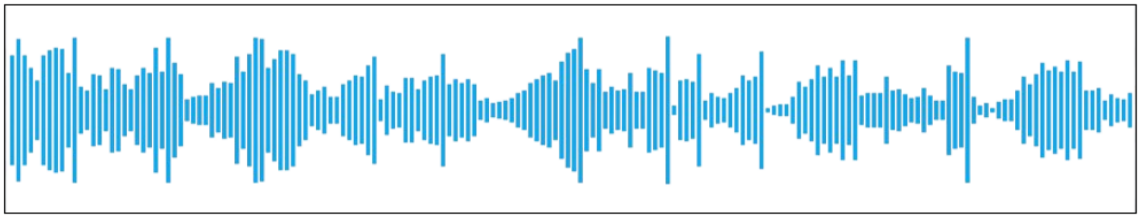
2. Background

This chapter presents the foundational concepts and key tools that form the basis of this research. The goal is to establish a strong base for the reader to better understand the research area and the proposed implementation. The structure of this chapter is as follows:

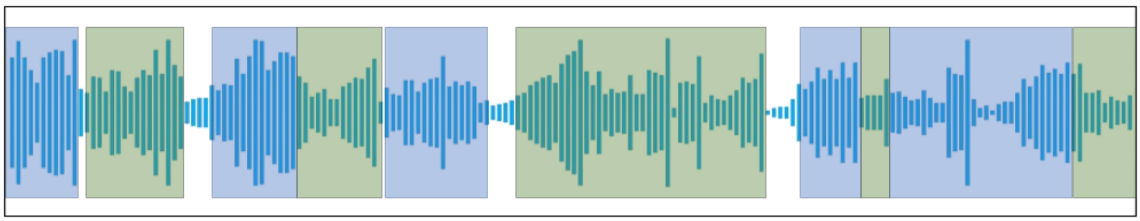
- Section 2.1 and Section 2.2 introduce the concepts of speaker diarization (SD) and active speaker detection (ASD) in detail, highlighting their importance, general architecture and sample outputs.
- Section 2.3 presents the general introduction and architecture of the pyannote.audio framework, along with the performance of its latest model. This forms the audio-only speaker diarization (AOSD) module of the pipeline and is used as a baseline to evaluate the performance of the proposed pipeline.
- In Section 2.4, the Talknet-ASD model is presented with its design and performance against the state-of-the-art. It is used in the video-based speaker diarization (VBSD) module of the pipeline to perform ASD.
- Section 2.5 talks about the design, training and performance of the Whisper-ASR model.
- Lastly, Section 2.6 describes the face recognition models used in this implementation.

2.1 Speaker diarization

Speaker diarization (SD) is the process of identifying and assigning segments of speeches to the respective speaker in an audio recording containing more than one speaker. It is not concerned with the identity of the person speaking or even the speech transcription. Also, it does not require prior information about the number of speakers in the recording. In simpler terms, the goal is to answer the question – *Who spoke when?* [Park et al., 2022] It involves analyzing audio recordings to identify segments of speeches and then distinguishing them based on the differences



(a) Without speaker diarization



(b) With speaker diarization

Figure 2.1: Audio (a)without and (b)with speaker diarization

in their audio characteristics. These segments are then grouped so that similar ones belong to the same group, and each group represents speech segments from one of the speakers.

Speaker diarization is a crucial step in a number of applications today involving audio and/or video processing, such as audio and speaker indexing, audio information retrieval and transcription content structuring [Park et al., 2022] [Anguera et al., 2012]. There are many domains where the analysis of speaker recordings is important, such as telephone conversations from call centers, broadcast news, debates and press conferences, talk shows, podcasts and interviews. With the world becoming more and more digital, even more, recordings emerge that need to be analyzed from online meetings, lectures, and question-answer sessions. In almost all of these cases, more than one active speaker is involved, so it can be advantageous to determine the number of speakers speaking and the intervals during which they were active.

An example of the application of speaker diarization is to combine it with a speech to text algorithm to perform speech and speaker indexing and document structuring. This involves combining the transcriptions generated from a speech-to-text algorithm with a speaker diarization algorithm to get a more meaningful output in the form of speaker-attributed transcriptions. Hence, it forms a key component in conversation analysis tools to help extract meaningful information from conversation content.

Traditionally, the process of speaker diarization consists of various sub-modules as shown in Figure 2.3 [Park et al., 2022] [Anguera et al., 2012] [Tranter and Reynolds, 2006].

Audio recordings come from various sources and might have been recorded in different acoustic environments, which could deteriorate the quality of the diarization output. So, preprocessing steps such as speech enhancement, dereverberation [Medennikov et al., 2020], normalization, and even denoising in some cases [Sun et al., 2018]. These steps result in a more consistent and clear audio file, which facilitates better processing and results. The preprocessed audio is then passed through a voice or speech activity detection module to detect the presence of human speech and distinguish it from

I'm going to name a sport. You have to tell me the greatest of all time. Basketball. LeBron James. Soccer. Messi. I agree with both. Baseball. Barry Bonds. Okay. Football like American football. Tom Brady, I guess. Okay, yeah. Good one. Tennis. Djokovic. Was it always Djokovic for you? Or you thought maybe. No, it was after the Australian Open thing that he became the greatest for me. Was it Nadal before or Federer? Probably Federer. But I always identified more with the way that Nadal plays.

Host:	I'm going to name a sport. You have to tell me the greatest of all time. Basketball.
Guest:	LeBron James.
Host:	Soccer.
Guest:	Messi.
Host:	I agree with both. Baseball.
Guest:	Barry Bonds.
Host:	Okay. Football like American football.
Guest:	Tom Brady, I guess.
Host:	Okay, yeah. Good one. Tennis.
Guest:	Djokovic.
Host:	Was it always Djokovic for you? Or you thought maybe.
Guest:	No, it was after the Australian Open thing that he became the greatest for me.
Host:	Was it Nadal before or Federer?
Guest:	Probably Federer. But I always identified more with the way that Nadal plays.

Figure 2.2: Transcription without and with speaker diarization

non-speech elements. This step is necessary to reduce computational expenses on non-speech sections of audio, subsequently reducing false predictions. The detected speech is then passed through a segmentation module, which divides the audio into several smaller segments so that similar segments can be grouped together. There are various approaches for segmentation, the main idea is to make sure that each segment does not consist of more than one speaker. For each of these segments, embedding vectors are generated to represent the acoustic features of the speech present in the segments. Lastly, clustering is performed to group these transformed segment embeddings and assign speaker labels to these groups. The clustering results are further refined in the post-processing stage and the output of the entire process is generated in the form of a Rich Transcription Time Marked (RTTM) file.

RTTM[1] is a space-delimited text file containing representations of the elements of recordings as objects or turns. Each turn is represented by ten fields listed below:

- Type: segment type; is always SPEAKER for diarization
- File ID: file name; name of the audio file without the extension
- Channel ID: channel number of the audio recording; 1 for mono channel recordings

[1] https://web.archive.org/web/20100606092041if_/http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf

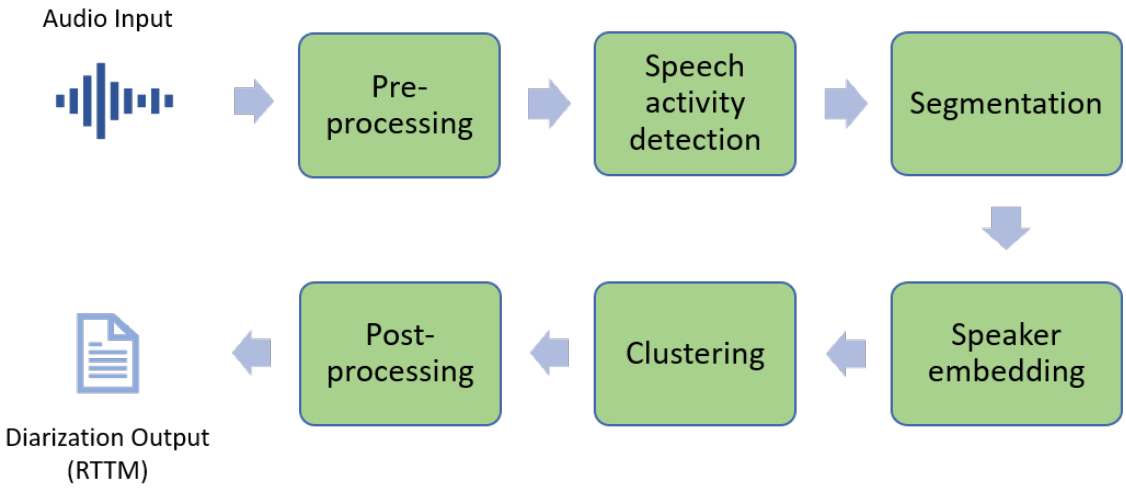


Figure 2.3: Traditional speaker diarization pipeline

- Turn Onset: start of the turn in seconds from the beginning of the recording
- Turn Duration: duration of turn in seconds
- Orthography Field: should always by <NA> for diarization
- Speaker Type: should always be <NA>
- Speaker Name: the name of the turn's speaker; is always unique within the scope of the file
- Confidence Score: the confidence of the system that the information is correct; should always be <NA>
- Signal Lookahead Time: should always be <NA> for diarization

Figure 2.4 represents a sample diarization output in the RTTM format.

```
SPEAKER _a9SWtcaNj8_c_01 1 1061.370000 2.616000 <NA> <NA> spk00 <NA> <NA>
SPEAKER _a9SWtcaNj8_c_01 1 1074.580000 2.170000 <NA> <NA> spk00 <NA> <NA>
SPEAKER _a9SWtcaNj8_c_01 1 1079.467000 1.093000 <NA> <NA> spk00 <NA> <NA>
SPEAKER _a9SWtcaNj8_c_01 1 1085.600000 0.730000 <NA> <NA> spk00 <NA> <NA>
```

Figure 2.4: RTTM Example

2.2 Active speaker detection

Active speaker detection (ASD) is the process of identifying whether a person in a visual scene is speaking or not. Video scenes are very fluid and change dynamically. It is often the case, that the person appearing on-screen is not always speaking, even if someone can be heard speaking in the video. Additionally, there can be multiple people on-screen but only one of them is speaking. Thus, an ASD model must identify and predict at very fine granularity in time, i.e. at the video frame level. ASD is an important front-end task in a number of applications such as audio-visual speech recognition [Afouras et al., 2022], speaker tracking [Qian et al., 2022] and speech separation [Pan et al., 2021].

For a human, the approach to identifying whether a person is speaking is based on the answer to the questions:

- Is the audio in question a human voice?
- Are the lips of the person in the frame moving?
- Is the audio in sync with the lip movement?

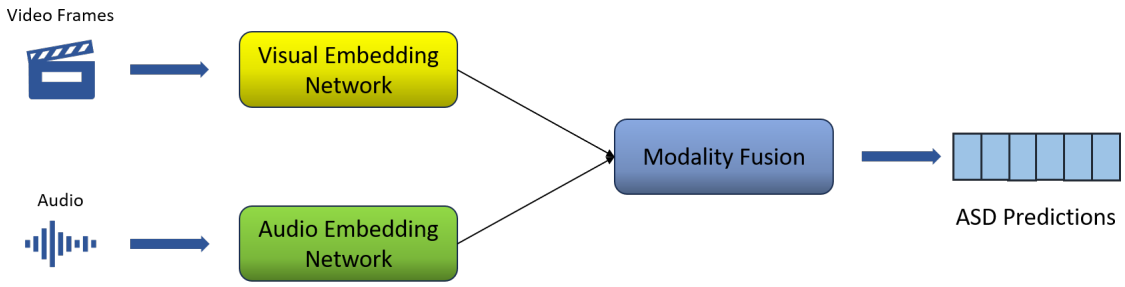


Figure 2.5: General end-to-end ASD architecture

An ASD framework also tries to mimic this behaviour. Figure 2.5 illustrates a general end-to-end ASD architecture [Roth et al., 2020][Tao et al., 2021]. Generally, it consists of two separate networks for encoding audio and visual data from a video. These models attempt to capture the speech activities in their respective modalities. The outputs are then fed to a modality fusion model that maps the relation between the captured audio and visual speaking activities to determine whether the person in the frame is speaking. Figure 2.6 demonstrates sample outputs from the end-to-end ASD model by Tao et al. [2021]. All the faces appearing in a frame are detected, but the ones speaking are marked with a green square, while the others are marked with a red square.

(a) ASD Output Example 1 **(b)** ASD Output Example 2

Figure 2.6: ASD output examples at different timestamps

2.3 pyannote.audio

pyannote.audio [Plaquet and Bredin, 2023] is an open-source toolkit designed in Python and is based on the PyTorch machine learning framework. As shown in Figure 2.3, speaker diarization involves a number of submodules that work together

to generate the diarization output. pyannote.audio provides an end-to-end neural network implementation for each of these sub-modules.

The input audio is treated as a sequence of fixed-length timesteps called audio frames, where each frame t is represented by a feature vector x_t. As a result, several of these sub-modules, such as voice activity detection [Gelly and Gauvain, 2018], speaker change detection [Yin et al., 2017], overlapped speech detection [Bullock et al., 2020], and re-segmentation [Yin et al., 2018] can be addressed as a sequence labelling task where the input is a sequence of feature vectors $X = \{x_1, x_2, ..., x_T\}$, where T is the number of audio frames. The expected output is also a sequence of labels $y = \{y_1, y_2, ..., y_T\}$ where $y_t \in [1, K]$ and K is the number of classes. The value of K depends on the specific task and can change for each task.

Several of these sub-modules are independent (usually recurrent) neural networks, and pyannote.audio provides a unified framework to train these networks. It also provides a generic code to train a neural network $f : X \longrightarrow y$, where X is the input feature sequence, and y is the corresponding label sequence.

The network architectures can be customized based on the user's choice. pyannote.audio also provides pre-trained PyTorch models[2]. These models share the same generic PyanNet base architecture shown in Figure 2.7. The architecture is used without pooling for sequence labelling and with pooling for embedding.

Audio files are usually long and of varying durations, and processing them directly would be impractical and inefficient. Hence, pyannote.audio breaks them down into shorter sub-sequences of fixed lengths. During training, it draws random fixed-length sub-sequences from the training set to form mini-batches. This helps in improving training sample invariability due to data augmentation, and also reduces training time due to shorter sequences. During testing, it uses overlapping sliding windows to process the audio files. The length of the windows is the same as used in training. This results in several overlapping sequences of K-dimensional prediction scores for each time step t. These scores are averaged to obtain the final scores of each class.

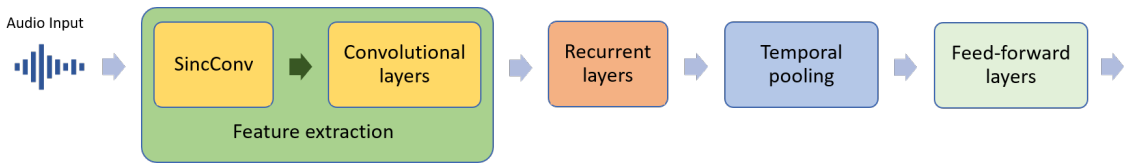


Figure 2.7: Generic PyanNet end-to-end architecture

Figure 2.8 illustrates the SD process that pyannote.audio framework follows to perform diarization. It introduces additional steps, such as overlapped speech detection and speaker change detection, to the generic SD process to improve its performance in a multi-speaker setting. The following subsections present the major components and submodules of the framework described in detail:

- Section 2.3.1 presents the feature extraction module that is common among several of the submodules, as they use the generic PyanNet architecture illustrated in Figure 2.7.

[2] https://huggingface.co/pyannote

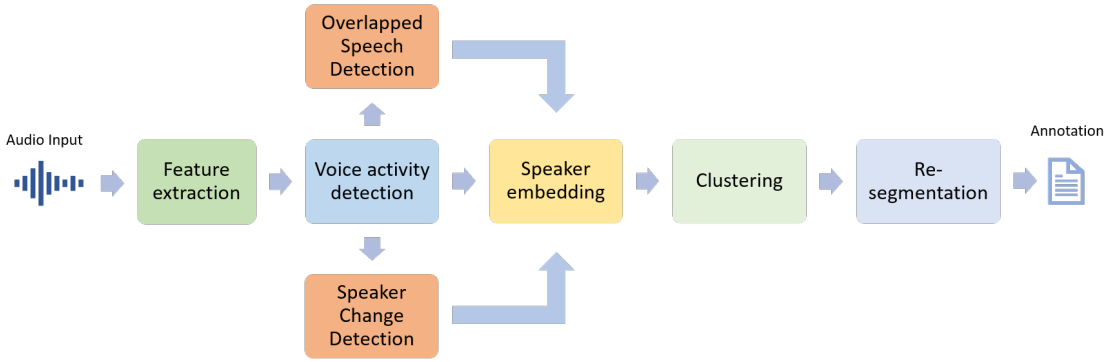


Figure 2.8: pyannote.audio pipeline

- Section 2.3.2, Section 2.3.3, Section 2.3.4, and Section 2.3.5 describe the voice activity detection, speaker change detection, overlapped speech detection and the resegmentation submodules in detail.
- Section 2.3.6 addresses the sequence embedding and clustering submodules.
- Lastly, Section 2.3.7 introduces the pre-trained model that is used in the proposed implementation.

2.3.1 Feature extraction

Although pyannote.audio supports model training from waveforms directly, using SincNet features [Ravanelli and Bengio, 2018] for instance, the features module of pyannote.audio also provides a collection of standard feature extraction techniques using the implementations available in the librosa library [Brian McFee et al., 2015] such as MFCCs and spectrograms. These implementations support on-the-fly data augmentation, which is very convenient for training neural networks. For example, it can extract features from random audio chunks while simultaneously applying additive noise from databases such as MUSAN [Snyder et al., 2015]. This enables it to generate virtually infinite versions of an audio chunk every time it is processed, as the augmentation is done on the fly. This contrasts with other tools that generate a fixed number of augmented versions of an audio file in advance.

2.3.2 Voice activity detection

Voice activity detection involves identifying speech or human voice sections in a given audio stream or recording. The sequence labelling principle discussed in Section 2.3 can be used to address this module with $K = 2$. So, at any given time step t:

- $y_t = 0$, if there is no speech present
- $y_t = 1$, if there is speech present

During testing, prediction scores are calculated for time steps, and the scores greater than a tunable threshold θ_{VAD} are marked as speech. The network used in this submodule is a simplified version of the voice activity detector originally described in Gelly and Gauvain [2018].

Table 2.1: Pre-trained voice activity detection model evaluation

	AMI			DIHARD			ETAPE		
	DetER	FA	Miss	DetER	FA	Miss	DetER	FA	Miss
Baseline				11.2	6.5	4.7	7.7	7.5	0.2
MFCC	6.3	3.5	2.7	10.5	6.8	3.7	5.6	5.2	0.4
waveform	6	3.6	2.4	9.9	5.7	4.2	4.9	4.2	0.7

*adapted from [Plaquet and Bredin, 2023]

pyannote.audio offers pre-trained models that achieve state-of-the-art performance on popular datasets AMI [Carletta et al., 2006], DIHARD [Ryant et al., 2019], and ETAPE [Gravier et al., 2012], as shown in Table 2.1. The performance is in terms of detection error rate (DetER %), which is a combination of the false alarm rate (FA %),i.e. the percentage of falsely detected speech timesteps, and the missed detection rate (Miss %), i.e. the percentage of missed speech timesteps. The table presents two variants of pyannote.audio used in the evaluation; the first was based on manually extracted features using MFCCs, and the second used the end-to-end model directly on input waveforms. As the table indicates, the model achieves state-of-the-art performance on these benchmark datasets. Also, the waveform variant outperforms the MFCC variant for all three datasets.

2.3.3 Speaker change detection

Speaker change detection is the task of identifying whether there is a change in the speaker at a given time step t in an audio stream or recording. It can be addressed using the same sequence labelling principle discussed in Section 2.3 with K = 2. So, at any given time step t:

- $y_t = 0$, if there is no speaker change
- $y_t = 1$, if there is speaker change

When it comes to training models for speaker change detection, most datasets suffer from a couple of problems. Firstly, as the number of time steps at which there is a speaker change is quite small compared to the time steps without a speaker change, there is a class imbalance problem. Secondly, as the datasets are manually annotated, it introduces a human annotation imprecision. To compensate for these problems, time steps $\{t \mid |t - t^*| < \delta\}$ which are in the close temporal neighbourhood of a speaker change time step t^* are artificially marked as positive. In practice, the value of δ is set around the order of magnitude of 200ms. During testing, the time steps are marked as speaker change points if the prediction score is greater than a tunable threshold θ_{SCD} and is the local maxima. This approach implements a simplified version of the speaker change detector originally described in Yin et al. [2017].

pyannote.audio offers pre-trained models that achieve state-of-the-art performance on popular datasets AMI [Carletta et al., 2006], DIHARD [Ryant et al., 2019], and ETAPE [Gravier et al., 2012], as shown in Table 2.2. The performance is in terms of coverage (%), i.e., the percentage of correctly identified speaker changes out of all

Table 2.2: Pre-trained speaker change detection model evaluation

	AMI		DIHARD		ETAPE	
	Purity	Coverage	Purity	Coverage	Purity	Coverage
Baseline					91	90.9
MFCC	89.4	78.7	92.4	74.5	90.1	95.9
waveform	90.4	84.2	86.8	93.7	89.3	97.2

*adapted from [Plaquet and Bredin, 2023]

correct speaker changes, and purity (%), i.e., the percentage of correctly identified speaker changes out of all predicted speaker changes. Two variants of pyannote.audio were used in the evaluation; the first was based on manually extracted features using MFCCs, and the second used the end-to-end model directly on input waveforms. Once again, the waveform variant outperforms the MFCC variant in almost all the categories.

2.3.4 Overlapped speech detection

Overlapped speech detection is the task of identifying sections of audio where more than one speaker is speaking at the same time. This can be addressed using the same sequence labelling principle discussed in Section 2.3 with K = 2. So, at any given time step t:

- $y_t = 0$, if there is zero or one speaker speaking
- $y_t = 1$, if there are more than one speaker speaking

As discussed in Section 2.3.3, the datasets used for model training in this case also suffer from the same class imbalance problem. To overcome this problem, the weighted sum of two random subsequences is used to create half of the training sub-sequences artificially. During testing, prediction scores are calculated for time steps, and the scores greater than a tunable threshold θ_{OSD} are marked as overlapped speech points.

Table 2.3: Pre-trained overlapped speech detection model evaluation

	AMI		DIHARD		ETAPE	
	Precision	Recall	Precision	Recall	Precision	Recall
Baseline	75.8	44.6			60.3	52.7
MFCC	91.9	48.4	58.0	17.6	67.1	57.3
waveform	86.8	65.8	64.5	26.7	69.6	61.7

*adapted from [Plaquet and Bredin, 2023]

pyannote.audio offers pre-trained models that achieve state-of-the-art performance on popular datasets AMI [Carletta et al., 2006], DIHARD [Ryant et al., 2019], and ETAPE [Gravier et al., 2012], as shown in Table 2.3. The performance is in terms of precision (%), i.e., the percentage of correct overlap detections out of all detected

overlaps, and recall (%), i.e., the percentage of correct overlap detections out of all correct overlaps. Two variants of pyannote.audio were used in the evaluation; the first was based on manually extracted features using MFCCs, and the second used the end-to-end model directly on input waveforms. Both variants outperform the baselines by a significant margin.

2.3.5 Re-segmentation

Towards the end of the pipeline, re-segmentation is performed in which speech turn boundaries and labels coming out of a diarization pipeline are refined. In pyannote.audio, this is also addressed using the same sequence labelling principle, even though this is an unsupervised task. In this case, K = k + 1, where k is the number of speakers predicted by the diarization pipeline. So, at a given time step t:

- $y_t = 0$, if no speaker is active
- $y_t = s$, if speaker $s \in [1, k]$ is active

As this is an unsupervised task by design, a re-segmentation model cannot be pre-trained. Hence, a new re-segmentation model is trained for each audio file from scratch. The output from the diarization pipeline is used as training labels, and the model is trained for a certain number of epochs ϵ. Each epoch is one complete pass on the file. Once trained, the model is applied to the same file and prediction scores for each class are computed for each time step. The time steps are assigned to either the non-speech class or one of the k speaker classes based on the highest score. This implementation is a version of the approach discussed initially in Yin et al. [2018], where it was derived that a reasonable value for ϵ is 20. The implementation can extended a bit more, by also assigning time steps with overlapped speech, the class with the second highest prediction score, which may result in significant performance improvement.

2.3.6 Sequence embedding and clustering

Like the voice activity detection submodule, clustering is also one of the most crucial steps of a speaker diarization pipeline. In this step, similar speech segments are grouped so that speech segments belonging to the same speaker are placed in the same group. Typically, speaker embeddings are generated using x-vectors [Snyder et al., 2018] extracted using a sliding window and compared using probabilistic linear discriminant analysis (PLDA) [Ioffe, 2006]. pyannote.audio implements a different approach.

In pyannote.audio, the speaker embeddings are trained by leveraging metric learning, simplifying clustering. The embeddings are directly optimized for a specific distance metric, often cosine similarity, eliminating the need for techniques like PLDA.

2.3.7 Pre-trained model

As the previous subsections indicated, pyannote.audio incorporates several state-of-the-art models for each of the submodules of its diarization pipeline. It also

provides the resulting pipeline from these models that can be directly applied to audio inputs to perform diarization. At the time of this research, the latest speaker diarization pipeline provided by pyannote was speaker-diarization-3.0[3]. It takes as input mono-audio sampled at 16kHz and produces the speaker diarization output as an instance of the pyannote's annotation class, which houses each of the speaker's speaking duration and can also generate a diarization plot as shown in Figure 2.9. It downmixes stereo or multichannel audio files to mono by averaging the channels. Also, audio files which are sampled at different rates are automatically resampled to 16kHz upon loading.

The speaker diarization pipeline has been evaluated on a large collection of datasets as shown in Table 2.4. The evaluation measure is the Diarization Error Rate (DER), which is a combination of the falsely predicted rate (FA%), the missed predictions (Miss%), and the confusions or the wrong predictions (Conf%). The processing of the datasets was fully automatic, with no manual voice activity detection, no manually provided number of speakers and no fine-tuning of the internal models to each dataset. As the table illustrates, the pipeline performs relatively well on several benchmark datasets. It also indicates that the pipeline has a very high generalizability, which makes it an ideal choice for this proposed implementation.

Table 2.4: Speaker diarization 3.0 evaluation

Benchmark	DER%	FA%	Miss%	Conf%
AISHELL-4	12.3	3.8	4.4	4.1
AliMeeting (channel 1)	24.3	4.4	10.0	9.9
AMI (headset mix, only_words)	19.0	3.6	9.5	5.9
AMI (array1, channel 1, only_words)	22.2	3.8	11.2	7.3
AVA-AVD	49.1	10.8	15.7	22.5
DIHARD 3 (Full)	21.7	6.2	8.1	7.3
MSDWild	24.6	5.8	8.0	10.7
REPERE (phase 2)	7.8	1.8	2.6	3.5
VoxConverse (v0.3)	11.3	4.1	3.4	3.8

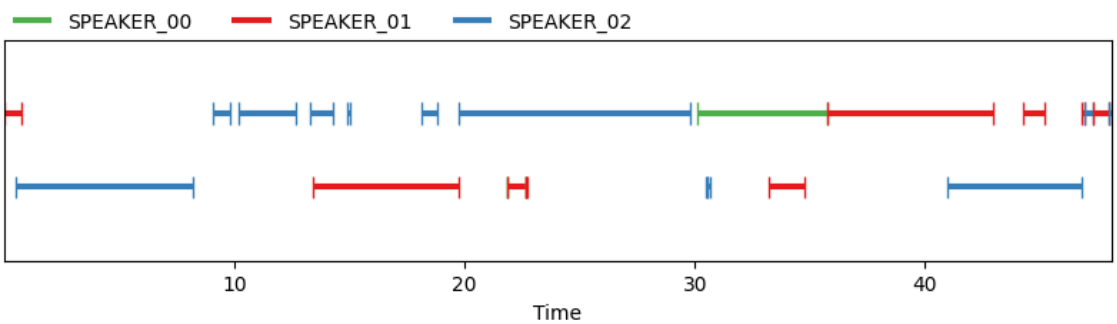


Figure 2.9: pyannote.audio annotation output

[3]https://huggingface.co/pyannote/speaker-diarization-3.0

2.4 Talknet-ASD

Talknet-ASD[Tao et al., 2021] is an ASD framework that uses long-term temporal context and the interaction of both audio and visual cues to decide whether a person in the video frame is speaking. It takes cropped face videos along with their corresponding audio as inputs and predicts whether a person is speaking in each video frame. Figure 2.10 represents the framework's architecture.

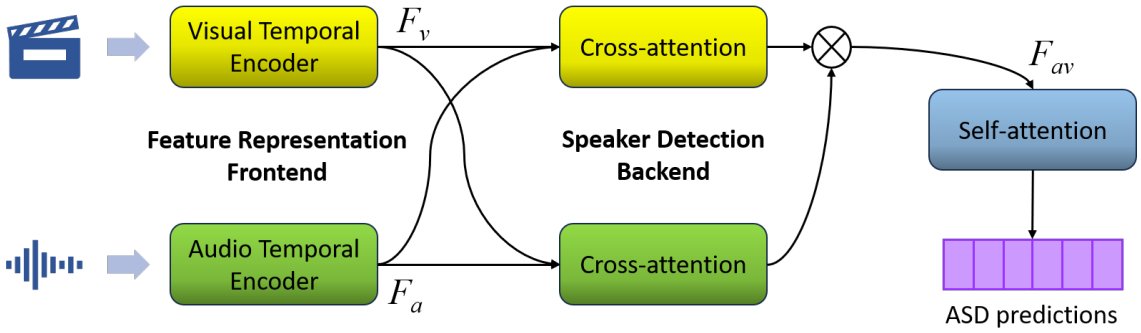


Figure 2.10: Talknet architecture

The frontend is a feature representation framework consisting of separate video and audio temporal networks that encode temporal context over multiple video and audio frames. The backend is a speaker detection classifier that passes these encodings to a cross-attention mechanism to capture the inter-modality evidence. Finally, a self-attention mechanism is used to capture the long-term speaking evidence.

The following subsections describe these submodules and their design in more detail. Additionally, Section 2.4.5 talks about the performance of this framework as compared to the state-of-the-art on the benchmark dataset.

2.4.1 Visual temporal encoder

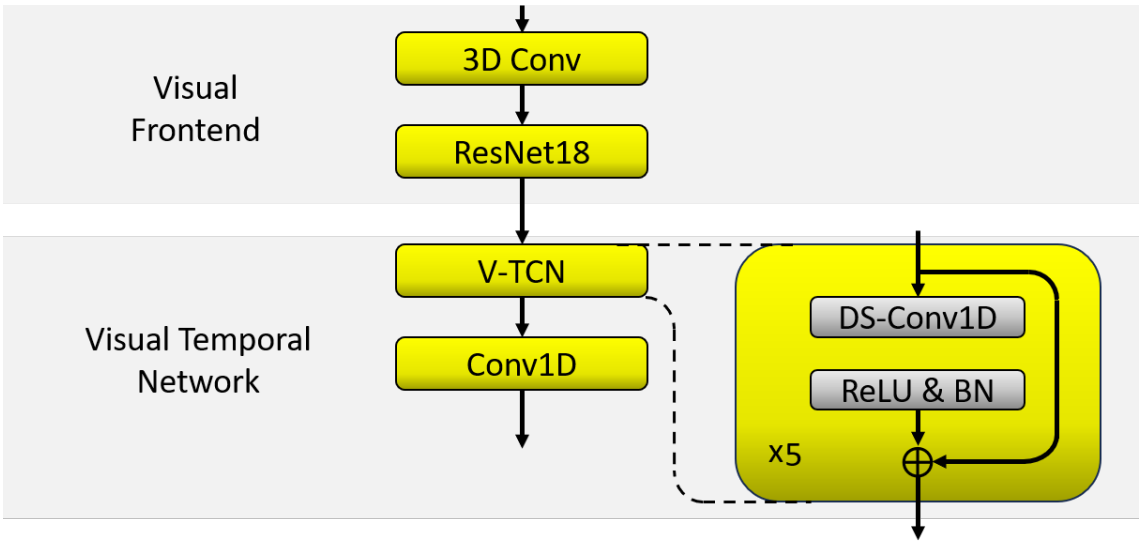


Figure 2.11: Visual temporal encoder structure

The visual temporal encoder consists of the visual frontend and a visual temporal network as illustrated in Figure 2.11. It captures the long-term representation of facial expression changes and transforms a video stream into a series of visual embeddings F_v.

The visual frontend focuses on the spatial information in each video frame. It combines a 3D convolution layer and a pre-trained ResNet18 block [Afouras et al., 2018], resulting in a sequence of frame-based embeddings.

The visual temporal network then captures the temporal information. It comprises a video-temporal convolution block (V-TCN), followed by a Conv1D layer, which reduces the feature dimensions. The V-TCN consists of five combinations of residual connected rectified linear units (ReLU), batch normalization (BN), and depth-wise separable convolutional layers (DS-Conv1D). It transforms the frame-based embeddings consisting of only spatial information into a spatio-temporal structure representing long-term context.

2.4.2 Audio temporal encoder

Like the video temporal encoder, the audio temporal encoder also aims to capture the temporal representation of audio content. It consists of a 2D ResNet34 network [Chung et al., 2020a] combined with a squeeze-and-excitation (SE) module [Hu et al., 2018].

The audio is initially broken down into audio frames (in the range of tens of milliseconds), and each audio frame is represented by a vector of Mel-frequency cepstrum coefficients (MFCCs).

The series of audio frame vectors is then passed to the audio temporal encoder, which generates a sequence of audio embeddings F_a as output. The ResNet34 network is designed with dilated convolutions to ensure that the time resolution of audio embeddings matches that of the visual embeddings F_v, which is needed in the subsequent attention mechanism.

2.4.3 Audio-visual cross-attention

F_a and F_v are temporal embeddings for audio and video signals representing the respective domains' speaking activities. Two cross-attention networks along the temporal dimension are then used to dynamically describe the interaction between signals from the two domains.

Figure 2.12 shows the attention layer, which is the core part of the cross-attention network. The linear layer projects the inputs as the query vectors (Q_a, Q_v), the key vectors (K_a, K_v) and the value vectors (V_a, V_v). The outputs of the respective cross-attention layers are the audio attention feature $F_{(a \to v)}$ and the visual attention feature $F_{(v \to a)}$. The network further consists of the feed-forward layer, and at the end, the residual connection and layer normalization are applied. This results in the cross-modal attention network. The outputs of the two networks are concatenated in the temporal direction.

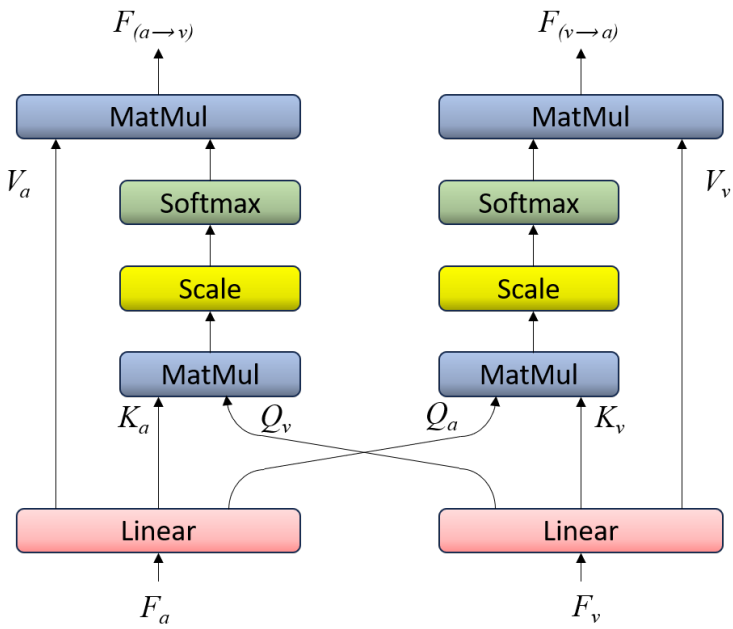


Figure 2.12: Attention layer in cross-attention network

2.4.4 Self-attention and classifier

After the cross-attention network captures the audio-visual interaction during speaking activities, a self-attention network is applied to model the temporal information at the audio-visual utterance level.

Figure 2.13 illustrates the self-attention network, which is quite similar to the cross-attention network. In this network, the query Q_{av}, key K_{av} and the value V_{av} inputs to the attention layer all come from the joint audio-visual feature F_{av}. The self-attention network helps distinguishing between the speaking and non-speaking frames.

2.4.5 Performance

TalkNet-ASD is evaluated on the AVA-ActiveSpeaker dataset [Roth et al., 2020]. Table 2.5 compares the performance of TalkNet with state-of-the-art on the validation set of the dataset in terms of the mean average precision (mAP). It achieves a 92.3% mAP value, outperforming the second-best system by 3.5%. It was also evaluated in terms of the area under the curve of ROC (AUC), as some studies use it to report their results. Table 2.6 presents the comparison of TalkNet with state-of-the-art on the validation set in terms of the AUC. It achieves an AUC of 96.8%, outperforming the second-best system by 3.6%.

Lastly, the TalkNet was also evaluated on the test set of the AVA-ActiveSpeaker dataset in terms of the mAP, and the results are presented in Table 2.7. As expected, TalkNet's 90.8% mAP outperforms best prior work of [Son Chung, 2019] by 3.0%.

2.5 Whisper

Whisper [Radford et al., 2023] is a state-of-the-art general-purpose automatic speech recognition (ASR) system. It is trained on a massive dataset of 680,000 hours of

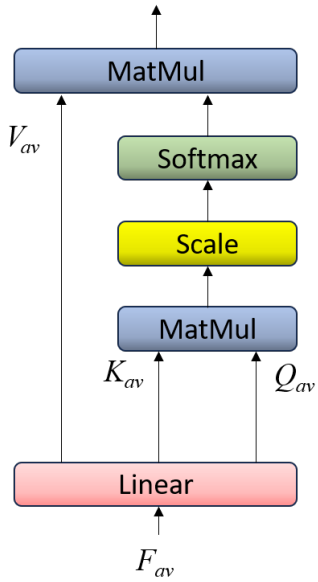


Figure 2.13: Attention layer in self-attention network

Table 2.5: Comparison with the state-of-the-art on AVA-ActiveSpeaker validation set in terms of mean average precision (mAP)

Method	mAP (%)
Roth et al. [2020]	79.2
Zhang et al. [2019]	84.0
MAAS-LAN Alcázar et al. [2021]	85.1
Active Speakers Context Alcázar et al. [2020]	87.1
Son Chung [2019]	87.8
MAAS-TAN Alcázar et al. [2021]	88.8
TalkNet	**92.3**

*adapted from [Tao et al., 2021]

Table 2.6: Comparison with the state-of-the-art on the AVA-ActiveSpeaker validation set in terms of area under the curve (AUC)

Method	AUC (%)
Sharma et al. [2020]	82.0
Roth et al. [2020]	92.0
Huang and Koishida [2020]	93.2
TalkNet	**96.8**

*adapted from [Tao et al., 2021]

Table 2.7: Comparison with the state-of-the-art on the AVA-ActiveSpeaker test set in terms of mean average precision (mAP)

Method	mAP (%)
Roth et al. [2020]	82.1
Zhang et al. [2019]	83.5
Active Speakers Context Alcázar et al. [2020]	86.7
Son Chung [2019]	87.8
TalkNet	**90.8**

*adapted from [Tao et al., 2021]

diverse, multilingual data collected from the web. Apart from ASR, it is also capable of performing multiple tasks related to the speech recognition problem, such as language identification, phrase-level timestamps, transcription of multilingual speech, and translation of multilingual speech to English. Additionally, the training on such a large and diverse dataset has made it extremely robust to accents, background noises, and technical language. Figure 2.14 presents the whisper model architecture.

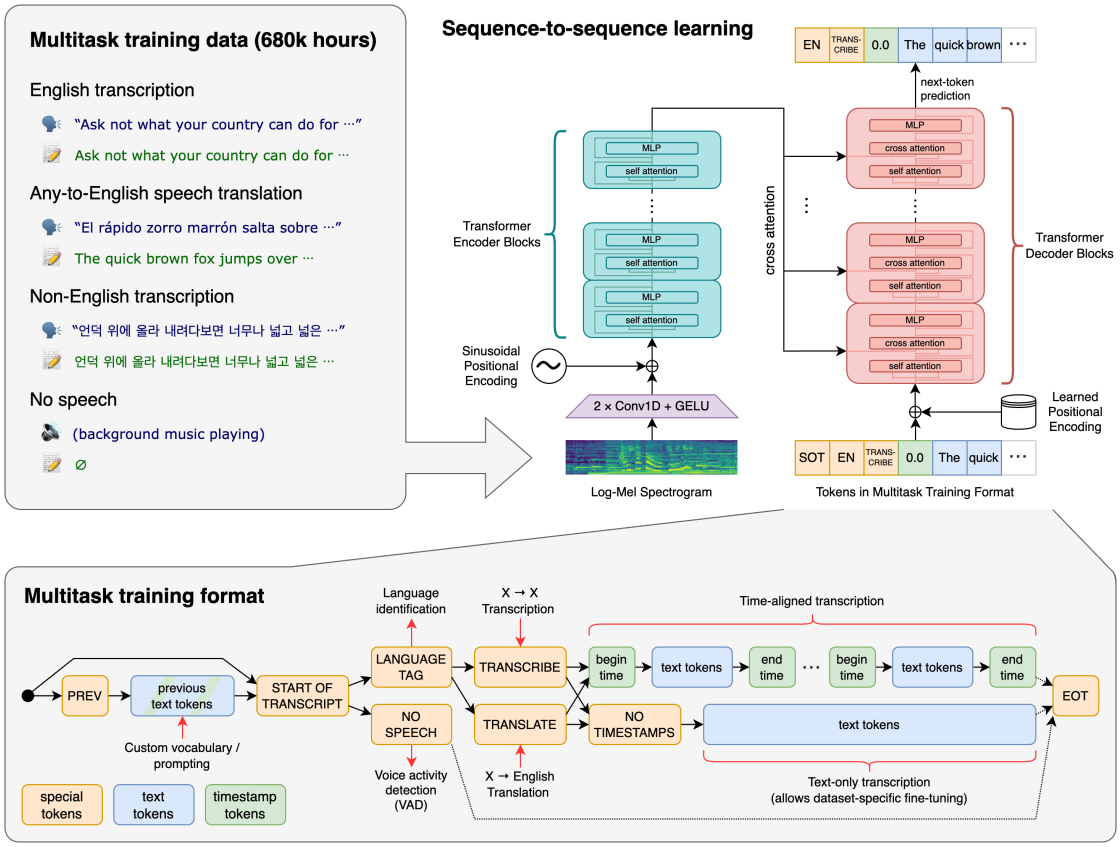


Figure 2.14: Whisper architecture

The core part of the architecture is the well-validated and reliably scalable encoder-decoder transformer [Vaswani et al., 2017]. The input audio is re-sampled at 16,000 Hz and divided into 25 millisecond windows with a stride of 10 milliseconds. These

partitions are transformed into an 80-channel log-magnitude Mel spectrogram representation and fed to the network. The input is initially processed through a small stem containing two convolution layers and the GELU activation function [Hendrycks and Gimpel, 2016], after which sinusoidal position embeddings are added to the output. Pre-activation residual blocks [Child et al., 2019] are employed in the transformer encoder block, and the encoder output is subjected to a final layer normalization. An additional token is supplied on the decoder side, directing the model to perform the specified task. The decoder block uses the token and the learned positional embeddings to produce the final output.

As the Whisper model was not fine-tuned to a specific dataset, such as Librispeech [Panayotov et al., 2015], a popular speech recognition dataset, it does not perform better than other models on such datasets. However, this also makes it much more robust and achieves a much better zero-shot performance on many diverse datasets, making as much as 50% fewer errors.

2.6 Face detection and recognition

Face detection and recognition are crucial tasks in computer vision and machine learning, due to their applications in a number of domains such as security and law enforcement, health, entertainment, education, and marketing [Taskiran et al., 2020]. The face is one of the main biometric traits that carries information about a person's identity and, hence, is a great way of distinguishing one person from another.

Face detection is the process of determining the size and location of a human face in an image. Face recognition, on the other hand, is the process of determining or verifying a person's identity by comparing it against a database of known faces. The proposed implementation uses separate models to perform both of these tasks. The following subsections highlight these models and the qualities that make them suitable for the respective tasks.

2.6.1 Single shot scale-invariant face detector

Single shot scale-invariant face detector (S3FD) [Zhang et al., 2017] is a face detection framework designed specially designed to handle faces of varying sizes within images. The main drawback of traditional anchor-based frameworks was that their performance dropped significantly as the faces became smaller [Huang et al., 2017]. To overcome this, S3FD introduced a network architecture with a wide range of anchor-based layers. The stride size of each layer doubles gradually from 4 to 128 pixels, ensuring that faces of different scales have adequate features for detection at the corresponding anchor-associated layers.

The framework is trained on 12,880 images of the WIDERFACE [Yang et al., 2016] training set. Table 2.8 shows the evaluation results of the framework on the WIDERFACE test set as compared to the baseline frameworks RPN-face [Ren et al., 2015] and SSD-face [Liu et al., 2016]. The creators also divided the test set into three subsets based on the level of difficulty of detection: 'Easy', 'Medium' and 'Hard'. The S3FD outperforms both the baselines on all three subsets. It performs significantly better on the hard subset, achieving almost 20% higher mAP as compared to the baselines.

Methods	Subsets		
	Easy	Medium	Hard
RPN-face	91.0	88.2	73.7
SSD-face	92.1	89.5	71.6
S3FD	**93.7**	**92.4**	**85.2**

Table 2.8: Comparison of S3FD with the state-of-the-art on the WIDERFACE test set in terms of mean average precision (mAP)

*adapted from Zhang et al. [2017]

Apart from its impressive face detection capability on faces of varying sizes, it is designed to have extremely low inference time, achieving a speed of 36FPS on VGA-resolution images with a single GPU. These qualities make it an excellent choice to perform face detection on video frames, as the videos can have faces of varying sizes, and the number of frames could easily reach 1000s of frames even for a 40-second video at 25FPS.

2.6.2 Dlib-based face_recognition model

Despite S3FD's impressive performance on size-invariant faces, it is not designed to perform face recognition. In the proposed implementation, face recognition is an important component as it groups together speakers from visuals and also determines the number of speakers in the video. Hence, a secondary model is needed that can perform robust face recognition.

The face_recognition[4] is a Python package that provides one of the most extensive facial recognition libraries. It was built using the state-of-the-art face recognition deep learning framework by lib [King, 2009], and it is extremely simple to use with Python or via the command line. The model achieves an accuracy of 99.38% on the benchmark Labeled Faces in the Wild [Learned-Miller et al., 2016] dataset.

[4] https://github.com/ageitgey/face_recognition

3. Related Work

A review of recent works in speaker diarization and active speaker detection was performed to provide context for the proposed system presented in this thesis. This chapter summarizes the traditional approaches, recent advances, and specific challenges addressed by this research. It also establishes the current state-of-the-art in speaker diarization and active speaker detection. The chapter is outlined as follows:

- Initially, Section 3.1 emphasizes the developments in the field of audio-only speaker diarization in the past couple of decades. It mentions how the speaker identity discriminative features evolved in this time and, subsequently, the emergence of the different types of speaker diarization systems.
- Section 3.2 follows up with the discussion about the limited development of audio-visual speaker diarization systems.
- Section 3.3 discusses the emergence of active speaker detection systems in recent times.
- Lastly, Section 3.4 also mentions some of the joint speaker diarization (SD)-automatic speech recognition (ASR) models developed recently.

3.1 Developments in audio-only speaker diarization systems

Audio-only speaker diarization systems have seen significant progress in the last couple of decades. The primary accelerator for this is the regular emergence of newer and better ways to represent speaker identity discriminative features over time. Hence, it is also important to focus on the development of these features before diving into the speaker diarization systems. Section 3.1.1 discusses these developments in detail. Post that, Section 3.1.2 dives deeper into the development of the speaker diarization systems based on these features.

3.1.1 Speaker identity discriminative features

During the early years of the last decade, i-vector [Dehak et al., 2011] was found to be an effective speaker-id discriminative feature. i-vectors are compact representations of speaker characteristics computed for each speaker utterance. Dehak et al. [2011] proposed to use it in combination with linear discriminant analysis (LDA) and within-class covariance normalization (WCNN) [Hatch et al., 2006] to compensate for channel-dependent variability, and the i-vector distances were defined with cosine similarity or a support vector machine (SVM). Matějka et al. [2011] showed that the performance of i-vectors can be further improved using probabilistic LDA (PLDA). This combination of i-vectors and PLDA proved to be extremely effective and, hence, was a popular technique for speaker verification and diarization until the emergence of deep learning-based approaches.

With the advent of deep learning and the increasing availability of annotated data, it was possible to exploit DNNs to obtain speaker-discriminative features in place of GMMs. Variani et al. [2014] was one of the first works in this direction, that showed that DNN-based features(so-called d-vectors) were able to outperform i-vectors, especially in noisy conditions. The DNN in Variani et al. [2014] was trained on large corpora with fixed-length segments, a vast number of speakers and varying acoustic conditions. It was a multi-class classification problem where the DNN had to assign the correct speaker to the segment among all the speakers present in the training set. The output of the last hidden layer was the d-vector that can be used as a speaker discriminative feature vector. Follow-up works continued to improve the DNN-based feature vectors, and a major advancement came with the introduction of the x-vector extractor [Snyder et al., 2018]. It utilizes a time-delay neural network (TDNN) and a statistical pooling layer to create a compact speaker-id representation.

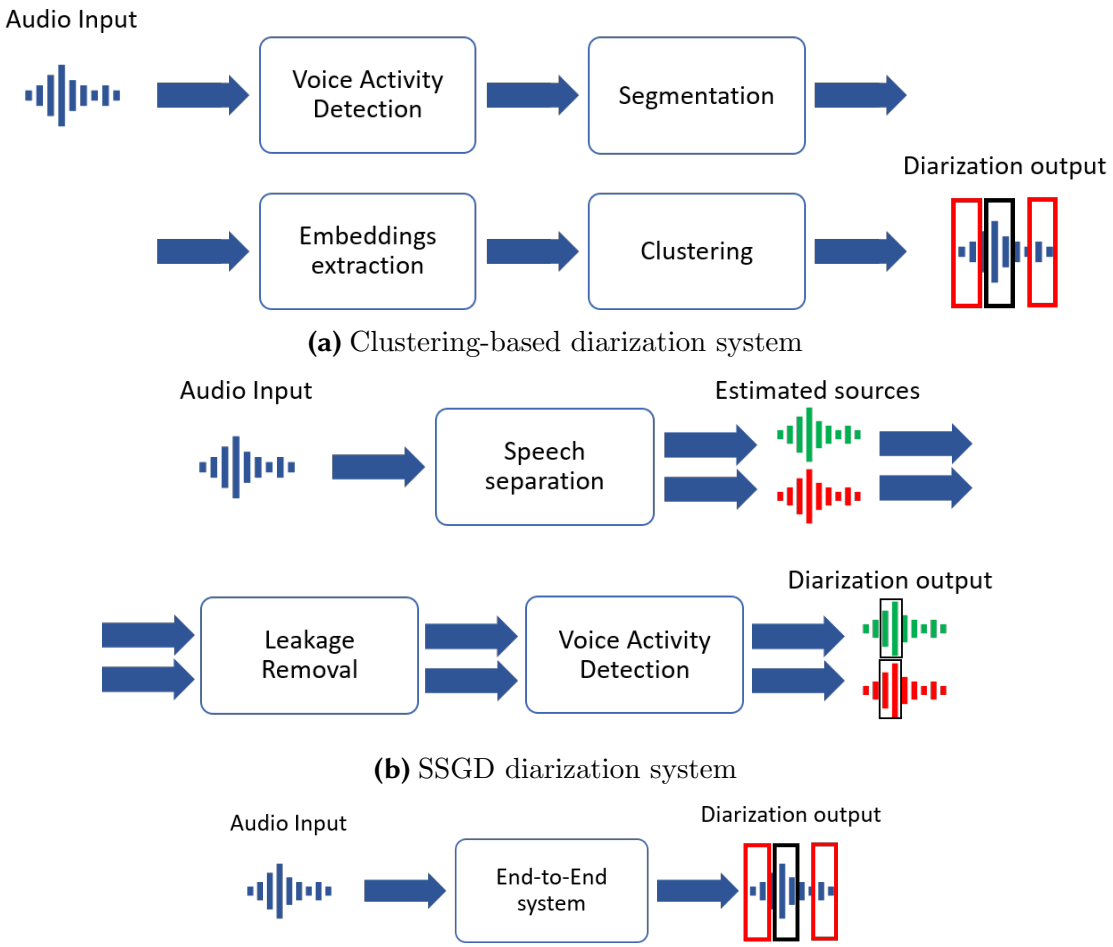
3.1.2 Recent speaker diarization systems

The diarization systems in recent times can be broadly categorized into three groups based on their design and implementation: clustering-based, end-to-end, and speech separation-guided diarization (SSGD) methods. Figure 3.1 shows the general architectures of these categories. These systems employ different approaches to perform speaker diarization, which are further explained in the following subsections, along with recent works that employed them.

3.1.2.1 Clustering-based methods

Clustering-based methods have been the most common approach for speaker diarization for several decades, and even today, they continue to be relevant and competitive. These methods generally comprise four modules as demonstrated in Figure 3.1(a)- VAD, segmentation, speaker embedding extraction and clustering.

A notable work in this category is the variational Bayesian-hidden Markov model clustering of x-vector sequences (VBx) by Landini et al. [2022b], which integrates variational Bayesian resegmentation [Diez et al., 2020] with x-vector clustering diarization. It proved to be a highly effective diarization technique, evidenced by its widespread use in popular diarization challenges such as DIHARD [Ryant et al., 2020].



(a) Clustering-based diarization system

(b) SSGD diarization system

(c) End-to-end diarization approach

Figure 3.1: General architectures of different categories of diarization systems

VBx produces an initial diarization hypothesis by employing a standard clustering-based diarization pipeline, combining x-vectors and PLDA+AHC clustering, and further refines it via a Bayesian hidden Markov model (BHMM). A limitation of this approach is its inability to model overlapped speech, as it employs HMM, and each HMM state is only associated with a single speaker.

Spectral clustering was another popular technique, and earlier speaker diarization works were mostly based on the NJW algorithm [Ng et al., 2001]. In this case, the speaker clusters were estimated using a similarity matrix constructed from speaker embeddings. However, this technique suffered with limited accuracy as it is very sensitive to noise in the similarity matrix due to intra-speaker variability. Recent works such as Wang et al. [2018] have tried to overcome this issue by introducing a scaling parameter that assigns different weights to pairwise similarities. In place of the popular PLDA+AHC pipeline, Wang et al. [2018] implements spectral clustering with maximum eigengap using DNN-based speaker embeddings (d-vectors). In this case, the hyperparameters had to be tuned on development sets, increasing its computational costs and making generalization more challenging. Park et al. [2020] proposes an auto-tuning strategy to overcome this issue, eliminating the need for a development set.

3.1.2.2 Speech separation guided diarization

SSGD-based methods have gained more attention recently due to the increasing effectiveness of DNN-based speech separation methods. Figure 3.1(b) illustrates a classical SSGD pipeline.

Fang et al. [2021] employs such a pipeline where the input conversational audio is first fed to a speech separation module, which separates the audio streams of the individual speakers. A conventional VAD is then applied to the individual streams to estimate speech boundaries, resulting in diarization annotations. Morrone et al. [2023] further builds upon this by adding a leakage removal post-processing module to reduce false alarms that occur due to imperfect separation.

In SSGD, the most computationally expensive step is speech separation, so the diarization process becomes extremely simple as a basic energy-based VAD can easily achieve decent performance, as shown by Morrone et al. [2023].

A major drawback with SSGD-based systems is that they require prior knowledge of the maximum number of speakers in a conversation.

3.1.2.3 End-to-end diarization approach

The recent advancements in deep learning also gave rise to end-to-end diarization systems that can perform all the steps of a speaker diarization system with a single neural network. A general architecture of an end-to-end approach is demonstrated in Figure 3.1(c).

A novel framework known as the end-to-end neural diarization (EEND) was proposed [Fujita et al., 2019a] [Fujita et al., 2019b] in this direction. Initially, Fujita et al. [2019a] proposed an EEND with a bidirectional long short-term memory (BLSTM) network, which was then extended to the self-attention-based network SA-EEND by Fujita et al. [2019b]. SA-EEND formulates the diarization task as a multi-label classification problem where a sequence of acoustic features are fed as input, and the output is the joint speech activities of multiple speakers.

End-to-end architectures hold multiple advantages over their clustering-based counterparts. Clustering-based methods consist of multiple submodules that need to be separately trained, whereas the end-to-end systems just consist of a single model that needs to be trained. Moreover, the model is also directly optimized to maximize diarization accuracy, resulting in a high accuracy. Also, traditionally, clustering-based architectures struggle to deal with overlapping speech, as the speaker embedding extractors are trained for speaker verification on single-speaker segments. On the contrary, end-to-end methods can be trained on datasets containing real conversations and hence can naturally learn to handle overlapping speech. However, this can also be considered a weakness as Fujita et al. [2019a] suggested that EEND tends to overfit on the distribution of speakers and overlap ratio in the training data. The cost of annotating real-world conversations is another major obstacle to adopting end-to-end methods [Landini et al., 2022a].

3.2 Developments in audio-visual speaker diarization

Historically, speaker diarization frameworks have predominantly used only audio streams to perform speaker diarization. However, a few audio-visual speaker diarization systems have also emerged recently. In complex environments, e.g., ones containing a large number of speakers and highly overlapped speech, speaker diarization is still a challenging prospect. In such cases, using a complimentary modality such as visual information is a promising way to improve the diarization results.

Gebru et al. [2018] proposed to combine visual tracking with multiple speech source localization to solve the problem of speech-to-person association. It uses audio input from two microphones to extract binaural spectral features and combines them with visual information using a semi-supervised alignment technique. Essentially, it uses positional information and relative audio variations to combine speakers in audio and visual modalities. Such an approach might fail for videos with complex scenarios and a lack of dual microphone recorded audio. Chung et al. [2019] employs an "enrol first, diarize later" approach, where an audio-visual synchronisation network first iterates over the videos to find instances of a clear correlation between audio and speaking faces. These instances are then used to generate user profiles that are used to perform speaker diarization. However, this approach is difficult to generalize in a realistic scenario for unseen videos because it is hard to collect ground-truth speaker profiles for such videos. Moreover, both of these methods are not well suited for off-screen speakers.

Xu et al. [2022] introduced the AVA audio-visual diarization (AVA-AVD) dataset made up of diverse movie clips carefully selected from the AVA-ActiveSpeaker dataset [Roth et al., 2020]. They also proposed a multi-stage audio-visual speaker diarization system which combines a voice activity detection (VAD) stage to detect speech segments and face inputs to learn the audio-visual correlation between speakers' faces and voices. The proposed network called the AVR-Net, jointly scores the speech-face pairs, and clustering is performed to group different speaker identities together. However, this approach is not completely autonomous, as the model needs manual annotations of faces in the videos to learn the correlation. Chung et al. [2020b] introduced another large-scale audio-visual diarization dataset called VoxConverse collected from "in the wild" videos. To create the dataset, they also proposed a semi-automatic audio-visual diarization method with a combination of audio-visual active speaker detection and self-enrolled speaker models for speaker verification. The proposed implementation also delves into the idea of using active speaker detection as an initial step to perform video-based speaker diarization.

3.3 Emergence of active speaker detection

The introduction of the first large-scale video dataset for active speaker detection (ASD), AVA-ActiveSpeaker [Xu et al., 2022], resulted in the emergence of novel approaches to perform ASD using both audio and visual modalities.

Zhang et al. [2019] proposed an approach in the direction of multi-modal ASD. They used a 3D-ResNet18 model for visual feature extraction and a modified VGG-M network to extract audio features. Their outputs were fed to a two-layer bidirectional Gated Recurrent Unit (GRU) to obtain the final frame-wise predictions. Son

Chung [2019] proposed a front-end architecture for audio and visual encoders and a bidirectional LSTM-based backend classifier. Two separate convolutional neural networks were used to generate audio and visual encodings using the video frames and MFCC representation of audio streams. In the backend, the encodings were fed to two separate bidirectional LSTM networks, and their outputs were concatenated. A linear classification layer predicts whether the person is speaking or not.

Alcázar et al. [2021] leverages graph convolutional networks (GCNs) to model the interaction between different modalities to perform ASD. It first uses a local assignation network (LAN) that maps the detected audio at frame level to the most likely speaker in the frame. Then, a temporal assignation network is used, extending LAN over a temporal window to find connections between audio/visual features from adjacent timesteps, allowing temporal consistencies in audio and video modality predictions.

Tao et al. [2021] achieved significant success in multi-model ASD by taking both short-term and long-term features into consideration. It consists of audio and visual temporal encoders for feature representation. The intermodality interaction is captured by an audio-visual cross-attention mechanism, and the long-term dependencies are captured by a self-attention mechanism. As discussed previously in Section 2.4, this approach significantly outperformed the above-mentioned approaches, and has been used as the ASD component for the proposed implementation.

3.4 Joint ASR-SD models

Apart from the traditional approach of generating speaker-attributed transcriptions by combining independent outputs from SD and ASR systems, joint ASR-SD have also been proposed in recent times. These works are generally end-to-end models that perform all the tasks with a single model.

Shafey et al. [2019] proposed a joint ASR-SD using a recurrent neural network transducer (RNN-T). It utilizes both linguistic and acoustic cues to infer speaker roles, and it was shown to perform better than the baseline, which was derived by combining an ASR and SD model independently. Another work by Kanda et al. [2020] proposed a model that unifies speaker counting, speech recognition and speaker identification for monaural overlapped speech. The model uses an attention-based encoder-decoder architecture built on serialized output training (SOT). It extends the SOT with the introduction of an auxiliary input in the form of speaker inventory to produce speaker labels and multi-speaker transcriptions. Kanda et al. [2022] proposed another end-to-end speaker-attributed ASR model, that consists of two attention-based encoder-decoders (AED), one for ASR and another for speaker identification. The ASR encoder, ASR decoder and the speaker decoder use a transformer-based network architecture, while the speaker encoder is based on Res2Net [Gao et al., 2021].

All of these works are based on audio-only and do not take into account any additional modalities, such as visuals from videos to perform speaker-attributed ASR. Additionally, while such end-to-end models take advantage of the relationship between the linguistics and acoustic cues of audio, they also suffer from the same problem

discussed earlier in relation to the end-to-end speaker diarization models. The cost of annotating real-world conversations is a major obstacle for such models [Landini et al., 2022a]. Also, such models may not generalize well on unknown data as they tend to overfit on certain aspects of the training data, such as the distribution of speakers and overlap ratio [Fujita et al., 2019a]. Lastly, the modularized approach, which consists of separate SD and ASR modules, can take advantage of the advancements in individual modules, such as the introduction of state-of-the-art speech models such as Whisper, which is not possible with an end-to-end approach.

4. Implementation

Chapter 2 provided a comprehensive overview of the key concepts fundamental to this work. Subsequently, Chapter 3 established the state of the art for the domains directly related to this work.

This chapter describes the complete implementation of the automated speaker attribution pipeline proposed in this research. The chapter is outlined as follows:

- Section 4.1 describes the overview of the entire pipeline to establish an overall perspective. The individual modules are discussed in more detail in the subsequent sections.
- Section 4.2 introduces the implementation of the audio-only speaker diarization (AOSD) module. It highlights the model used to perform diarization and its corresponding output.
- Section 4.3 discusses the design of the video-based speaker diarization (VBSD) module. The implementation of the submodules active speaker detection (ASD) and face recognition and clustering are also described.
- Section 4.4 presents the implementation of the automatic speech recognition (ASR) model Whisper in the pipeline.
- Lastly, Section 4.5 illustrates the rule-based algorithm that combines outputs from the three modules described in the previous sections to produce the final results.

4.1 Pipeline overview

Figure 4.1 represents the general flow of the pipeline. It takes a video file as input. The pipeline has 3 major components - an AOSD module, a VBSD module that is a combination of active speaker detection and face clustering, and the Whisper-ASR, which generates the transcriptions, along with word-level timestamps. These three components work independently, and the output from these modules is then combined using a rule-based algorithm that generates the speaker-attributed transcriptions as the final output.

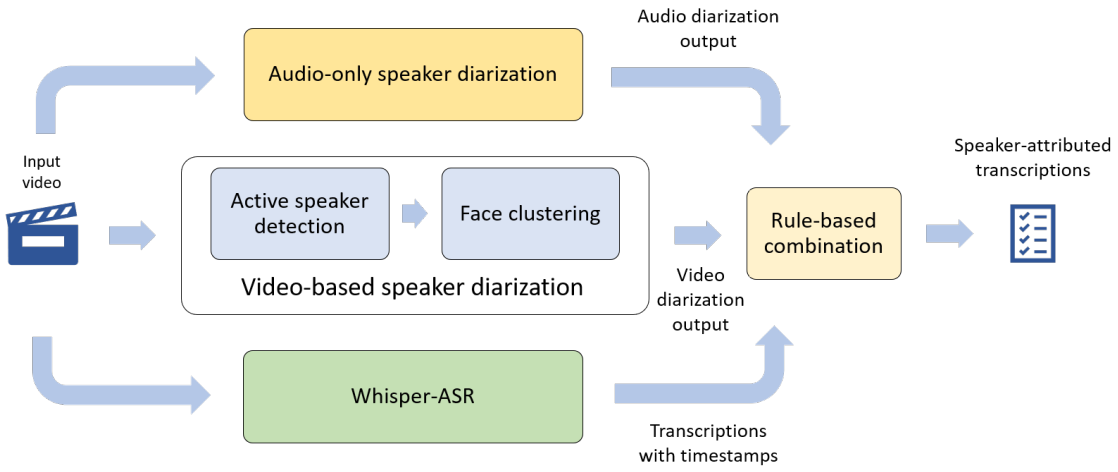


Figure 4.1: Proposed pipeline overview

4.2 Audio-only speaker diarization

In this module, the audio component of the video is extracted and then a speaker diarization pipeline is applied to it to generate the audio diarization output. For the purpose of this implementation, a pre-trained pipeline speaker-diarization-3.0[5] based on pyannote.audio [Plaquet and Bredin, 2023] framework is used. The pipeline has been shown to achieve state-of-the-art performance on several datasets. The design and performance of the pipeline, as well as the framework, have been described in detail in Section 2.3 of this thesis.

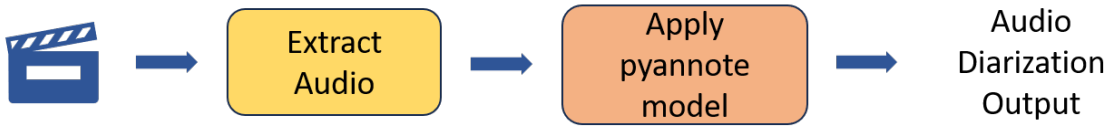


Figure 4.2: Design of the audio-only speaker diarization module

The pipeline can be easily applied to audio files. Figure 4.2 presents the steps taken to perform the audio-based speaker diarization. To perform diarization by applying the pipeline, audio is extracted from the video using ffmpeg. This generates a wav file, and the model is applied to this file to produce the diarization result.

4.3 Design of the video-based speaker diarization module

The video-based part of the pipeline can be divided into two submodules: ASD and face clustering. The ASD part of this module helps identify whether the face appearing on the screen is speaking. Once all the speaking faces in a video are identified, face recognition and clustering are used to group faces belonging to the same person. The output of these two steps produces a diarization output that only considers on-screen speakers. The design of these submodules is discussed in the following subsections.

[5] https://huggingface.co/pyannote/speaker-diarization-3.0

4.3.1 Applying active speaker detection

For ASD, the pre-trained TalkNet-ASD [Tao et al., 2021] model has been used. The model is already introduced and described in more detail in Section 2.4.

TalkNet-ASD works by interpreting short-term and long-term audio and visual information and audio-visual interaction. For a successful application, it must be applied on continuous clips of an on-screen person. Hence, the video needs to be cropped and transformed to a format that the model can consume. Figure 4.3 presents the steps implemented to achieve this.

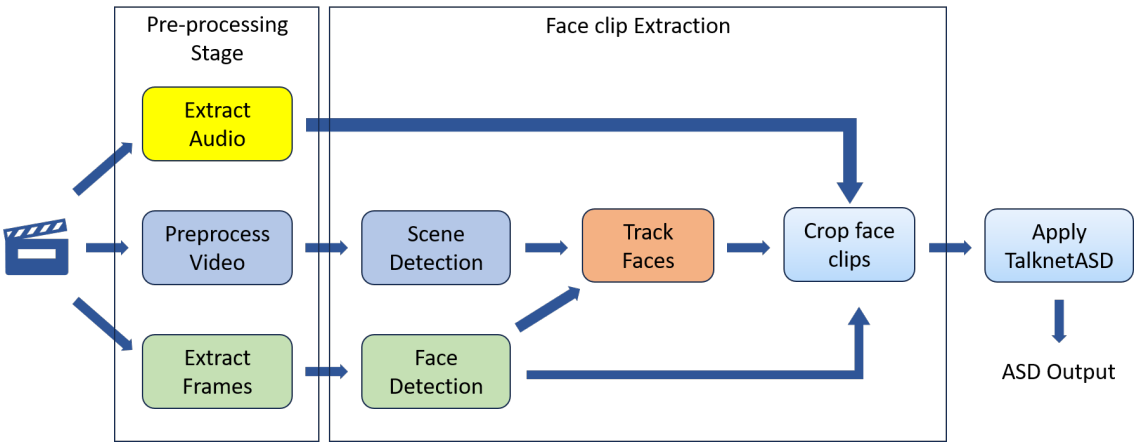


Figure 4.3: Implementation of TalkNet-ASD

Before applying TalkNet-ASD, an input video is processed in 2 stages. The first stage is the pre-processing stage. To make processing easier and more efficient, all videos are converted into the same format of 25fps avi files. This makes the number of frames more manageable without a significant loss of information. It also makes all videos consistent so that the later parts of the pipeline do not have to cater to a large number of video formats. Similarly, the audio is extracted from this video as a 16000 Hz mono wav file. Lastly, the video is separated into individual frames, and the extracted frames are stored for further processing.

The second stage is the face clip extraction stage. When a person appears in a video, they appear for a continuous set of frames. The start and end frames of a person appearing in a video are important to identify in order to extract face clips from the video. The way this happens can be different in different types of videos. In some cases, the video may contain clips from a single camera that move around, covering the face of a person for some time before moving on, and the face is no longer visible in the frames. The opposite can also happen when a person arrives in the frame and then walks away in front of a stationary camera. Figure 4.4 shows an example of such a scene. In other cases, the video may contain scene cuts, where instead of a single camera being moved, the video contains clips from multiple cameras merged together, so there could be drastic and sudden changes in the video from one frame to another. Figure 4.5 demonstrates an example of such a scene. Both of these examples have frames extracted from a single video clip. Moreover, it is quite common these days that video clips generally contain a combination of both of these cases. Most of the

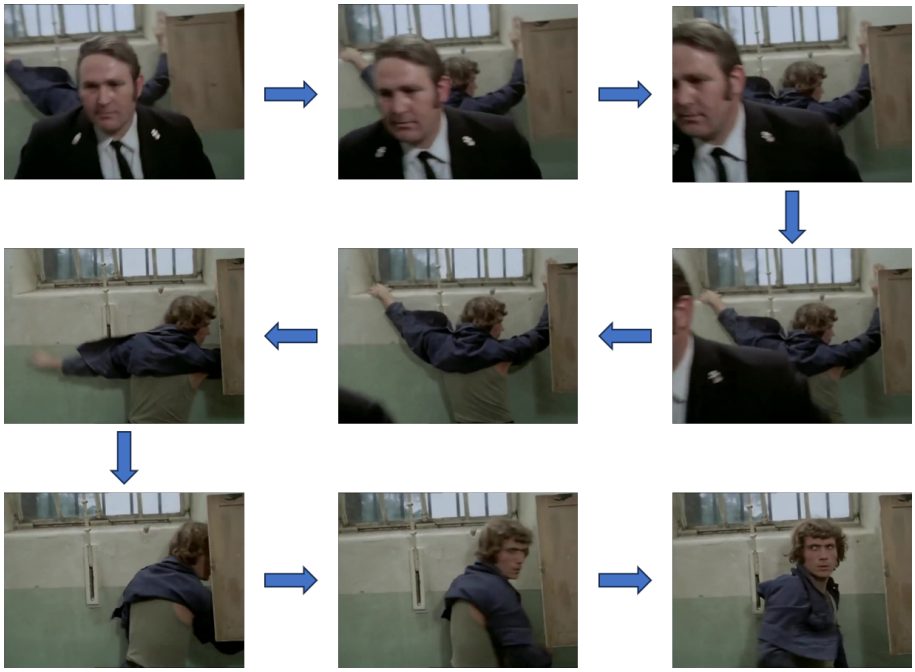


Figure 4.4: Example of a change of on-screen face in front of a single stationary camera

videos from sources such as movies and news channels are a combination of both of these methods.

Both of these scenarios are being handled in a systematic manner in this pipeline. Initially, the scene cuts are detected. Python-based PySceneDetect[6] is used for this purpose, which can not only detect scene cuts but also split the videos into separate clips based on the detected scene cuts. This ensures that there are no shot changes in the separated video clips, and these clips only contain continuous scenes. Consecutively, face detection is applied on all the frames. The bounding boxes of the faces appearing in each frame are extracted and stored. For face detection, S3FD [Zhang et al., 2017] has been used. S3FD has been shown to achieve superior performance on varying scales of faces, especially on small faces. This is a major advantage when performing face detection on videos, as the distance between the camera and the person varies quite often in videos. Another advantage of using S3FD is its processing speed, as it is shown that it can run at 36 FPS with a single GPU. This is an important factor as the number of frames in a video can be quite high depending on the length of the video.

The detected scenes and faces are used to create a face track, which is a cropped clip of a person's face from a continuous shot. To identify and track individual faces appearing in a clip, the Intersection over Union (IoU) metric is used between the face positions in consecutive frames. IoU is an evaluation metric primarily used to measure the accuracy of object detection. It is a ratio between the region of overlap and the region of union. The higher the overlap, the higher the IoU value. If the change of face position between consecutive frames results in an IoU value less than a predefined threshold, they are not considered the continuous shot of the same

[6] https://www.scenedetect.com/

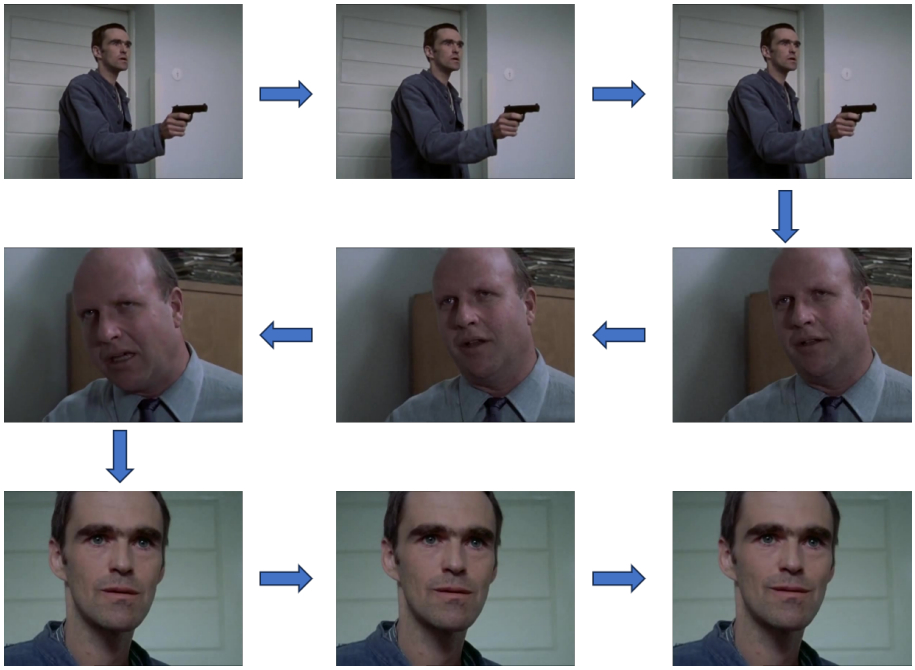


Figure 4.5: Example of a change of on-screen face due to merging of clips from multiple cameras

face. For object detection-related problems in computed vision, a threshold of 0.5 is considered good [Everingham et al., 2010]. In the case of videos, this threshold allows for slight changes in face positions in consecutive frames, which results from a person's movement while talking. This ensures that the face tracks are not cut too short due to minor positional changes.

Once the tracks are marked, the faces are cropped from the frames and the corresponding audio is clipped from the main audio file to create face tracks. Each face track represents a continuous shot of a single person along with the corresponding audio during that part of the video.

Now, the TalkNet-ASD can be applied to the face tracks, which will use the face in the clips and the corresponding audio to predict whether the person is speaking or not, and also to predict during which frames the person is speaking.

4.3.2 Face recognition and clustering

The face clips represent only a small section of the input video clip. Even though it can be concluded that a person appearing throughout a single face clip is the same person, the entire input video can produce a number of such face clips depending on the number of shot changes and the variance of face positions in each clip. Hence, the next step is to determine all the clips belonging to the same person.

Figure 4.6 demonstrates the face recognition and clustering part of the pipeline. Initially, the talking faces recognised by the ASD step are converted into embeddings that will be used to perform clustering. Even though in the previous stage, S3FD was used for face detection, it is not capable of generating face embeddings. Hence, a different model was selected to generate the embeddings. For this purpose, the

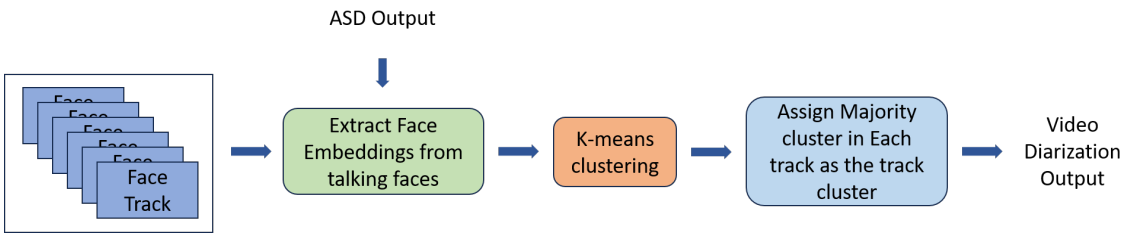


Figure 4.6: Implementation of face recognition and clustering

face_recognition[7] Python package is used, which is designed using state-of-the-art Dlib [King, 2009] face recognition framework and has been shown to achieve an accuracy of 99.38% on Labeled faces in the wild [Learned-Miller et al., 2016] benchmark dataset. As the bounding boxes were already provided by the S3FD in the previous stage, the face recognition can be directly applied to the talking faces, and a list of 128-dimensional face embeddings is generated.

Next, K-means is applied to the list of embeddings to determine the faces belonging to the same person. As each video can have a varying number of speaking faces, the optimum value of k changes from video to video. To determine the optimum value of k for a particular video, k means is applied with values ranging from 2-10, and the silhouette coefficient is computed to determine the best candidate. The value with the highest silhouette coefficient is selected as the optimum k-value for the particular video.

The clustering process may not result in perfect cluster assignment, as there might be some noise points that could drift into other clusters. As the videos are free-flowing, the faces appearing in the frame may not generate the most appropriate embeddings due to variations in angles and face sizes, which could result in incorrect cluster assignments. Additional information can be used to minimize such mistakes. As mentioned in the previous section, all the faces appearing in a face track belong to the same person; hence, for each track, the majority cluster is chosen as the track cluster.

Finally, the combination of the detected talking faces in the tracks and clustering results provides the intervals during which every person appearing in the video is speaking. This produces the video diarization output from this stage.

4.4 Whisper-ASR

OpenAI's Whisper model [Radford et al., 2023] has been used in this pipeline to transcribe videos. However, it can be replaced with other speech recognition models as long as they can generate the transcriptions along with word level time-stamps, as those are key in determining the respective word speakers. Whisper provides an easy-to-use Python API that can be applied to any audio or video file to generate transcriptions. It also provides a range of English and multi-lingual models that can be used based on the input video languages and available system resources. As all the videos present in this dataset contain English speakers, the "medium.en" and "large" models have been used to transcribe the videos.

[7] https://github.com/ageitgey/face_recognition

4.5 Rule-based combination

Once the pipeline has produced the three required results in the forms of audio and video-based diarization results and ASR-based transcripts, the next step is to use the diarization results to assign a speaker to each word in the transcript. For this purpose, a simple rule-based algorithm has been developed in this implementation. The algorithm has been summarized in the Algorithm 1.

Algorithm 1 Word-speaker assignment algorithm

```
procedure AssignWordsToSpeakers(AudioSpeakers, VideoSpeakers, Transcripts)
    Mappings ← FindSpeakerMappings(AudioSpeakers, VideoSpeakers)
    for each word w in Transcripts do
        AssignedSpeaker ← FindSpeaker(w, VideoSpeakers)
        if AssignedSpeaker is not null then
            Assign w to AssignedSpeaker
        else
            AudioSpeaker ← FindSpeaker(w, AudioSpeakers)
            if AudioSpeaker is not null then
                AssignedSpeaker ← FromMapping(AudioSpeaker, Mappings)
                Assign w to AssignedSpeaker
            else
                Mark w as Unassigned
            end if
        end if
    end for
end procedure
```

As the speakers in both the audio-only and video-based results could be different, the first step is to find a mapping between the two results. To compute this, the speaking intervals for each speaker in the audio-only result are used to find the most overlapping speaker in the video-based result. This mapping provides a link between speakers in the audio-only results to the video-based results. If a speaker in the audio-only result could not be mapped to any speaker in the video-based result, they are marked as a separate unknown speaker. While matching the words to the speaker, the first preference is given to the video-based diarization results, meaning that if a person is speaking on screen, the words are assigned to that person. If no on-screen person is present during the word utterance interval, the audio speaker is used in conjunction with the audio-video mapping to find the respective video speaker. Lastly, if both the audio and video-based results could not yield any results, the word is marked as unassigned.

To further explain this in simpler terms with an example, consider a part of the transcript generated for a video from the dataset in Table 4.1 with the respective word intervals. It consists of 2 phrases uttered between time intervals $(30.8, 33.36)$ and $(37.58, 41.12)$ respectively. The audio and video-based diarization results for the video are also shown in Figure 4.7 with the two intervals highlighted in green and

Table 4.1: Part of Whisper-generated transcript from a sample output

Word	Word Start	Word End
We	30.8	30.8
have	30.8	30.96
astrophysicist	30.96	31.7
and	31.7	31.86
author	31.86	32.24
Neil	32.24	32.54
deGrasse	32.54	32.8
Tyson	32.86	33.18
here	33.2	33.36
Why	37.58	37.82
is	37.82	37.96
this	37.96	38.12
landing	38.12	38.44
so	38.44	38.72
close	38.72	38.92
to	38.92	39.08
the	39.08	39.22
moon's	39.22	39.7
South	39.7	39.92
Pole	40.02	40.22
so	40.22	40.52
significant?	40.52	41.12

Audio Speaker	Video Speaker
SPEAKER_00	SPEAKER_03
SPEAKER_01	SPEAKER_04
SPEAKER_02	SPEAKER_00
SPEAKER_03	SPEAKER_02

Table 4.2: Audio-visual speaker mapping example

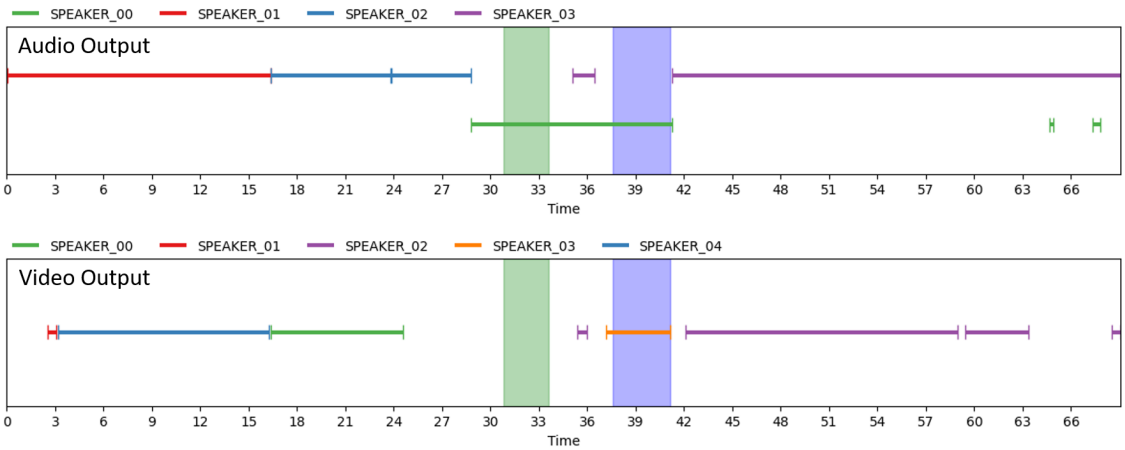


Figure 4.7: Audio-only and video-based diarization results for a sample output

blue colours, respectively. The speaker mapping between audio result speakers and video result speakers shown in Table 4.2 can be concluded based on the visuals.

As Figure 4.7 indicates, each word in phrase 2 can be mapped to SPEAKER_03 in the video diarization results. However, the words in phrase 1 cannot be mapped to any speaker in the video results. Although the audio diarization result indicates that the words correspond to SPEAKER_00, a quick lookup in the mapping proves that the words from phrase 2 also belong to the SPEAKER_03 from the video diarization result. Hence, both phrases will be assigned to the same speaker.

5. Experiment

The previous section presented the implementation of the pipeline in detail. This section elaborates on the experiment to evaluate the performance of the pipeline. In particular, the dataset used for testing and the evaluation measure are described in the following sections.

5.1 Dataset

To prove the effectiveness of the pipeline, it needs to be evaluated on a transcribed video dataset. However, due to the peculiar nature of this research, such a dataset was not readily available. Existing datasets are more suited towards related domains such as speech recognition, speaker diarization, or speaker detection.

Popular speech recognition datasets such as Librispeech [Panayotov et al., 2015] and The People's Speech [Galvez et al., 2021] consist of 1000s of hours of audio recordings and transcribed speech and can be used for applications such as training automatic speech recognition (ASR) models. While these datasets do contain transcriptions, and in some cases, even with word-level timestamps, the focus of this research involves the analysis of the relationship between audio and visuals in a video file and using them to perform speaker attributions of transcriptions. Hence, such datasets are not suitable for this research.

On the other hand, AVA-ActiveSpeaker [Roth et al., 2020] is a benchmark dataset for active speaker detection (ASD) released by Google. It consists of about 3.65 million human-labelled frames spanning about 38.5 hours. The labels indicate whether each face instance in the frames is speaking. While this dataset has greatly helped in advancements of various audio-visual processing systems, it is also unsuitable for this research due to its lack of transcriptions and annotations of audio-only sections of the videos.

Lastly, speaker diarization datasets come close to fulfilling the requirements of this research. The AMI dataset [Carletta et al., 2006] consists of 100 hours of meeting recordings along with their transcripts and word-level timestamps. Even though it

also contains meeting videos from room-view cameras, the camera view is from the top and does not capture the faces of the participants speaking.

Hence, a new dataset was created for this research using YouTube videos of multiple people having a discussion. In most cases, clips were extracted from longer-duration videos to maximize variability and minimize redundancy. It was also made sure that the selected clips had at least some contributions from all the visible speakers and that there was a conversational scenario with multiple speaker changes so that the diarization could be effectively evaluated.

The transcripts and word timestamps were generated using Whisper [Radford et al., 2023]. During manual verification, it was noticed that, in some cases, the "medium.en" model produced better transcripts, and in other cases, the "large" model produced better results. So, both results were considered, and a combined transcript was created for each file to minimize the loss of words during transcription. Then, the words were manually annotated with speaker information. The statistics of the dataset are shown in Table 5.1. As the table indicates, the videos averaged over 3 speakers per video, with the number of speakers ranging from 2-6. The percentage of off-screen words and words with overlapping speakers are also key factors that can impact the final diarization performance. As the proposed system is a combination of audio and video diarization systems, the system will have to rely on one of the two modalities when making a decision regarding assigning such words.

The data regarding the off-screen speakers is an approximate number calculated using the TalkNet-ASD [Tao et al., 2021] model. Additionally, the talking faces identified by TalkNet-ASD [Tao et al., 2021] in each video were also annotated so that the face clustering performance could be evaluated for the videos.

Table 5.1: Dataset statistics

Metric	**In-person meeting**	**Virtual meeting**
Number of videos	15	61
Total video duration (in seconds)	2577.4	4590
Avg. video duration (in seconds)	171.8	75.2
Avg. words spoken per video	≈ 599	≈ 212
Avg. % of off-screen words	31.1%	12.33%
Avg. % words with overlapping speakers	15.39%	14.2%
Avg. number of speakers	4	3.1

The videos can be primarily categorized into 2 categories:

- In-person meeting recordings, where the participants are all sitting in one room, like a meeting room or news studio
- Virtual meeting recordings, where the participants are at separate locations and are communicating via a webcam

These categories were specifically selected for this research, primarily because they represent videos from a wide variety of applications that might need these transcriptions. Both of these categories are found in various domains, such as education

(teaching online vs teaching in a classroom), business meetings, interviews (both offline and online), and newsroom debates. Additionally, various experiments that involve predictions based on facial expressions such as deception prediction and emotion detection, also make use of videos recorded with full frontal face views [Kumar et al., 2021][Hans and Rao, 2021][Wu et al., 2018]. These experiments also need speaker attributed transcriptions to either evaluate or enhance the performance of models in these experiments.

The first category is the recordings from in-person meetings. In this case, the participants have a free head movement while talking to each other, and the camera distance also varies during the course of a video. Thus, such videos tend to have a higher number of partially visible faces and off-screen speakers, and variable face sizes.

The second category includes videos of meeting recordings in a virtual setting. In this case, the participants are at separate locations and are communicating via a webcam. The participants tend to face the camera more often, so the number of partially visible faces is comparatively lower than in the in-person videos due to limited head movement. Similarly, the number of off-screen speakers is smaller, and the face sizes are more consistent compared to the in-person videos.

5.2 Evaluation metric

A common approach to evaluating speaker diarization systems is to use the Diarization Error Rate (DER), which compares the predicted diarization results with reference speaker-labeled segments in the time domain. In contrast, the pipeline designed in this thesis assigns speakers directly to identified words. Hence, a more appropriate measure to evaluate such a system would be the Word Diarization Error Rate (WDER), inspired by the proposed measure in Shafey et al. [2019]. As this implementation is focused more on the speaker attribution part rather than the ASR task, the formula is modified to only consider speaker assignments. Specifically, the WDER for this research is defined as:

$$WDER = \frac{W_{\text{IS}} + W_{\text{NS}}}{W}$$

where,

- W_{IS} is the number of words with incorrect speaker assignments
- W_{NS} is the number of words with no speaker assignments
- W is the number of total words in the video transcript

As pointed out by Mao et al. [2020], WDER is only a measurement of classification error. In the multispeaker diarization setting, the goal is to assign distinct labels to distinct speakers in a conversation; however, these labels may not always match their ground truth counterparts. Thus, WDER on its own is not an appropriate measure for multi-speaker word diarization error. To measure this error, Mao et al. [2020] introduced the Multi-Speaker Word Diarization Error (MWDE), which first

computes the optimal alignment between the predicted labels and the ground truth labels out of all possible alignments M and then calculates the WDER with the new alignments:

$$MWDE = \min_{m \in M} WDER_m$$

Additionally, a key component of the proposed implementation is the automatic determination of the number of clusters and the subsequent face clustering. As the dataset has been annotated with the ground truth values, these values can be directly used to evaluate the clustering output. This is essentially then a classification problem and a classification error rate (CER) can be computed for each video by comparing the outputs to the ground truths. However, similar to the WDER, the goal here is to identify distinct groups of faces correctly. The predicted labels in the clustering outputs may not always match the ground truth labels. Hence, to accurately compute the CER in this case, the optimal alignment is computed between the predicted labels and the ground truth labels out of all possible alignments N, and then the CER is calculated for with the new alignment:

$$CER = \min_{n \in N} \{ \frac{F_{\mathrm{IC}}}{F} \}_n$$

where,

- F_{IC} is the number of faces images with incorrect cluster assignment
- F is the total number of face images

The lower the value of CER, the lower is the number of face images assigned to the wrong clusters. A CER value of 0 indicates perfect clustering.

6. Result and Discussion

This chapter presents the experiment results using the implementation, datasets and evaluation metrics discussed in the previous chapters in order to answer the research questions raised in Chapter 1. The chapter is outlined as follows, with each section corresponding to a research question:

- **RQ1** - How effectively can an active speaker detection (ASD) system be used in conjunction with face clustering to create a video-based speaker diarization (VBSD) system?

  Section 6.1 examines the performance of the VBSD module, especially, how well the face clustering works in converting an ASD output into the diarization results. It also highlights the ability of the VBSD module to more accurately predict the number of speakers as compared to the audio-only speaker diarization (AOSD).

- **RQ2** - How can the audio-only and video-based diarization systems be optimally combined to improve the speaker assignment?

  Section 6.2 reports a more comprehensive view of the performance of the entire pipeline, which uses a combination of both audio and video-based results to perform speaker assignment, and how it compares to the use of the audio-only module.

- **RQ3** What are the major challenges concerning audio-visual characteristics that the combined approach still faces when performing speaker assignments?

  Section 6.3 dives into the audio-visual challenges that the proposed system still faces and their potential solutions.

6.1 RQ1 Evaluation of video-based diarization system

As discussed in Section 4.3, the VBSD module has two primary components - ASD and face clustering. A good ASD module is judged based on the number of speaking faces it can correctly identify in a video. Due to a lack of appropriate annotations in

the dataset, the ASD performance cannot be accurately evaluated on this dataset. However, the nature of this research does not require the ASD module to perform exceptionally well, as the purpose is to assign faces to voices. Hence, any ASD module that can identify a decent number of faces in a video is sufficient for this module. Additionally, as discussed in the Section 2.4, the TalkNet-ASD, used in this implementation, performs well compared to the other state-of-the-art systems on benchmark datasets.

The evaluation of the VBSD module is divided into the following three subsections:

- Section 6.1.1 discusses the ability of the module to correctly predict the number of speakers in a video
- Section 6.1.2 presents the results from the evaluation of the automatic clustering component of the module
- Section 6.1.3 discusses the overall results from the evaluation of the final output of the VBSD module

6.1.1 Detecting the correct number of speakers

Face recognition and clustering is a key factor that heavily governs the performance of the VBSD module, as it has the responsibility to identify the number of speakers in the video, assign faces to the different speakers in the audio, and effectively produce the diarization results. As this implementation is intended to be completely unsupervised, with no prior information regarding the number of speakers, the correct predictions of the number of speakers is an important step that will directly affect the final diarization results.

In the proposed implementation, the optimal number of face clusters (k) for a video is determined automatically using the elbow method, which is the resulting number of speakers for the video. Table 6.1 presents the number of videos for which the module correctly predicted the number of speakers. As expected, the video-based module performs significantly better than the audio-only module in predicting the number of speakers present in a video for virtual meeting videos. On the other hand, it performs equally worse in correctly predicting the number of speakers for in-person meeting videos, which is a result of the higher number of face variations in these videos.

	In-person meeting	**Virtual meeting**	**Total correct predictions**
Audio-only	12	33	45
Video-only	7	59	66
Total Videos	15	61	76

Table 6.1: Correct number of speaker predictions

6.1.2 Evaluating face clustering

Applying face recognition in real-world scenarios like video recordings is still a challenging task, even with state-of-the-art systems. Face recognition systems such as

the face_recognition[8] model based on King [2009] generate embeddings that capture the facial features such as the relative size and positions of the person's eyes, nose, and mouth. In a video setting, a person's face appearing in different frames might be perceived as a different face due to a variety of reasons, such as a change in video angle, face orientation, distance from the camera, lighting conditions and partially covered faces.

To provide a bit more context on the effects of free face movements on face clustering, the talking face images identified and generated for each video in the dataset were manually annotated, and the clustering results were evaluated. Table 6.2 presents the classification error rate (CER) values for both the categories of videos in the dataset.

	In-person meeting	**Virtual meeting**	**Overall**
Mean CER w/o track info	9.63%	0.14%	2.01%
Mean CER w track info	7.75%	0.06%	1.58%

Table 6.2: Mean CER values with and without track information

As the table indicates, when the video has consistently good-quality frontal faces visible in the frames, which is the case with virtual videos, the CER value is almost 0%. When that is not the case, the face clustering suffers. Also, as the table indicates, the clustering results can be improved further by incorporating additional information. In this case, the track information refers to a case where it is known that the faces belonging to a single track are of the same person and, hence, belong to the same cluster. These tracks are generated during the ASD stage of the pipeline mentioned in Section 4.3.1.

6.1.3 Performance of the overall VBSD module

The previous subsections discussed the performance of the face clustering submodule, and its ability to predict the number of speakers accurately. These results directly impact the final output of the VBSD module, as the incorrectly assigned faces and inaccurate number of speakers can deteriorate the quality of the final output. Table 6.3 presents the final evaluation results of the VBSD module in terms of Multi-Speaker Word Diarization Error (MWDE).

For the video-only diarization results, the coverage is not a key factor, as there could be a varying number of off-screen speakers that an ASD module cannot identify by design. Hence, the only point of concern for this module is the accuracy of the unsupervised face clustering and, in turn, the ability of the module to correctly assign speakers to the identified faces.

As expected, the virtual meeting videos have a very low MWDE value due to incorrect assignment. As mentioned earlier, as the face clustering CER is almost 0 for these videos, the effective MWDE is also very low. On the other hand, the MWDE value due to incorrect assignment is relatively higher for the in-person videos. This can be attributed to two major factors: the high face clustering CER mentioned earlier

[8] https://github.com/ageitgey/face_recognition

Table 6.3: Performance of VBSD in terms of MWDE

	In-person meeting	Virtual meeting	Overall
Video missed	31.1%	12.33%	16.03%
Video incorrect	6.33%	2.87%	3.56%
Video overall	37.43%	15.2%	19.59%

Figure 6.1: Trends comparison of CER, word overlap % and video incorrect MWDE for in-person videos

in the section and a combination of a higher word overlap % and a higher number of average speakers per video. Word overlap is the number of words where more than one speaker was identified as speaking. Figure 6.1 attempts to visualize this trend by smoothing the three inputs - face clustering CER, word overlap % and video incorrect MWDE for all the videos using a 1D Gaussian filter. The figure demonstrates that the MWDE tends to increase as the word overlap % increases. Similarly, the MWDE is at its lowest when all three values are low.

6.2 RQ2 Pipeline evaluation results

The previous section dealt with the evaluation of the VBSD module. As the result indicated, the module performed well when the speakers were visible on-screen. However, as the Table 6.3 indicated, there is a major part of the results that is missed by the standalone VBSD module. The pipeline was designed to take advantage of the results of both the VBSD and AOSD modules. So, it is important to evaluate the

performance of the overall pipeline to understand how well the combined modalities perform as compared to the individual modules.

To put a little bit of perspective into the dataset factors that impact the final results, the evaluation is divided into multiple subsections:

- Section 6.2.1 investigates the comparison of final results on the words with overlapping and non-overlapping speech.
- Section 6.2.2 explores the performance of the pipeline on off-screen as well as on-screen spoken words.
- Section 6.2.3 expands the final results of the pipeline on the entire dataset.

6.2.1 Performance evaluation based on overlapped speech

The video transcripts can be divided into two categories- words with and without overlapping speakers. Assigning speakers to non-overlapping words is rather straightforward. However, for words with overlapping speakers, a decision needs to be made on which speaker should the word be assigned to. In the proposed implementation, this decision was made using a simple rule - the speaker with the majority overlap during the word utterance is assigned the word. The first priority is given to the video-only results, and if it couldn't be determined by that, then the audio-only results are used. This is a very naive approach to solving this problem, and it does not always work accurately.

Table 6.4 presents the pipeline results for both categories of words independently. As the results indicate, the MWDE values are extremely poor for both types of videos. Even though the combined approach does improve slightly on the audio-only approach, the results are still not good enough. The quality of the transcriptions and word timestamps is also a major factor affecting these results, discussed in detail in Section 6.3.

However, the proportion of such words is quite low in the videos, close to 15% in both types of videos. Thus, the error resulting due to this gets suppressed in the combined results, as evident in Table 6.6.

		In-person meeting	**Virtual meeting**	**Overall**
Overlap	**Audio-only**	47.15%	43.73%	44.40%
	Combined	35.69%	29.77%	30.94%
Non-overlap	**Audio-only**	5.90%	9.14%	8.50%
	Combined	7.24%	4.14%	4.75%
Overlap %		15.39%	14.2%	14.43%

Table 6.4: MWDE values for words with and without overlapping speakers

6.2.2 Performance comparison on off-screen and on-screen words

Another key factor to consider while evaluating speaker assignments is its performance on on-screen and off-screen speakers. As the primary contribution of the proposed pipeline is targeted towards improving speaker assignments using visual information, it is important to evaluate how it performs while assigning on-screen spoken words. Table 6.5 presents the comparison of the performance of the audio-only module with the combined approach on both on-screen and off-screen words. As expected, the use of visual information has significantly improved the MWDE for on-screen words for virtual meetings, where the video-only module performed better as well. For the off-screen words, the algorithm mostly relies on the output from the audio-only module as no visual information is available.

		In-person meeting	Virtual meeting	Overall
Off-screen	**Audio-only**	21.21%	28.51%	27.07%
	Combined	22.19%	36.02%	33.29%
On-screen	**Audio-only**	10.07%	12.16%	11.74%
	Combined	12.02%	3.59%	5.25%
Off-screen %		31.1%	12.33%	16.03%

Table 6.5: MWDE values for words spoken by off-screen and on-screen speakers

6.2.3 Performance evaluation of the pipeline on the entire dataset

The Table 6.6 shows the overall results as well as the results based on the AOSD module of the pipeline. It also breaks the results down for the individual categories of the videos in the dataset. The output from the audio-only module is considered as the baseline for this evaluation.

Table 6.6: Proposed pipeline evaluation on the entire dataset in terms of MWDE

	In-person meeting	Virtual meeting	Overall
Audio-only	**11.94%**	14.1%	13.67%
Combined	12%	**7.73%**	**8.58%**

The audio-only part performs well on both types of videos with an average overall MWDE of 13.67%. However, the combined approach that this research proposes improves it even further by achieving an MWDE of 8.58%, an improvement of 5.09%. It can be seen that the combined approach performs much better on the virtual meeting recordings, achieving an MWDE of 7.73%, as compared to the audio-only part, which achieves a value of 14.1%, an improvement of 6.37%. On the other hand, the combined approach performs slightly worse than the audio-only part for the in-person meeting recordings, deteriorating from 11.94% in audio-only to 12% for combined.

6.3 RQ3 Challenges of speaker assignment

While the use of visual information certainly improved the speaker assignment in certain aspects, as compared to the use of audio-only diarization, there are still a few

challenges that certain aspects of the current implementation face. Some of these are listed in the following subsections.

6.3.1 Whisper overlap issue

The primary issue with applying diarization results to an automatic speech recognition (ASR) model's output is the inaccuracy that might be introduced due to missing words from speech and inaccurate timestamps. Whisper ASR also suffers from a similar problem. In Whisper's case, these inaccuracies are introduced due to the presence of overlapping speech in a multi-speaker setting. This is a known limitation with Whisper, as the training data often had transcription for one speaker while the other voices were treated as background noise. This limitation results in Whisper sometimes dropping words from speeches during overlap, and the accuracy of the timestamps is severely deteriorated.

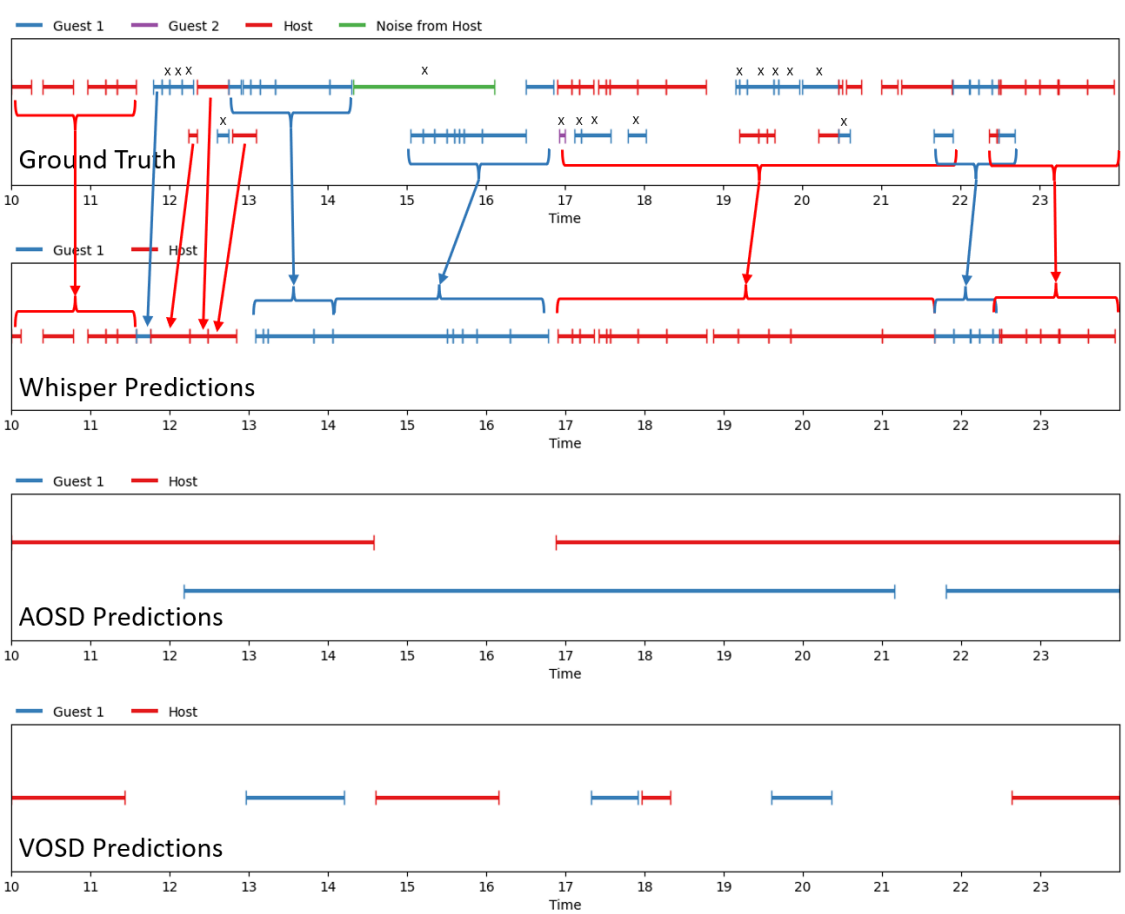


Figure 6.2: Comparison of Whisper transcriptions with the ground truth on a sample output

As the transcriptions for the dataset were derived from Whisper and there was no ground truth available, a complete analysis of this behaviour is not feasible. However, a sample clip containing such overlapping was manually derived. Figure 6.2 demonstrates the results of the pipeline from this clip. The ground truth result shows the word borders of various words spoken by the two primary speakers, and

the Whisper predictions graph shows the word borders of the words predicted by Whisper. The words that are present in both outputs are linked with an arrow, and the words in the ground truth that are not a part of the Whisper transcriptions are marked with a cross. Comparing the 2 results shows that Whisper had missed about 21 words from the expected 71 words. Also, the timestamps get altered in such a way that, during overlap, the speech from one speaker gets' stretched out across a wider timeframe, while the speech from other speakers gets congested into the subsequent timeframe.

Despite these issues, Whisper's state-of-the-art transcription ability, ease of use and support for a wide variety of accents and languages still make it a go-to option for ASR based tasks. Moreover, due to its remarkable transferability, work is already in progress to further improve its performance on multi-talker speech either by adaptation [Li et al., 2023] or prompt tuning [Ma et al., 2024]. Additionally, fine-tuning it on an overlapping speech dataset might also be a way to improve its performance in a multi-talker setting.

6.3.2 Challenges with face clustering in videos

Face clustering is also a challenging task when the faces from video clips are involved. Section 6.1 already presented an analysis regarding this. To demonstrate this problem even further, a couple of videos were annotated and clustering was performed on them. As the face embeddings were a 128-dimension vector, dimensionality reduction using umap [McInnes et al., 2018] was performed to visualize the clustering output. Figure 6.3 and Figure 6.4 present both the ground truth and clustering output of the selected videos.

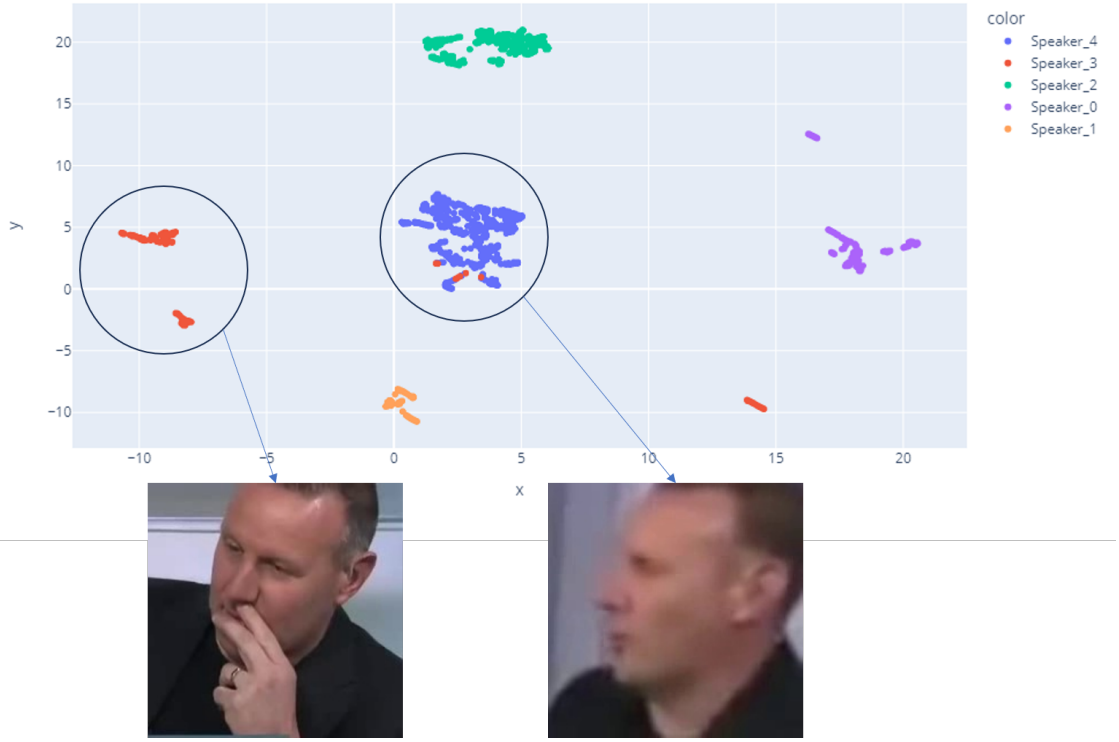
In the case of Figure 6.3, the same person's face appears at different camera angles and distances. Also, in some cases, the face is partially blocked. Due to this, two different face images of the same person end up being part of different clusters. A sample face image from each cluster is shown in Figure 6.3 for reference.

In contrast, video 3 suffers from a different problem as demonstrated in Figure 6.4. In this case, very often, a side view of the faces of two of the speakers in the videos was captured. Due to this, the complete facial landmarks, such as the positions and sizes of the eyes, nose and mouth, could not be captured. As a result, the embeddings generated ended up putting the side faces of two different persons in the same cluster, as demonstrated in Figure 6.4.

Due to these discrepancies, face clustering on its own might not be the best solution when performing VBSD on videos with a more free-flowing nature. Recent attempts to improve the performance of face recognition models on side profiles of faces [Jantarasorn et al., 2023] have shown promise. Bacause of the highly dynamic nature of video recordings, this will always remain a challenging task for face recognition alone to determine whether the person appearing in one frame is the same in another frame. For restricted cases, such as the videos in this dataset where the scenarios in videos don't change too often, models could be trained that take into account additional attributes of a person, such as hairstyles and clothing. As the goal is only to uniquely identify a person, these additional attributes, along with the face information, can also serve as features to distinguish people on screen. Alternatively,

using a combination of both appearance and voice [Xu et al., 2022] could also provide a solution to this problem.

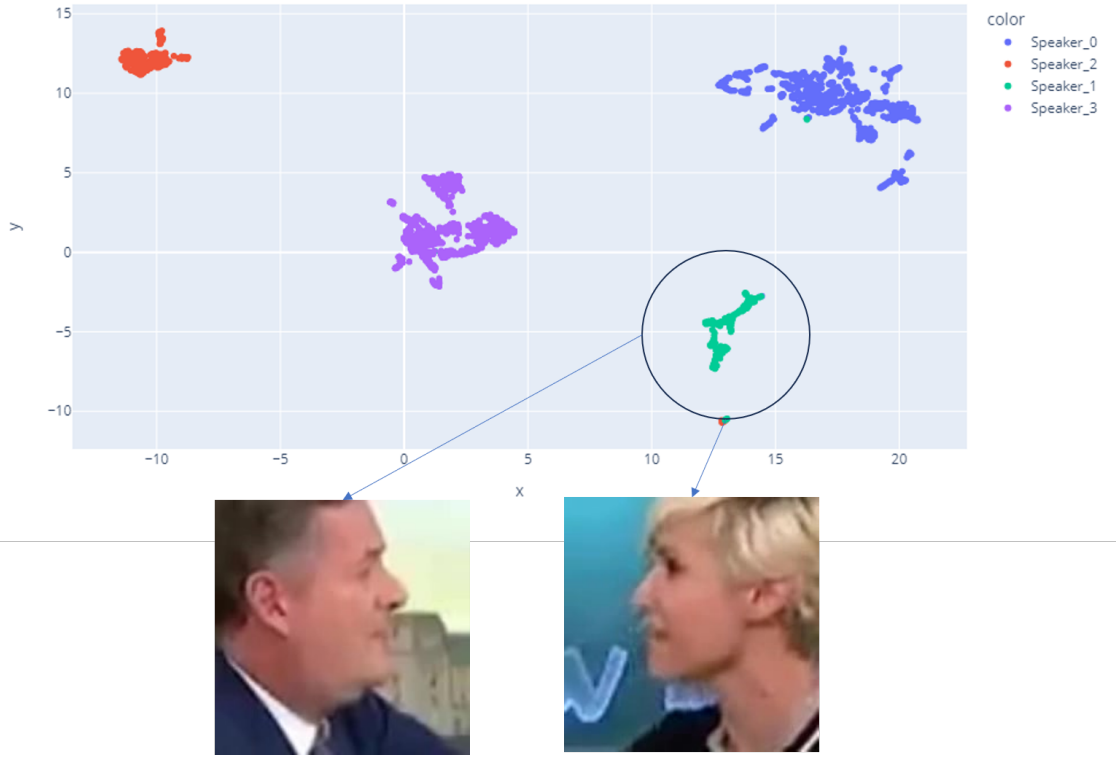
(a) Video 1 ground truth



(b) Video 1 predicted

Figure 6.3: Visualization of face clustering for video 1

(a) Video 2 ground truth



(b) Video 2 predicted

Figure 6.4: Visualization of face clustering for video 2

7. Conclusion

This research presented an end-to-end pipeline that generates speaker-attributed video transcriptions automatically using just the video as input. The exponential growth of video data in the current digital age in domains such as education, business and entertainment has provided an unprecedented opportunity to extract valuable information from such a rich data source. It was further augmented by the pandemic when video became the mainstream form of communication for business, classrooms and other aspects of daily life. Additionally, the development of applications such as natural language processing (NLP), speech emotion detection and spoken document retrieval has seen significant progress recently due to advancements in deep learning. These applications rely heavily on comprehensive and accurate transcripts generated by ASR systems.

Despite the significant progress in ASR systems, exemplified by the introduction of large speech models such as Whisper [Radford et al., 2023], they fall short in distinguishing speech from different speakers in a multi-speaker environment. This is an important requirement in various applications as such differentiation can provide crucial insights into conversational dynamics and speaker-specific contributions. Recent works have tried to address this limitation by combining the ASR output with speaker diarization (SD) tools. However, most of the work on SD has been focused solely on the usage of audio to perform diarization. This limits the capability of an SD system as acoustic data are inherently ambiguous and are easily affected by external factors such as recording environment and noise interference, and even individual speech variations. These factors tend to have deteriorating effects on the output of an SD system.

To address these challenges, this research proposed a novel approach that combines the information available from both audio and visual modalities to improve the quality of speaker-attributed transcriptions, as compared to the use of only audio. In this regard, a pipeline is designed that takes the video as input, performs ASR using Whisper, generates speaker diarization results for both audio and visual modalities, and then combines all the outputs to produce speaker-attributed transcriptions as a result. For the audio-only module, pyannote.audio is used to perform speaker

diarization, while for the VBSD, a new approach is proposed coupling an ASD system with a face clustering algorithm. Additionally, a new dataset is created by annotating videos from YouTube with speaker-attributed transcriptions and face identifiers for evaluation. Effectively, the research attempted to answer the following research questions-

- **RQ1** - How effectively can an ASD system be used in conjunction with face clustering to create a VBSD system?
- **RQ2** - How can the audio-only and video-only diarization systems be optimally combined to improve the speaker assignment?
- **RQ3** What are the major challenges concerning audio-visual characteristics that the combined approach still faces when performing speaker assignments?

During experiments, some of the individual components of the pipeline, as well as the overall performance of the pipeline, were evaluated to answer the mentioned research questions. Firstly, the experiments concluded that the VBSD module, which is a combination of ASD and face clustering, provides an effective way to diarize input videos based on the people appearing on-screen, with the module achieving an overall MWDE value of 3.56% for on-screen speakers. It performed much better on virtual meeting videos as compared to the in-person meeting videos, partially due to the nature of the faces appearing in the videos. Further evaluation of the face clustering module proved that the clustering performance suffered when there was a free movement of faces appearing on screen, which is the case with the in-person meeting videos. In the case of virtual meetings, the clustering achieved an CER of almost 0%.

Secondly, the evaluation of the overall pipeline indicated that the usage of visual cues in conjunction with the audio does help in reducing ambiguity during speaker assignments. The speaker assignment on virtual meeting videos saw a significant improvement with the addition of the visual cues. However, it struggles to achieve any significant improvement for the in-person videos, due to the drawback of the face clustering approach to deal with faces appearing in different angles, especially with the side-profile of faces.

Lastly, the challenges and drawbacks of the current implementation were investigated. A couple of issues were identified. The first is the inability of the Whisper ASR to deal with overlapping speech in a multi-speaker setting, where the resulting transcriptions have inaccurate timestamps and missing words. The other one was the drawback of the face clustering approach discussed earlier and how it fails when the videos have a higher proportion of side profiles or partially covered faces.

Future Work

This research investigated the impact of the use of visual cues in conjunction with audio to enhance speaker attribution of transcriptions. However, as the results demonstrated, there is potential for improvement in certain aspects of the pipeline in future research. Although this research combines multiple independent models

that can always be improved, this section will only focus on the potential areas of future research concerning the major contributions of this thesis.

One of the key components of this pipeline was the face clustering algorithm. It was intended as a way to identify the same speakers from different frames in a video. However, due to the dynamic nature of in-person video recordings, where the faces are not always clearly visible, this algorithm suffers. A potential area of research is the development of models that can, within the scope of a video, distinguish speakers based on additional attributes such as appearance and/or positional information. Videos in various applications, such as meeting rooms or newsrooms, are mostly restricted in the sense that the attributes of the participants, such as hair and clothes, don't change throughout the duration of the video. Hence, these additional attributes could prove effective in distinguishing the speakers.

Another potential area for future research is to improve the performance of large language models such as Whisper on overlapping speech in a multi-speaker setting. As the training data for Whisper often had transcriptions for one speaker while the other voices were treated as background noise, it struggles to accurately transcribe overlapping audio. The research could involve fine-tuning the models on an overlapping speech dataset. Alternatively, nudging the model to only transcribe one speaker at a time and doing it for all the speakers could also be another potential way to improve its performance on overlapping speech.

Lastly, another potential area of improvement for future research is to investigate better ways to assign speakers to transcriptions during overlapping speech. Due to the restrictions of the datasets in this research, a naive approach was used where the overlap between speakers and words was used to perform the assignment. If a better dataset with enough overlapping speech is created in the future, a more suitable algorithm could be devised that can perform assignments with more precision.

Bibliography

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. In *INTERSPEECH*, 2018. (cited on Page 17)

Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8717–8727, December 2022. ISSN 1939-3539. doi: 10.1109/tpami.2018.2889052. (cited on Page 8)

Juan León Alcázar, Fabian Caba, Long Mai, Federico Perazzi, Joon-Young Lee, Pablo Arbeláez, and Bernard Ghanem. Active speakers in context. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12462–12471, 2020. doi: 10.1109/CVPR42600.2020.01248. (cited on Page 19 and 20)

Juan León Alcázar, Fabian Caba Heilbron, Ali K. Thabet, and Bernard Ghanem. Maas: Multi-modal assignation for active speaker detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 265–274, 2021. doi: 10.1109/ICCV48922.2021.00033. (cited on Page 19 and 28)

Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012. doi: 10.1109/TASL.2011.2125954. (cited on Page 2 and 6)

Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and Music Signal Analysis in Python. In Kathryn Huff and James Bergstra, editors, *Proceedings of the 14th Python in Science Conference*, pages 18 – 24, 2015. doi: 10.25080/Majora-7b98e3ed-003. (cited on Page 11)

Latané Bullock, Hervé Bredin, and Leibny Paola Garcia-Perera. Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7114–7118, 2020. doi: 10.1109/ICASSP40776.2020.9053096. (cited on Page 10)

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried

Post, Dennis Reidsma, and Pierre Wellner. The ami meeting corpus: A pre-announcement. In Steve Renals and Samy Bengio, editors, *Machine Learning for Multimodal Interaction*, pages 28–39, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32550-5. (cited on Page 12, 13, and 41)

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *ArXiv*, abs/1904.10509, 2019. (cited on Page 21)

Joon Son Chung, Bong-Jin Lee, and Icksang Han. Who said that?: Audio-visual speaker diarisation of real-world meetings. *ArXiv*, abs/1906.10042, 2019. (cited on Page 27)

Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. In defence of metric learning for speaker recognition. In *Interspeech 2020*, interspeech_2020. ISCA, October 2020a. doi: 10.21437/interspeech.2020-1064. (cited on Page 17)

Joon Son Chung, Jaesung Huh, Arsha Nagrani, Triantafyllos Afouras, and Andrew Zisserman. Spot the conversation: Speaker diarisation in the wild. In *Interspeech 2020*, interspeech_2020. ISCA, October 2020b. doi: 10.21437/interspeech.2020-2337. (cited on Page 27)

Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011. doi: 10.1109/TASL.2010.2064307. (cited on Page 24)

Mireia Diez, Lukáš Burget, Federico Landini, and Jan Černocký. Analysis of speaker diarization based on bayesian hmm with eigenvoice priors. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:355–368, 2020. doi: 10.1109/TASLP.2019.2955293. (cited on Page 24)

Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 06 2010. doi: 10.1007/s11263-009-0275-4. (cited on Page 35)

Xin Fang, Zhen-Hua Ling, Lei Sun, Shu-Tong Niu, Jun Du, Cong Liu, and Zhi-Chao Sheng. A deep analysis of speech separation guided diarization under realistic conditions. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 667–671, 2021. (cited on Page 26)

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with permutation-free objectives. In *Interspeech*, 2019a. (cited on Page 26 and 29)

Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Yawen Xue, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with self-attention. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 296–303, 2019b. doi: 10.1109/ASRU46091.2019.9003959. (cited on Page 26)

Daniel Galvez, Greg Diamos, Juan Ciro, Juan Felipe Cerón, Keith Achorn, Anjali Gopi, David Kanter, Maximilian Lam, Mark Mazumder, and Vijay Janapa Reddi. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage, 2021. (cited on Page 41)

Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):652–662, February 2021. ISSN 1939-3539. doi: 10.1109/tpami.2019.2938758. (cited on Page 28)

Israel D. Gebru, Silèye Ba, Xiaofei Li, and Radu Horaud. Audio-visual speaker diarization based on spatiotemporal bayesian fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1086–1099, 2018. doi: 10.1109/TPAMI.2017.2648793. (cited on Page 27)

Gregory Gelly and Jean-Luc Gauvain. Optimization of rnn-based speech activity detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):646–656, 2018. doi: 10.1109/TASLP.2017.2769220. (cited on Page 10 and 11)

Guillaume Gravier, Gilles Adda, Niklas Paulson, Matthieu Carré, Aude Giraudel, and Olivier Galibert. The etape corpus for the evaluation of speech-based tv content processing in the french language. *International Conference on Language Resources, Evaluation and Corpora*, 01 2012. (cited on Page 12 and 13)

Arnold Sachith A Hans and Smitha Rao. A cnn-lstm based deep neural networks for facial emotion detection in videos. *International Journal Of Advances In Signal And Image Sciences*, 7(1):11–20, 2021. (cited on Page 43)

Andrew O. Hatch, Sachin Kajarekar, and Andreas Stolcke. Within-class covariance normalization for SVM-based speaker recognition. In *Proc. Interspeech 2006*, pages paper 1874–Wed1A1O.5, 2006. doi: 10.21437/Interspeech.2006-183. (cited on Page 24)

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. (cited on Page 21)

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. doi: 10.1109/CVPR.2018.00745. (cited on Page 17)

Chong Huang and Kazuhito Koishida. Improved active speaker detection based on optical flow. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4084–4090, 2020. doi: 10.1109/CVPRW50498.2020.00483. (cited on Page 19)

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3296–3297, 2017. doi: 10.1109/CVPR.2017.351. (cited on Page 21)

Sergey Ioffe. Probabilistic linear discriminant analysis. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 531–542, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33839-0. (cited on Page 14)

Nattapong Jantarasorn, Jakkaphan Whasphuttisit, and Watchareewan Jitsakul. Face recognition using deep learning. In *2023 7th International Conference on Information Technology (InCIT)*, pages 470–474, 2023. doi: 10.1109/InCIT60207.2023.10413069. (cited on Page 52)

Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, Tianyan Zhou, and Takuya Yoshioka. Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers. *ArXiv*, abs/2006.10930, 2020. (cited on Page 28)

Naoyuki Kanda, Xiong Xiao, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, and Takuya Yoshioka. Transcribe-to-diarize: Neural speaker diarization for unlimited number of speakers using end-to-end speaker-attributed asr. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8082–8086, 2022. doi: 10.1109/ICASSP43922.2022.9746225. (cited on Page 28)

Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. (cited on Page 22, 36, and 47)

Srijan Kumar, Chongyang Bai, V. S. Subrahmanian, and Jure Leskovec. Deception detection in group video conversations using dynamic interaction networks. In *International Conference on Web and Social Media*, 2021. (cited on Page 43)

Federico Landini, Alicia Lozano-Diez, Mireia Diez, and Lukáš Burget. From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization. In *Proc. Interspeech 2022*, pages 5095–5099, 2022a. doi: 10.21437/Interspeech.2022-10451. (cited on Page 26 and 29)

Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget. Bayesian hmm clustering of x-vector sequences (vbx) in speaker diarization: Theory, implementation and analysis on standard tasks. *Computer Speech & Language*, 71:101254, 2022b. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2021.101254. (cited on Page 24)

Erik Learned-Miller, Gary B. Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua. *Labeled Faces in the Wild: A Survey*, pages 189–248. Springer International Publishing, Cham, 2016. ISBN 978-3-319-25958-1. doi: 10.1007/978-3-319-25958-1_8. (cited on Page 22 and 36)

Chenda Li, Yao Qian, Zhuo Chen, Naoyuki Kanda, Dongmei Wang, Takuya Yoshioka, Yanmin Qian, and Michael Zeng. Adapting multi-lingual asr models for handling multiple talkers. *ArXiv*, abs/2305.18747, 2023. (cited on Page 52)

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37.

Springer International Publishing, 2016. ISBN 9783319464480. doi: 10.1007/978-3-319-46448-0_2. (cited on Page 21)

Hao Ma, Zhiyuan Peng, Mingjie Shao, Jing Li, and Ju Liu. Extending whisper with prompt tuning to target-speaker asr. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12516–12520, 2024. doi: 10.1109/ICASSP48485.2024.10447492. (cited on Page 52)

Huanru Henry Mao, Shuyang Li, Julian McAuley, and Garrison Cottrell. Speech recognition and multi-speaker diarization of long conversations. In *Interspeech*, 2020. (cited on Page 43)

Pavel Matějka, Ondřej Glembek, Fabio Castaldo, M.J. Alam, Oldřich Plchot, Patrick Kenny, Lukáš Burget, and Jan Černocky. Full-covariance ubm and heavy-tailed plda in i-vector speaker verification. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4828–4831, 2011. doi: 10.1109/ICASSP.2011.5947436. (cited on Page 24)

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3 (29):861, 2018. (cited on Page 52)

Ivan Medennikov, Maxim Korenevsky, Tatiana Prisyach, Yuri Khokhlov, Mariya Korenevskaya, Ivan Sorokin, Tatiana Timofeeva, Anton Mitrofanov, Andrei Andrusenko, Ivan Podluzhny, Aleksandr Laptev, and Aleksei Romanenko. Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario. In *Interspeech 2020*, interspeech_2020. ISCA, October 2020. doi: 10.21437/interspeech.2020-1602. (cited on Page 6)

Giovanni Morrone, Samuele Cornell, Desh Raj, Luca Serafini, Enrico Zovato, Alessio Brutti, and Stefano Squartini. Low-latency speech separation guided diarization for telephone conversations. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 641–646, 2023. doi: 10.1109/SLT54892.2023.10023280. (cited on Page 26)

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 849–856, Cambridge, MA, USA, 2001. MIT Press. (cited on Page 25)

Zexu Pan, Ruijie Tao, Chenglin Xu, and Haizhou Li. Muse: Multi-modal target speaker extraction with visual cues. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6678–6682, 2021. doi: 10.1109/ICASSP39728.2021.9414023. (cited on Page 8)

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015. doi: 10.1109/ICASSP.2015.7178964. (cited on Page 21 and 41)

Tae Jin Park, Kyu J. Han, Manoj Kumar, and Shrikanth Narayanan. Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap. *IEEE Signal Processing Letters*, 27:381–385, 2020. ISSN 1558-2361. doi: 10.1109/lsp.2019.2961071. (cited on Page 25)

Tae Jin Park, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J. Han, Shinji Watanabe, and Shrikanth Narayanan. A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, 72:101317, 2022. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2021.101317. (cited on Page 1, 2, 5, and 6)

Alexis Plaquet and Hervé Bredin. Powerset multi-class cross entropy loss for neural speaker diarization. In *Proc. INTERSPEECH 2023*, 2023. (cited on Page 3, 9, 12, 13, and 32)

Xinyuan Qian, Alessio Brutti, Oswald Lanz, Maurizio Omologo, and Andrea Cavallaro. Audio-visual tracking of concurrent speakers. *IEEE Transactions on Multimedia*, 24:942–954, 2022. doi: 10.1109/TMM.2021.3061800. (cited on Page 8)

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. (cited on Page 1, 2, 3, 18, 36, 42, and 57)

Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, 2018. doi: 10.1109/SLT.2018.8639585. (cited on Page 11)

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. (cited on Page 21)

Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, and Caroline Pantofaru. Ava active speaker: An audio-visual dataset for active speaker detection. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4492–4496, 2020. doi: 10.1109/ICASSP40776.2020.9053900. (cited on Page 9, 18, 19, 20, 27, and 41)

Neville Ryant, Kenneth Ward Church, Christopher Cieri, Alejandrina Cristia, Jun Du, Sriram Ganapathy, and Mark Y. Liberman. The second dihard diarization challenge: Dataset, task, and baselines. In *Interspeech*, 2019. (cited on Page 12 and 13)

Neville Ryant, Kenneth Ward Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Y. Liberman. Third dihard challenge evaluation plan. *ArXiv*, abs/2006.05815, 2020. (cited on Page 24)

Laurent Shafey, Hagen Soltau, and Izhak Shafran. Joint speech recognition and speaker diarization via sequence transduction. pages 396–400, 09 2019. doi: 10.21437/Interspeech.2019-1943. (cited on Page 28 and 43)

Rahul Sharma, Krishna Somandepalli, and Shrikanth S. Narayanan. Crossmodal learning for audio-visual speech event localization. *ArXiv*, abs/2003.04358, 2020. (cited on Page 19)

David Snyder, Guoguo Chen, and Daniel Povey. MUSAN: A Music, Speech, and Noise Corpus, 2015. arXiv:1510.08484v1. (cited on Page 11)

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018. doi: 10.1109/ICASSP.2018.8461375. (cited on Page 14 and 24)

Joon Son Chung. Naver at ActivityNet Challenge 2019 – Task B Active Speaker Detection (AVA). *arXiv e-prints*, art. arXiv:1906.10555, June 2019. doi: 10.48550/arXiv.1906.10555. (cited on Page 18, 19, 20, and 27)

Lei Sun, Jun Du, Chao Jiang, Xueyang Zhang, Shan He, Bing Yin, and Chin-Hui Lee. Speaker Diarization with Enhancing Speech for the First DIHARD Challenge. In *Proc. Interspeech 2018*, pages 2793–2797, 2018. doi: 10.21437/Interspeech.2018-1742. (cited on Page 6)

Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 3927–3935, 2021. (cited on Page 3, 9, 16, 19, 20, 28, 33, and 42)

Murat Taskiran, Nihan Kahraman, and Cigdem Eroglu Erdem. Face recognition: Past, present and future (a review). *Digital Signal Processing*, 106:102809, 2020. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2020.102809. (cited on Page 21)

S.E. Tranter and D.A. Reynolds. An overview of automatic speaker diarization systems. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5): 1557–1565, 2006. doi: 10.1109/TASL.2006.878256. (cited on Page 6)

Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056, 2014. doi: 10.1109/ICASSP.2014.6854363. (cited on Page 24)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (cited on Page 20)

Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno. Speaker diarization with LSTM. In *2018 IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP), pages 5239–5243, 2018. doi: 10.1109/ICASSP.2018.8462628. (cited on Page 25)

Zhe Wu, Bharat Singh, Larry S. Davis, and V. S. Subrahmanian. Deception detection in videos. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, pages 1695–1702. AAAI Press, 2018. (cited on Page 43)

Eric Zhongcong Xu, Zeyang Song, Satoshi Tsutsui, Chao Feng, Mang Ye, and Mike Zheng Shou. Ava-avd: Audio-visual speaker diarization in the wild. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22. ACM, October 2022. doi: 10.1145/3503161.3548027. (cited on Page 27 and 53)

Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (cited on Page 21)

Ruiqing Yin, Hervé Bredin, and Claude Barras. Speaker change detection in broadcast TV using bidirectional long short-term memory networks. In *Proc. Interspeech 2017*, pages 3827–3831, 2017. doi: 10.21437/Interspeech.2017-65. (cited on Page 10 and 12)

Ruiqing Yin, Hervé Bredin, and Claude Barras. Neural speech turn segmentation and affinity propagation for speaker diarization. In *Interspeech*, 2018. (cited on Page 10 and 14)

Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li. S3fd: Single shot scale-invariant face detector. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 192–201, 2017. (cited on Page 21, 22, and 34)

Yuanhang Zhang, Jing-Yun Xiao, Shuang Yang, and S. Shan. Multi-task learning for audio-visual active speaker detection. In *The ActivityNet Large-Scale Activity Recognition Challenge*, pages 1–4, 2019. (cited on Page 19, 20, and 27)

I herewith assure that I wrote the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

Magdeburg, 7th June 2024