

UNDERGRADUATE FINAL YEAR PROJECT REPORT

Department of Computer and Information Systems

NED University of Engineering and Technology



Active Speaker Detection

Group Number: 02

Batch: 2019-2023

Group Member Names:

Kashaf Khan	CS-19002
Hafsa Zafar	CS-19012
Aleesha Ahmed	CS-19013

Approved by

.....

Dr. Urooj Ainuddin
Assistant Professor
Project Advisor



Author's Declaration

We declare that we are the sole authors of this project. It is the actual copy of the project that was accepted by our advisor(s) including any necessary revisions. We also grant NED University of Engineering and Technology permission to reproduce and distribute electronic or paper copies of this project.

Signature and Date	Signature and Date	Signature and Date
Kashaf Khan	Hafsa Zafar	Aleesha Ahmed
CS-19002	CS-19012	CS-19013

.....

.

.

.



Statement of Contributions

- All the members contributed equally in performing Literature Review for the provided problem via readings and exploration of the relevant research papers.
- Ms. Kashaf Khan has performed video processing, implemented video model and integrated audio-visual model.
- Ms. Hafsa Zafar has performed audio processing and implemented speaker change model.
- Ms. Aleesha Ahmed has performed audio processing, implemented audio model and integrated audio models.
- Ms. Hafsa Zafar has contributed in writing final-year report.

Executive Summary

1. Problem Statement

Meetings are important in our society, whether they are formal or informal. Meetings are frequently held in colleges and enterprises to coordinate professional things like projects, research, money, etc. The subject of the conversation is always the one speaking because they often have everyone's attention.

Additionally, we can see specific behaviors or gestures that are used in communicating along with voice. There are various situations where multiple people are gathering in a crowded location, and it might be helpful to interpret this audiovisual information in order to estimate the person of interest or active speaker.

2. Background Information

To start and improve our project, one of the most crucial things we had to complete was to research numerous research articles. The subject of data analytics, specifically the subfield of machine learning, is the sole cornerstone of our project.

The goal is to create and integrate machine learning and deep learning models for active speaker detection in crowded environments where multiple people move and speak at once. Multiple audio and video sensors are used by the system to provide both audio and visual data. Active Speech (AVA) dataset, which comprises labelled audio for the audio module, is used for this purpose. We simply combine the output of the two models after constructing the audio and video models.

3. Methodology used to solve the problem

After preprocessing the dataset, the auditory model is fed with input audio which uses a Convolutional Neural Network (CNN) to classify if any participant in the meeting is speaking or not. In this model the dataset is preprocessed, audios and features are extracted and the CNN model is applied to get the desired output. Then the speaker change model is implemented, that determines when the speaker in the audio changes when multiple speakers are present. This model makes use of speaker diarization and other audio processing techniques in order to improve the accuracy of the system.

The video model is then implemented that preprocess the video and takes video frames as an input. Features are then extracted using Retina Face library for face detection. When the face and facial features are extracted, separate matrices of landmarks of every face in each

Frame are formed. The difference between the two matrices of subsequent frames are calculated and the resulting matrix is compared with the threshold.

If the resulting matrix is greater than the threshold the speaker is active. If the resulting matrix is less than the threshold then the speaker is non-active.

The output is a probability distribution over the two possible outcomes (speaking or not speaking). Since the goal is a binary classification, the detection of the active speaker happens when the corresponding probability exceeds the threshold value. The evaluation of each method is performed by computing the accuracy of the predictions on frame-by-frame basis.

Our final system is obtained by performing integration of individual audio and video models.

4. Major findings

After exploring multiple research papers active speaker detection is not an easy task. It makes use of both audio and visual information instead of making use of any one information for the prediction of an active speaker. Multiple techniques have been implemented in order to perform this task as discussed in literature review chapter.

5. Conclusion

In our project, we have implemented models of audio and video. Audio model takes audio signal as an input and detects whether the speaker is speaking or not. While, speaker change model predicts when the speaker in the audio changes. On the other hand, video model predicts whether the speaker is active or not by processing the video.

Acknowledgments

First and foremost, praise is to Almighty Allah who gives us the strength and ability to think, work and deliver what we are assigned to do. Secondly, we are grateful to our internal supervisor Dr. Urooj Ainuddin (Assistant Professor at N.E.D University of Engineering and Technology), who guided us in this project.

We also acknowledge our teachers who guided, taught and helped us during our study period. We would also like to thank all departmental staff and university staff, who had assisted us during our stay at the university. Finally, we would like to express our deep sense of gratitude and earnest thanksgiving to our dear parents for their moral support and heartfelt cooperation in doing the main project.

TABLE OF CONTENTS

Author's Declaration.....	ii
Statement of Contributions	iii
Executive Summary	iv
Acknowledgments	v
Table of Contents	vi
List of Figures	vii
United Nations Sustainable Development Goals	viii
Similarity Index Report.....	ix
Chapter 1 Introduction	11
1.1 Background Information.....	11
1.2 Significance and Motivation	11
1.3 Aims and Objectives	12
1.4 Methodology	13
1.5 Report Outline	15
Chapter 2 Literature Review	16
2.1 Introduction.....	16
2.2 Literature Review	16
2.3 Summary	19
Chapter 3 Audio Model.....	20
3.1 Introduction	20
3.2 Data Preprocessing	20
3.3 Audio Extraction	22
3.4 Features Extraction	23
3.5 CNN Implementation	24
3.6 Libraries used in the Model	25
3.6 Summary	26
Chapter 4 Video Model.....	28
4.1 Introduction	28
4.2 Video Preprocessing	28
4.3 Features Extraction	29
4.4 Result Analysis	31
4.5 Summary	32

Chapter 5 Speaker Change Model	34
5.1 Introduction	34
5.2 Voice Activity Detection	34
5.3 Clustering	35
5.4 Training GMM	37
5.4 Segmentation	38
5.4 Speaker Diarization	39
5.5 Summary	42
Chapter 6 Fusion of Audio Visual Model.....	43
6.1 Introduction	43
6.2 Integration of Audio Models	44
6.3 Score Based Fusion	44
6.4 Integration of Audio-Visual Model	45
6.5 Summary	45
6.6 Predicted Outputs	46
Chapter 7 Libraries Implemented	52
6.1 Introduction	52
6.2 Integration of Audio Models	52
Chapter 8 Conclusion.....	58
8.1 Summary	58
8.2 Methods for Implementation	61
8.3 Challenges	64
8.4 Recommendations for Future Work.....	67
8.5 References.....	70

United Nations Sustainable Development Goals

The Sustainable Development Goals (SDGs) are the blueprint to achieve a better and more sustainable future for all. They address the global challenges we face, including poverty, inequality, climate change, environmental degradation, peace, and justice.

- ☐ No Poverty
- ☐ Zero Hunger
- ☐ Good Health and Well being
- ☐ Quality Education
- ☐ Gender Equality
- ☐ Clean Water and Sanitation
- ☐ Affordable and Clean Energy
- ☐ Decent Work and Economic Growth
- ☐ Industry, Innovation and Infrastructure
- ☐ Reduced Inequalities
- ☐ Sustainable Cities and Communities
- ☐ Responsible Consumption and Production
- ☐ Climate Action
- ☐ Life Below Water
- ☐ Life on Land
- ☐ Peace and Justice and Strong Institutions
- ☐ Partnerships to Achieve the Goals

Similarity Index Report

Following students have compiled the final year report on the topic given below for partial fulfillment of the requirement for Bachelor's degree in Computer Systems Engineering.

Project Title **Active Speaker Detection**

1.	<u>Kashaf Khan</u>	<u>CS-19002</u>
2.	<u>Hafsa Zafar</u>	<u>CS-19012</u>
3.	<u>Aleesha Ahmed</u>	<u>CS-19013</u>

This is to certify that Plagiarism test was conducted on complete report, and overall similarity index was found to be less than 20%, with maximum 5% from single source, as required.

Signature and Date

Dr. Urooj Ainuddin

.....

Chapter 1

Introduction

1.1 Background Information

One of the most important tasks we had to do was to research various research papers to initiate and develop our project. The foundation stone of our project solely lies in the field of data analytics, specifically in the sub-field of machine learning.

The aim is to develop and integrate machine and deep learning models for the detection of active speaker in a cluttered environment where more than one person speaks and move. The system utilizes both audio and visual information from multiple audio and video sensors. For this purpose Active Speech (AVA) dataset is used that contains labelled audio for audio module. After developing both audio and video models we simply integrate the output of both the models.

1.2 Significance and Motivation

Active speaker detection seeks to classify if a person at a given time in a video is speaking or not. Research in active speaker detection from videos is faced with challenges such as the presence of multiple people. Therefore, this projects helps in automating the process of streaming real-time videos and is helpful for the groups to make complex decisions without the luxury of face to face contact.

In many speaker situations like video conferences, webinars, and teleconferences, active speaker recognition is essential for enhancing communication. Participants can concentrate on the speaker they are currently listening to by identifying the active speaker. This removes ambiguity, boosts participation, and guarantees effective information exchange.

1.3 Aims and Objectives

This project helps to detect the active speaker in a video with number of participants possibly changing during the interaction. It helps to crop the face of the active speaker and focus out the camera from the face of the speaker when the active speaker changes, and refocuses on the new active speaker. In situations where accurate transcription and captioning of audio or video content are required, active speaker detection becomes essential. Associating the correct text with the corresponding speaker enhances the quality and accuracy of transcribed content. This is particularly valuable in creating accessible content for individuals with hearing impairments.

Active speaker detection provides valuable context for analyzing multimedia content. It enables content creators, researchers, and analysts to understand the dynamics of conversations, discussions, or events. This context can be harnessed to extract insights, sentiment analysis, and content summarization. In scenarios where immediate decisions are required, such as live broadcasting or emergency response, active speaker detection facilitates real-time processing. This is crucial for ensuring seamless communication and timely actions.

1.4 Methodology

Generally, there are three different approaches: 1) systems that use audio-only; 2) audio-visual systems; and 3) systems that use other forms of inputs for detection. We believe that complementing the auditory modality with visual information can be useful, therefore to increase the accuracy of the system, we combine information from both the audio and the video model.

After preprocessing the dataset, the auditory model is fed with input audio which uses a Convolutional Neural Network (CNN) to classify if any participant in the meeting is speaking or not. In this model the dataset is preprocessed, audios and features are extracted and the CNN model is applied to get the desired output. Then the speaker change model is implemented, that determines when the speaker in the audio changes when multiple speakers are present. This model makes use of speaker diarization and other audio processing techniques in order to improve the accuracy of the system.

The video model is then implemented that preprocess the video and takes video frames as an input. Features are then extracted using Retina Face library for face detection. When the face and facial features are extracted, separate matrices of landmarks of every face in each frame are formed. The difference between the two matrices of subsequent frames are calculated and the resulting matrix is compared with the threshold. If the resulting matrix is greater than the threshold the speaker is active. If the resulting matrix is less than the threshold then the speaker is non-active.

The output is a probability distribution over the two possible outcomes (speaking or not speaking). Since the goal is a binary classification, the detection of the active speaker happens when the corresponding probability exceeds the threshold value. The evaluation of each method is performed by computing the accuracy of the predictions on frame-by-frame basis.

Our final system is obtained by performing integration of individual audio and video models.

Following is the flowchart for the system:

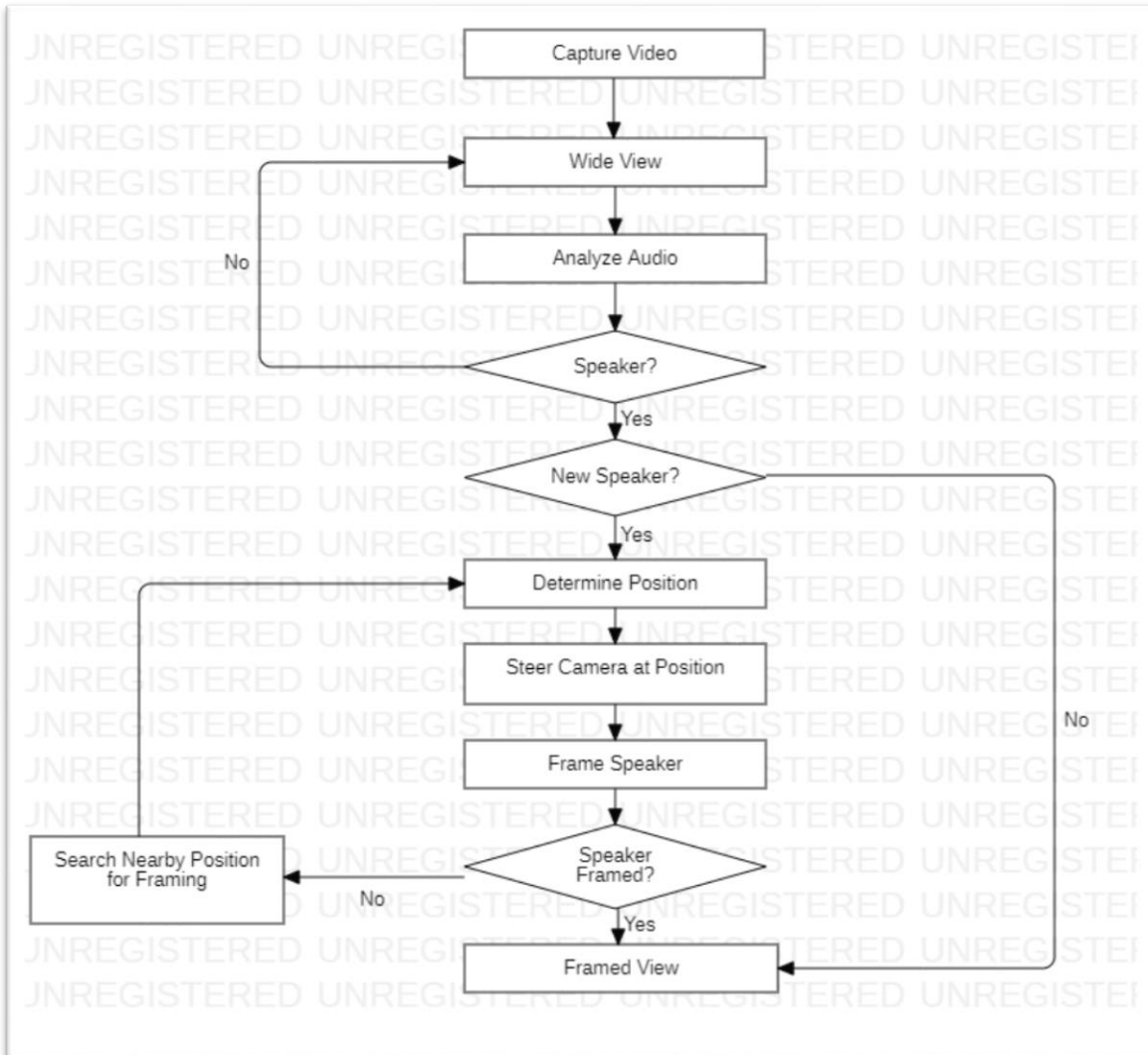


Figure 1: Flow Diagram of the Project

1.5 Relevance to Mapped SDGS

- **Industry, Innovations and Infrastructure**

This SDG is selected and is relevant to our project as it encourages innovation with increased resource-use efficiency, thus upgrading technological capabilities of industrial sector through automation and supporting domestic technology development.

- **Sustainable Cities and Communities**

This SDG is selected as it supports least developed countries, through technical assistance. Our project provides affordable, accessible and sustainable communication systems for everyone. Specially fulfilling the needs of those in business sector, education sector etc. It supports positive economic, social and environmental links.

1.6 Report Outline

The scope of the project is that it helps in automating the process of streaming real-time videos such as press conferences, media broadcasts, briefings, and important meetings at professional levels as well as in the entertainment sector. Any company with remote teams who regularly meet online can benefit from active speaker windows. The technology is particularly useful for groups that must regularly reach consensus on complex decision points without the luxury of face-to-face contact.

Chapter 2

Literature Review

2.1 Introduction

Here is the literature review of multiple findings and research that we have done for this project. We studied different techniques that are implemented for this system until now.

2.2 Literature Review

- The research paper was published in 2022 by Adekunle Akinrinmade, Emmanuel Adetiba, and Joke A.Badejo. This paper proposes a technique to determine as the speaker's lips open and shut, revealing the speaker's mouth's inside contents. The standard deviations of color histograms of the mouth region can be used to identify active speakers in movies. For the first time, the standard deviation of the mouth region from frame to frame is being employed for the prediction of active speakers in a unique concept for active speaker recognition in digital videos. Future research will examine the combination of visual and aural cues to enhance the outcomes. The paper introduces a novel approach for active speaker recognition in digital videos by focusing on the visual cues derived from the movement of a speaker's mouth. The technique leverages the opening and closing of the speaker's lips, which reveal the contents inside the mouth during speech. This motion is considered as a distinctive visual cue that can be used for identifying when a speaker is actively speaking in a video.
- The proposed method utilizes the standard deviations of color histograms within the mouth region to perform this recognition. The primary objective of this paper is to develop a technique for identifying active speakers in digital videos based on the distinctive visual cue of the opening and closing of a speaker's mouth. This technique aims to enhance speaker recognition accuracy using only the visual cues provided by the mouth region's motion. The proposed technique involves analyzing the mouth region of speakers in video frames. The variation in color distribution within the mouth region is measured by calculating the standard deviation of

- The research paper was published in 2021 by Francisco Madrigal, Frédéric Lerasle, Lionel Pibre, and Isabelle Ferrané. This paper propose speaker detection framework incorporating audiovisual elements from the conference environment. CNN processes visual cues by analysing motion and raw pixels (RGB pictures). A concept of three features—audio, video, and social—that are based on participant gaze—whose direction is considered to be the same as the orientation of the head—are used to estimate speakers in a meeting situation. The outcomes might be enhanced by additional social behavior study.

- The research paper was published in 2021 by Tae Jin Parka, Naoyuki Kandab, Dimitrios Kyu J. Hanc, and Shinji Watanabed. This paper propose speaker diarization technique to enable speech recognition on multi-speaker audio recordings. By combining the most recent advancements in neural approaches, this research makes a significant contribution to the field by offering a survey work that will help the field move closer to an effective speaker diarization. The complete overview of speaker diarization strategies presented in this work highlights the most recent progress made in deep learning-based diarization techniques. A speaker diarization system was initially created as a pipeline of smaller modules, including front-end processing, SAD, segmentation, speaker embedding extraction, clustering, and post-processing, resulting in a system that was largely independent from other speech application components.

- The research paper was published in 2019 by Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, Stefanos Zafeiriou. RetinaFace, a reliable single-stage face detector, is presented in this paper. It uses joint extra-supervised and self-supervised multi-task learning to achieve pixel-wise face localization on various scales of faces. The technique of Retina Face for face identification, is put forth to address the difficult problem of concurrent dense localization and alignment of faces of arbitrary scales in images. Retina Face also produces far more accurate results in detecting faces when paired with cutting-edge techniques.

- The research paper was published in 2019 by Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver. This paper propose that in video analysis algorithms for uses like speaker diarization, video re-targeting for meetings, voice augmentation, and human-robot interaction, active speaker detection is a crucial step. For this assignment, the labelled audio-visual dataset has limited evaluations of algorithms with regard to data diversity, environments, and accuracy.

- The research paper was published in 2021 by Juan Leon Alcazar, Fabian Caba, Heilbron2, and Ali K. Thabet1 & Bernard Ghanem. This paper produce that to successfully recognize active speakers, multi-modal cues must be carefully integrated. Currently used techniques concentrate on modelling and combining frame-level short-term audiovisual properties for specific speakers. It provides a unique method for identifying active speakers that directly addresses the multimodal character of the issue and offers a simple method in which speakers in the scene are allocated to previously identified speech events based on their independent visual attributes. For the active speaker recognition problem, MAAS is presented, a brand-new multi-modal assignation method based on graph convolutional networks. In order to simultaneously detect speech events and determine the optimal source (active speaker), our method directly optimizes a graph. As a tough transfer dataset for future research, we also propose Talkies, a novel benchmark for active speaker detection with difficult circumstances.

- The research paper was published in 2020 by Juan Leon Alcazar, Fabian Caba Heilbron, Long Mai2, Federico Perazzi, Joon-Young Lee, and Pablo Arbelaez. Modelling audiovisual data from a single speaker is the main focus of current active speaker detection techniques. This approach may work well in situations with a single speaker, but it hinders reliable detection when trying to figure out which of several potential speakers is speaking. A context-aware model for active speaker detection has been developed that makes use of long-term horizons and co-occurring speaker signals. We have demonstrated that our approach outperforms the state-of-the-art in active speaker detection and performs exceptionally well in difficult situations where there are a large number of potential speakers or only little faces on the screen.

- The research paper was published in 2020 by Abudukelimu Wuerkaixi, You Zhang, Zhiyao Duan, Changshui Zhang. Systems for active speaker detection (ASD) are crucial components for determining multi-talker conversations. They seek to identify which speakers, if any, are conversing at any particular moment in a visual scene. There is disagreement among researchers on ASD over what constitutes an engaged speaker. In this work, the terminology is clarified, and synchronisation between the aural and visual speaking activities is necessary. The paper suggest a cross-modal contrastive learning approach and use positional encoding in attention modules for supervised ASD models to take advantage of the synchronization cue to solve this issue. Our model addresses the restriction of existing models by successfully detecting unsynchronized speaking as not speaking, according to experimental results.

2.3 Summary

A literature review is a critical and integral part of any research project, and its importance cannot be overstated. It serves as a foundation for your research and plays a pivotal role in shaping the direction, scope, and methodology of your project. In the above findings, we evaluate the results of the previous work that is already done on this idea and we have gone through all the future work of the previous projects so that we can try to implement those factors in our project to make it unique.

Chapter 3

Audio Model

3.1 Introduction

This chapter will give the overall view of how the audio model is implemented. Audio model uses Convolutional Neural Network (CNN) to classify if any participant in the meeting is speaking or not and to distinguish as the speaker changes. This chapter will define the steps that has been taken to implement this model in detail like how the dataset is preprocessed, features are extracted and CNN model is implemented to get the desired output.

3.2 Preprocessing

Data preprocessing is a fundamental step in data analysis and machine learning pipelines. It involves preparing and cleaning raw data to make it suitable for analysis, modeling, and training machine learning algorithms. The quality and effectiveness of the final analysis or model often depend on the quality of data preprocessing.

Audio dataset consists of features youtube identifiers, start_time_stamps, end_time_stamps and output speech having labels clean_speech, no_speech, speech_with_music, speech_with_noise. Our dataset consists of some duplicate and missing values that may give an incorrect view of the overall statistics of data. Duplicate data values are dropped and there were many null or missing values that were filled using numerical methods like mean and mode. .The count of missing values (null or NaN) in each column of a Data Frame named df is determined via sum (). The amount of missing values for each column is displayed in a series that is returned.

- Missing timestamp feature values are filled with arithmetic mean to remove the null values.
- Missing values of youtube identifier feature and speech are filled using mode in order to remove the null values.

- A data frame with the name df uses the function value_counts () to count the instances of unique values in a certain column. It gives back a series that displays the counts of every distinct value in the given column.

	YouTube Identifier	label_start_timestamp_seconds	label_end_timestamp_seconds	Speech
0	JNb4nWexD0I	900.00	901.15	NO_SPEECH
1	JNb4nWexD0I	901.15	902.20	CLEAN_SPEECH
2	JNb4nWexD0I	902.20	902.66	SPEECH_WITH_NOISE
3	JNb4nWexD0I	902.66	904.79	NO_SPEECH
4	JNb4nWexD0I	904.79	905.40	CLEAN_SPEECH

	YouTube Identifier	label_start_timestamp_seconds	label_end_timestamp_seconds	Speech_CLEAN_SPEECH	Speech_NO_SPEECH	Speech_SPEECH_WITH_MUSIC	Speech_SPEECH
0	JNb4nWexD0I	900.00	901.15	0	1	0	
1	JNb4nWexD0I	901.15	902.20	1	0	0	
2	JNb4nWexD0I	902.20	902.66	0	0	0	
3	JNb4nWexD0I	902.66	904.79	0	1	0	
4	JNb4nWexD0I	904.79	905.40	1	0	0	
...
7133	2fwni_kjf2M	1780.59	1789.95	0	0	0	
7134	2fwni_kjf2M	1789.95	1791.27	0	1	0	
7135	2fwni_kjf2M	1791.27	1795.23	0	0	0	
7136	2fwni_kjf2M	1795.23	1796.31	0	1	0	
7137	2fwni_kjf2M	1796.31	1800.00	0	0	0	

Fig 2: Data Preprocessing

3.3 Audio Extraction

In machine learning, the technique of extracting and isolating particular elements or traits from audio signals in order to use them as input features for machine learning models is known as audio extraction. In order for machine learning algorithms to interpret and analyze the structured audio data, useful information must be extracted from the raw audio data. Depending on the application and the particular elements of interest, audio extraction can cover a wide range of topics. In machine learning, a few typical methods of audio extraction include:

Feature extraction is the process of obtaining significant characteristics from audio signals that accurately reflect the key components of the audio material. In addition to statistical traits, these variables may also have spectral, temporal, and rhythmic characteristics. Many jobs, like speech recognition, use feature extraction.

In our project, youtube videos are downloaded using pytube library. All the unique youtube identifiers in our dataset having different time stamps are attached with the URL of youtube and are redirected to that link. The audios are then filtered and are streamed from those YouTube videos. Once the audios are extracted they are saved in a separate folder named ‘audios’ in the form of mp4.

```
[ ] import os
    for file in os.listdir('/content/drive/MyDrive/audios'):
        print(file)

Lispettore Derrick - Chi ha ucciso Johann Kahl 691979.mp4
Paris Mon Paradis (Film Burkinabè) - Sous-titré français - Film complet.mp4
Sword of Vengeance - Nigeria Nollywood Movie.mp4
The Executioners Song 1982 Tommy Lee Jones Eli Wallach Full Length Movie.mp4
Agents of Secret Stuff.mp4
Berkeley Square 10 BBC 1998 (Último capítulo de la serie) (Last Chapter).mp4
Distorting Mirror of the Soul Episode 2 Russian TV Series StarMedia Melodrama English Subtitles.mp4
Kill Stalin - Episode 6 Russian TV Series StarMedia Military Drama English Subtitles.mp4
palace treasure 1 - Nigeria Nollywood movie (online-audio-converter.com) 2.mp3
Petites coupures (online-audio-converter.com).mp3
Greatest Kung Fu movie ever!.mp3
The Trap.m4a
Gröna hissen - Hela föreställningen från 2010 med Johan Ulveson och Eva.m4a
Eterna sonrisa de New Jersey con Daniel Day Lewis -.m4a
Return of the Tiger.m4a
Jest Sprawa [Komedia Polska 2002].m4a
L'ispettore Derrick - La tentazione 631979.m4a
Checkpoint 1956.m4a
കിസ്തൂർ കിസ്തൂർ Part01.m4a
L'ispettore Derrick - Il campione 291976.m4a
L'ispettore Derrick - Eco di un omicidio (Mordecho) - 26096.m4a
Peter O'Toole Rogue Male.m4a
[Kung Fu] Death Duel Of Kung Fu (1979).m4a
```

Fig 3: Audio Extraction

3.4 Features Extraction

In audio signal processing, a branch of signal processing, audio feature extraction is a crucial stage. The process of extracting features from audio input is converting unprocessed audio signals into a group of exemplary features or attributes that adequately reflect the audio content. These features are used as inputs for pattern recognition, machine learning algorithms, and numerous audio processing tasks. For the purpose of enabling the analysis, categorization, and interpretation of audio data, significant feature extraction is essential. An overview of the feature extraction from audio input procedure is provided below:

Audio preprocessing: To improve the quality of the audio data and make future analysis easier, preprocessing is frequently done before feature extraction. Tasks like noise reduction, resampling, and normalization may fall under this category. It deals with the modification or processing of audio signals. It has been found that using features extracted from the audio signal as input to the basic model would result in significantly better performance than using the raw audio signal as input. The widely used method for removing characteristics from an audio stream that has been used in our project is called MFCC.

```
[ ] features

{'/content/drive/MyDrive/audios/Lispettore Derrick - Chi ha ucciso Johann Kahl 691979.mp4': array([-214.07213 , 130.55882 , -32.65921 ,
46.007438 ,
0.52740026, 2.4344192 , 1.820533 , -6.5410147 ,
-2.3567536 , -1.5370429 , -5.2236166 , -1.9661151 ,
-4.3845506 , -2.2181535 , -2.8729577 , -1.8609872 ,
-4.9982038 , -2.873962 , -2.7101005 , -3.722831 ,
-4.0535464 , -2.9775712 , -2.7118552 , -2.87247 ,
-2.4349265 , -2.07165 , -2.9671308 , -2.7221851 ,
-2.4709272 , -2.8322768 , -3.5805566 , -1.7948803 ,
-2.3798807 , -2.411937 , -2.5008103 , -2.1788058 ,
-2.7587368 , -2.4003363 , -2.0608222 , -2.2612042 ,
-2.354794 , -2.223812 , -2.4655125 , -2.2870095 ,
-2.0007505 , -2.1519258 , -2.2769318 , -2.2502947 ,
-2.2573683 , -1.9837425 ], dtype=float32),
'/content/drive/MyDrive/audios/Paris Mon Paradis (Film Burkinabè) - Sous-titré français - Film complet.mp4': array([-2.32546006e+02, 1.00531166e+02,
-9.17185020e+00, 1.41128626e+01,
-1.02591038e+01, -5.65466833e+00, -1.15199223e+01, -6.85282898e+00,
-1.18432751e+01, -5.29749441e+00, -1.00165911e+01, -6.15125370e+00,
-7.21190214e+00, -5.91911316e+00, -4.85913754e+00, -4.68365049e+00,
-6.14955807e+00, -3.07597089e+00, -4.94863415e+00, -2.80024171e+00,
-3.98405337e+00, -2.94372535e+00, -3.02474189e+00, -3.01380754e+00,
-3.30812526e+00, -2.64011216e+00, -2.73674464e+00, -2.28998590e+00,
-2.52756906e+00, -1.28349161e+00, -2.49455380e+00, -8.49991620e-01,
-1.62669384e+00, -2.44803488e-01, -1.18589640e+00, 5.47404541e-03,
```

Fig 4: Features Extraction

3.5 CNN Implementation

Convolutional Neural Networks, or CNNs, are a subset of deep learning models that are generally used for processing grid-like input, such spectrograms of images and sounds. It is a particular neural network architecture that has excelled at visual identification, classification, and feature extraction tasks. CNNs have made major contributions to the field of computer vision and are inspired by the visual processing mechanisms in the human brain. Here is a thorough description of CNN:

- **Convolutional Layers:** CNN's convolutional layers are its brain. The input data is processed by a convolutional layer using a number of filters (also known as kernels). These filters execute element-wise multiplication and summation as they slide across the input data. A feature map that highlights specific patterns or features in the data is the end result.
- **Pooling Layers:** By using pooling layers, in particular max pooling and average pooling, it is possible to shrink the size of the feature maps while still retaining crucial data. The model becomes less computationally demanding and more resilient to little distortions or translations in the input data thanks to pooling.

- **Activation Functions:** After convolutional and pooling layers, activation functions like ReLU (Rectified Linear Activation) are used to add non-linearity to the model. CNNs can now recognize intricate correlations and patterns in the data.
- **Full Layer Connectivity:** CNNs frequently have fully linked layers after a number of convolutional and pooling layers. These layers compile the previously learnt characteristics and map them to the final output classes.

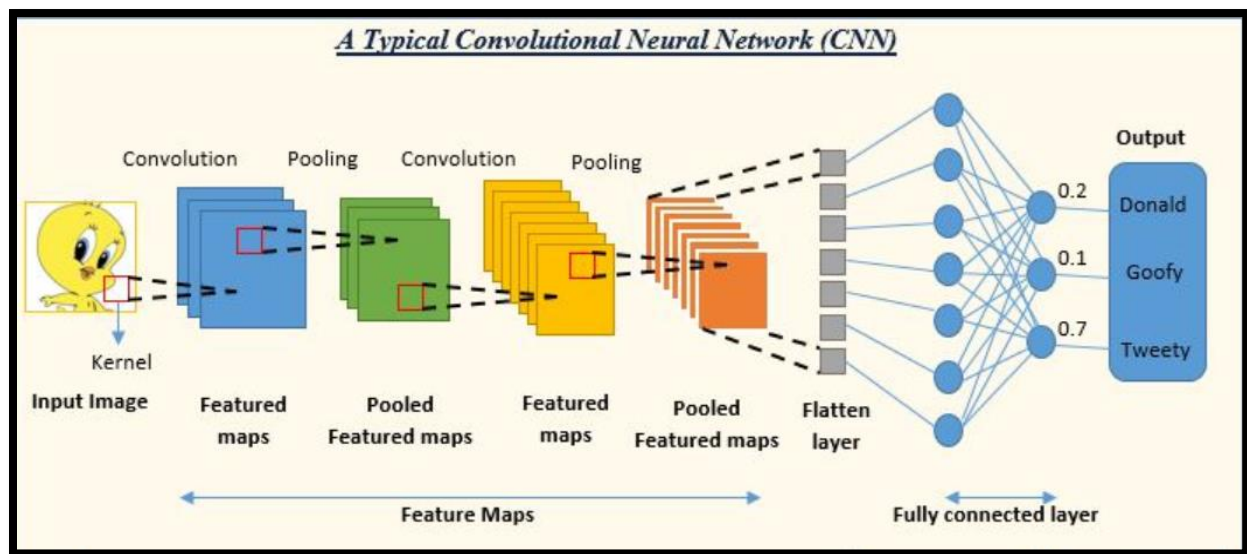


Fig 5: CNN Model

In our project, CNN model is implemented by importing essential libraries like numpy, tensorflow and keras. The `numpy.random.random()` and `numpy.random.randint()` functions are used to create the training data. The input samples (`X_train`) and related labels (`y_train`) make up the generated data.

The layers of the Keras Sequential API are used to define the model architecture. `Conv1D` With 32 filters and a kernel size of 3, represents a 1D convolutional layer. Rectified Linear Unit, or ReLU, is the activation function that is used. The shape of a single training example is represented by the `input_shape` setting of (7138, 1). `MaxPooling1D` layer uses a pool size of 2 layers to perform max-

pooling. The output of the preceding layer is flattened into a 1D array by this layer. Fully connected and dense layer has 64 units and ReLU activation. Layers.Dense has four units (for four classes) and a softmax activation function that transforms the output into probabilities. The training data (X_train and y_train) are used to train the model. Training is done with a batch size of 32 across 5 epochs. The weights of the model are modified throughout training in order to reduce the desired loss. Finally, the accuracy of the model is evaluated.

```
Epoch 1/5
172/172 [=====] - 25s 139ms/step - loss: 1.6414 - accuracy: 0.2560
Epoch 2/5
172/172 [=====] - 25s 143ms/step - loss: 1.3249 - accuracy: 0.3916
Epoch 3/5
172/172 [=====] - 24s 143ms/step - loss: 1.0811 - accuracy: 0.6102
Epoch 4/5
172/172 [=====] - 24s 139ms/step - loss: 0.6037 - accuracy: 0.8373
Epoch 5/5
172/172 [=====] - 24s 137ms/step - loss: 0.2256 - accuracy: 0.9647
```

Fig 6: CNN Implementation

3.6 Summary

In the above chapter, audio model is implemented by the above steps like data preprocessing, audios and feature extraction and using CNN to predict whether the speaker is speaking or not. After training and testing the model the accuracy of the model will predicted and will be further tuned improve the accuracy in order to get the best results.

```
# Example usage:
audio_file = '/content/drive/MyDrive/audios/Agents of Secret Stuff.mp4'
predicted_class = predict_audio_class(audio_file)

print('Predicted class:', predicted_class)
```

```
<ipython-input-38-f188030f32f6>:15: UserWarning: PySoundFile failed. Trying audioread instead.
  audio, sr = librosa.load(audio_file, sr=sample_rate, duration=duration)
/usr/local/lib/python3.10/dist-packages/librosa/core/audio.py:184: FutureWarning: librosa.core.audio.__audioread_load
  Deprecated as of librosa version 0.10.0.
  It will be removed in librosa version 1.0.
  y, sr_native = __audioread_load(path, offset, duration, dtype)
1/1 [=====] - 0s 109ms/step
Predicted class: ['SPEECH WITH MUSIC']
```

Fig 7: Predicted Class Output

Chapter 4

Video Model

4.1 Introduction

This chapter will give the overall view of how the video model is implemented. Video model distinguishes between active and non-active speakers by detecting facial features and expressions with the help of Retina Face library. This chapter will define the steps that has been taken to implement this model in detail like how the live video is processed, and facial features are extracted to get the desired output.

4.2 Video Preprocessing

Video processing has been carried out to implement this model. The video model takes input frames of the video and compares two consecutive frames in order to detect whether the speaker is active or not. The difference between the two frames is found in order to capture the motion. The output of the video model is a rectangular box formed on the face of the active speaker.

- `def read_image (frame1, frame2):` This `read_image` function defined takes two parameters, `frame1` and `frame2`, which are file paths to image files.
- `cv2.imread (frame1) = img1` uses OpenCV's `imread` function to read the image supplied by the `frame1` file location and assigns the result to the variable `img1`.
- `cv2_imshow (img1):` This line uses the `cv2_imshow` function, which is frequently used in Google Colab, to display the picture `img1` on the screen.
- `img2 = cv2.imread (frame2):` The image supplied by the `frame2` file location is read and displayed.
- The code then outputs images `img1` and `img2`.



Fig 8: Video Processing

4.3 Feature Extraction

In the context of videos, the process of choosing and converting pertinent information from raw video data into a condensed and useful representation is referred to as feature extraction. Videos are made up of a series of frames, each of which has a wide range of pixel values. The goal of feature extraction is to identify these frames' key traits so that analysis, classification, or other machine learning tasks may be completed quickly.

Retina-Face library is used for face detection and for extracting features. . It is a deep learning-based face identification and alignment library that specializes in recognizing faces in images with different sizes. It is intended to perform effectively with high-resolution images, making it appropriate for applications such as facial recognition.

It takes frames of the video as input and performs pixel-wise face localization on various scales of the faces. After analyzing, it returns facial area co-ordinates and some landmarks (eyes, nose and mouth) with a score.

```
{'score': 0.9996041655540466, 'facial_area': [189, 31, 219, 73], 'landmarks': {'right_eye': [196.77998, 48.62]
{'score': 0.99954754114151, 'facial_area': [335, 33, 364, 68], 'landmarks': {'right_eye': [341.8558, 47.88452]
{'score': 0.9995457530021667, 'facial_area': [464, 31, 493, 70], 'landmarks': {'right_eye': [471.49368, 47.20]
{'score': 0.999426007270813, 'facial_area': [63, 36, 90, 72], 'landmarks': {'right_eye': [71.21129, 50.47539]
{'score': 0.9996041655540466, 'facial_area': [189, 31, 219, 73], 'landmarks': {'right_eye': [196.77998, 48.62]
{'score': 0.99954754114151, 'facial_area': [335, 33, 364, 68], 'landmarks': {'right_eye': [341.8558, 47.88452]
{'score': 0.9995457530021667, 'facial_area': [464, 31, 493, 70], 'landmarks': {'right_eye': [471.49368, 47.20]
{'score': 0.999426007270813, 'facial_area': [63, 36, 90, 72], 'landmarks': {'right_eye': [71.21129, 50.47539]
```

Fig 9: Facial Area Co-ordinates and Landmarks

4.4 Result Analysis

When the face and the facial features are detected, separate matrices of landmarks of every face in each frame are formed. The difference between the two matrices of subsequent frames are calculated and the resulting matrix is compared with the threshold. If the resulting matrix is greater than the threshold the speaker is active. If the resulting matrix is less than the threshold then the speaker is non-active.

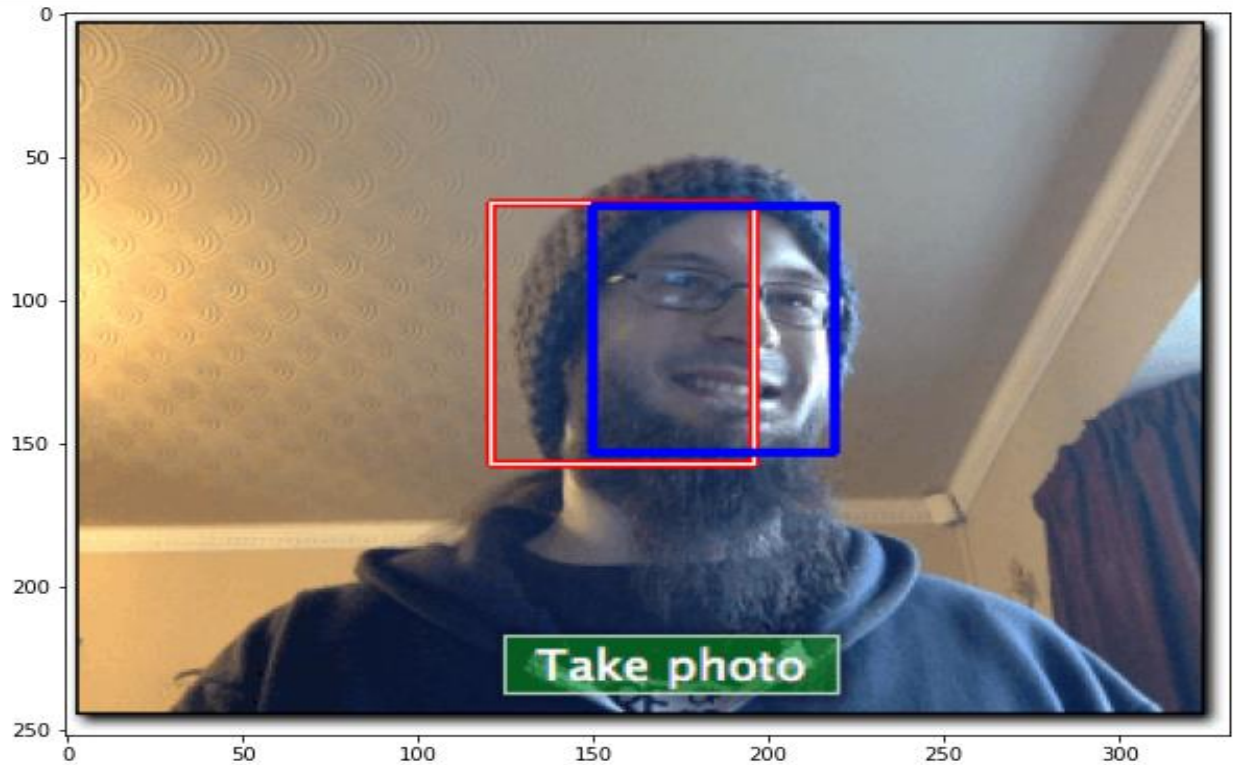


Fig 10: Result Analysis

4.5 Summary

Meetings are a common activity that provide certain challenges when creating systems that assist them. Such is the case of the speaker recognition systems using both audio and video model. In the above chapter video model is implemented by performing the above steps as discussed above and the model can predict the facial features ad landmarks of the speaker's face that helps to predict that whether the speaker is active or not.

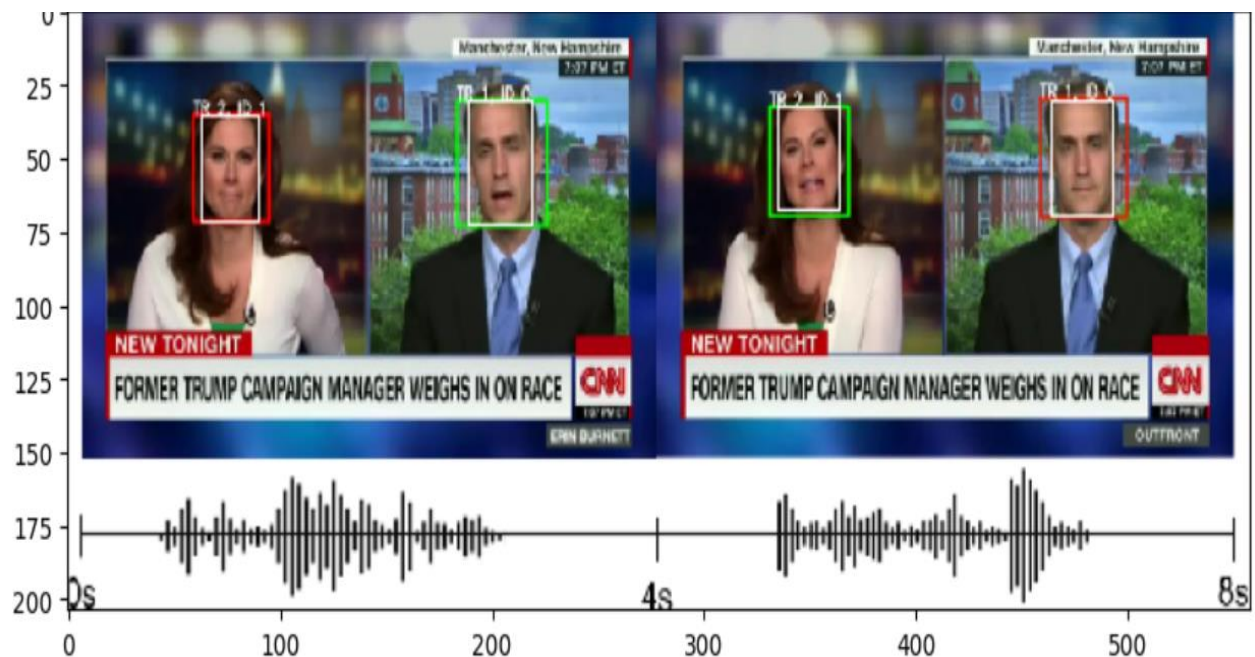


Fig 11: Predicted Output

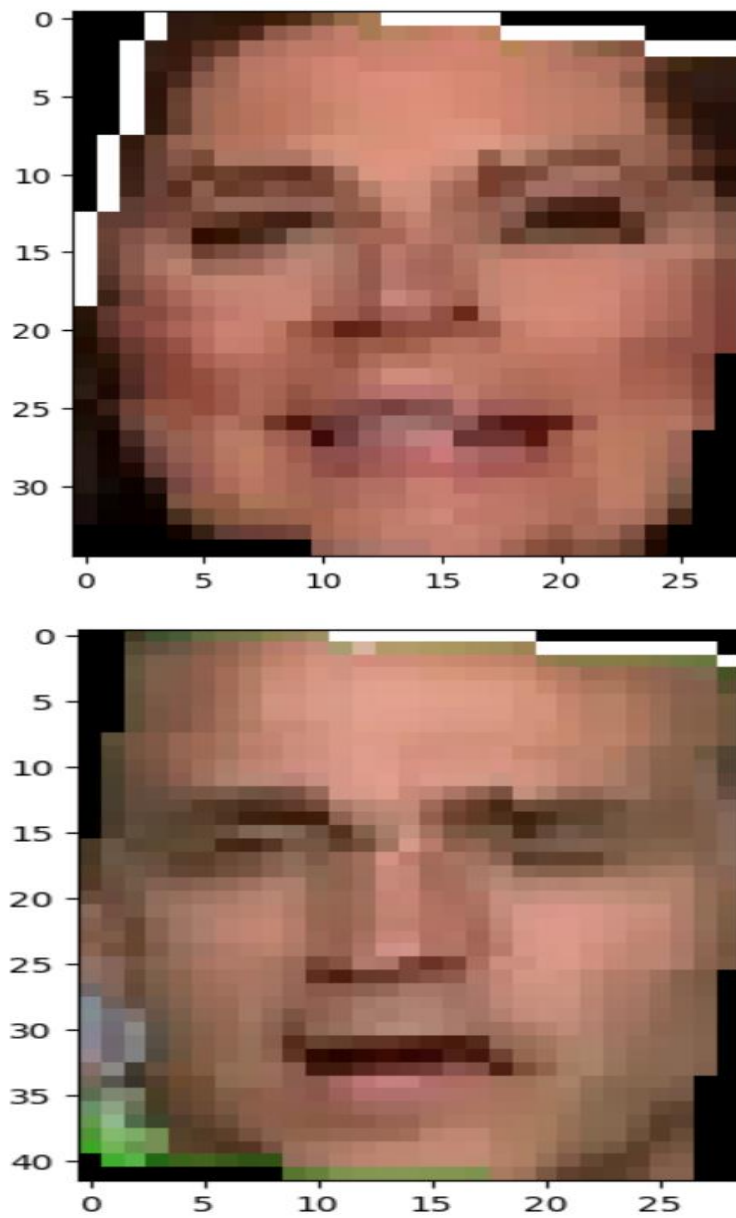


Fig 11: Predicted Output

Chapter 5

Speaker Change Model

5.1 Introduction

This chapter will give the overall view of how speaker change model is implemented. The model uses speaker diarization and other audio preprocessing techniques to determine when the speaker in the audio changes. This chapter will define the steps that has been taken to implement this model in detail like how the audio is preprocessed, and features are extracted to get the desired output.

5.2 Voice Detection

Signal processing method called Voice Activity Detection (VAD) is used to locate areas of audio data where human speech is present. It is a crucial part of many applications that process audio, including voice communication systems, audio compression, speech recognition, and more.

VAD attempts to differentiate between speech and non-speech (quiet or noisy) parts of an audio source. Systems can concentrate their processing on analysing and interpreting the speech content while disregarding or minimizing the processing of non-speech segments by accurately identifying when speech is present.

Voice Activity Detection function analyzes the provided audio data for speech activity. To identify whether or not each frame contains voice activity, it calculates the RMS (Root Mean Square) energy of the audio frames and compares it with a predefined threshold. A boolean array representing speech activity for each frame is returned by the function, that determines whether a person is audible or not.

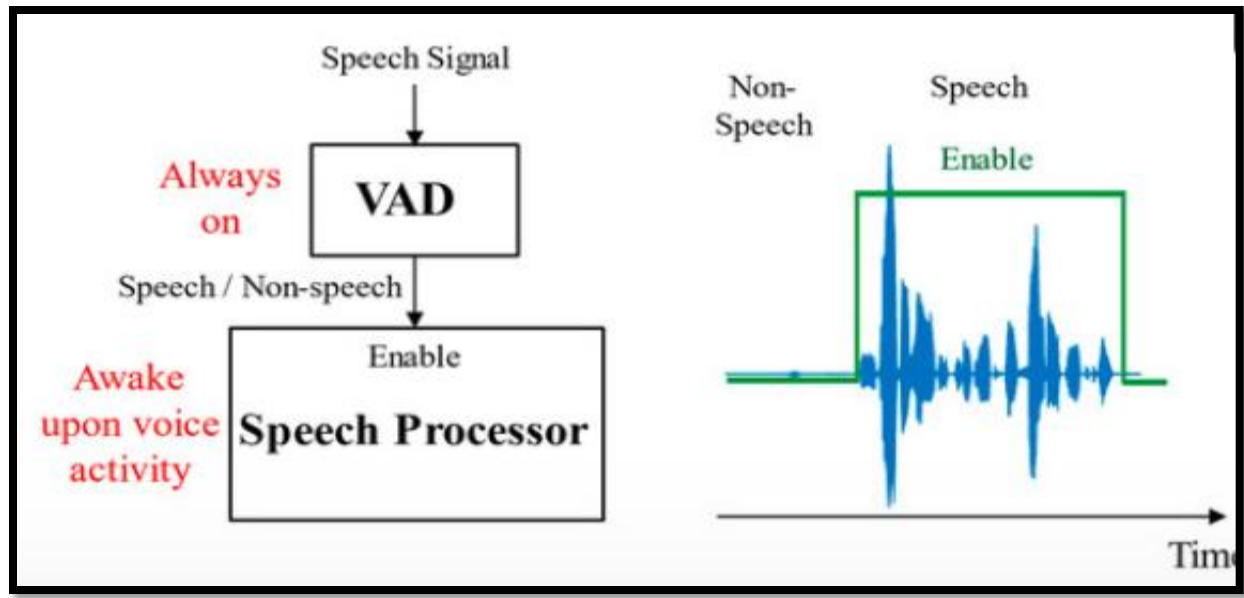


Fig 12: Voice Activity Detection

5.3 Clustering

A machine learning approach called clustering is used to group together similar data points in a way that makes them more similar to one another than to those in other groupings. Without prior knowledge of the class labels, clustering aims to find patterns or structures within a collection based on the similarity of data points.

In simple words, clustering aids in the grouping of a collection of data points into meaningful clusters where data points are more similar to one another within the same group and less similar to one another within different groups. Customer segmentation, picture segmentation, document categorization, and other processes are frequently carried out using it.

The Bayesian Information Criterion (BIC) is used to calculate the ideal number of clusters for Gaussian Mixture Models (GMM) using the function `determineOptimalClusters`. The correct number of clusters is critical for effective modelling of the data distribution in many applications, including speaker diarization, picture segmentation, and more, where this procedure is essential.

The purpose of this function is to calculate the ideal number of clusters for GMM-based speaker diarization. Bayesian Information Criterion (BIC) scores for various cluster counts is calculated as it is essential for achieving accurate and meaningful results. It plots the BIC score graph so that optimal number of clusters can be determined. Input parameters include:

- mfcc: The input data that must be used to establish the ideal number of clusters. It might be a feature matrix that was taken from audio or some other type of data.
- MaxClusters: The most clusters that will be taken into account throughout the optimization process.
- Cluster Numbers Loop: The function loops through a set of cluster numbers between 2 and maxClusters+1. The following actions are taken for each cluster with the number n_clusters:
 - A diagonal covariance structure (each component has its own diagonal covariance matrix) and n_clusters components are used to generate a Gaussian Mixture Model (GMM). The GMM is then fitted to the input data mfcc using the fit method.
 - Calculate BIC: For the fitted GMM, the Bayesian Information Criterion (BIC) is computed. The BIC levies fines.

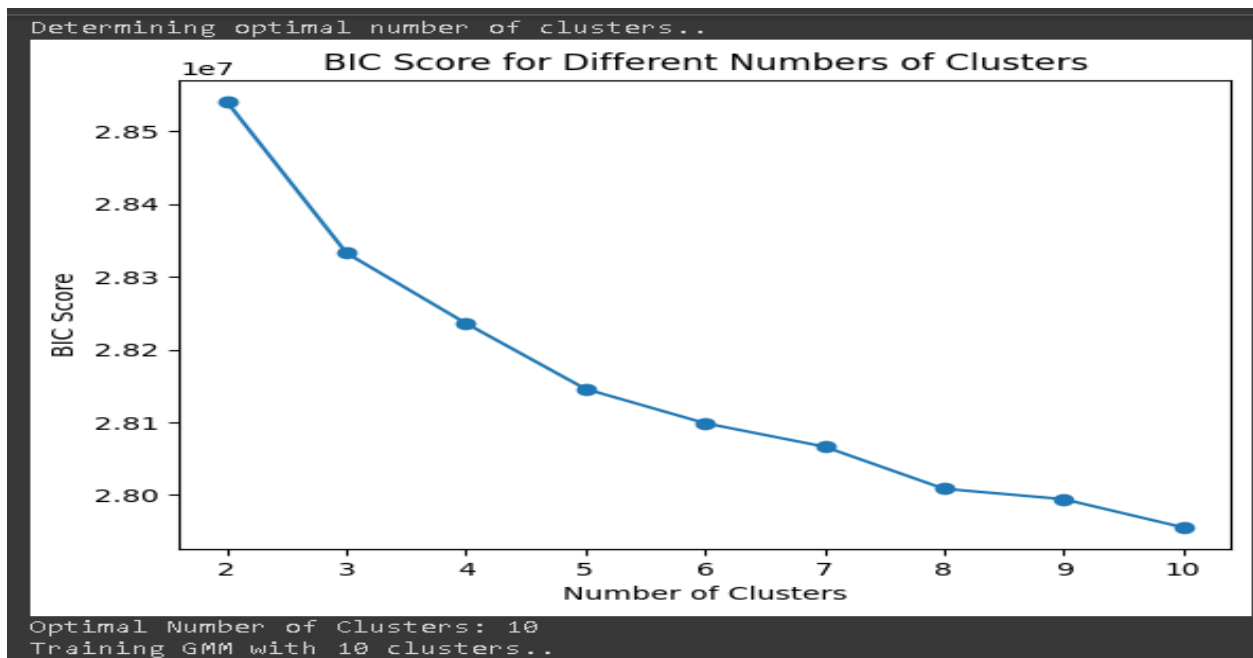


Fig 13: BIC Score Analysis

5.4 Training GMM

Gaussian Mixture Model is the abbreviation. It is a probabilistic model that is used to depict complex data distributions as a mashup of various Gaussian (normal) distributions. In many disciplines, including statistics, pattern recognition, and machine learning, GMMs are often employed.

A Gaussian Mixture Model, to put it simply, is a means of combining different Gaussian distributions to represent a dataset. A cluster or component of the data is represented by each Gaussian distribution. GMMs are particularly helpful when working with data that is a composite of multiple underlying patterns rather than belonging to a single, clearly defined cluster.

This function loads the audio data, extracts the MFCC features, and then trains a GMM using the `determineOptimalClusters` function's recommended cluster size. Agglomerative Clustering is also carried out on the probabilities of segments belonging to various clusters. The function returns both the cluster assignments and the segment likelihoods.

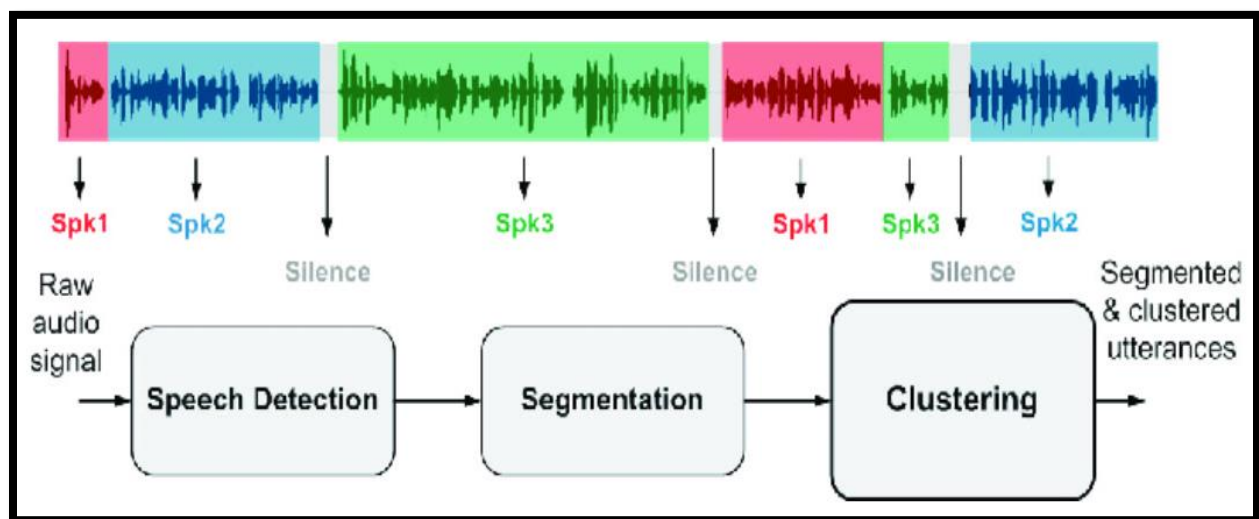


Fig 14: Speaker Diarization using GMM

5.5 Segmentation

Segmentation is a continuous signal or dataset that is segmented into smaller segments or pieces as part of the signal processing or data analysis processes. Each segment is a subset of the original data and is frequently examined independently. Time-series data, image analysis, audio processing, and other areas all frequently use segmentation.

This function requires the segment length, frame rate, and overall number of frames in addition to the cluster assignments from the GMM-based clustering. It creates a binary array where each frame is associated with a specific cluster label by assigning cluster labels to individual frames. The parameters used are defined below:

- **clust:** The clustering assignments for each data segment are represented by this array. The cluster assignment for a certain segment corresponds to each element of the clust array.
- **segLen:** This parameter specifies the required number of frames or samples for each segment.
- **frameRate:** The number of frames or samples processed in a given amount of time.
- **numFrames:** Number of frames in the original data.

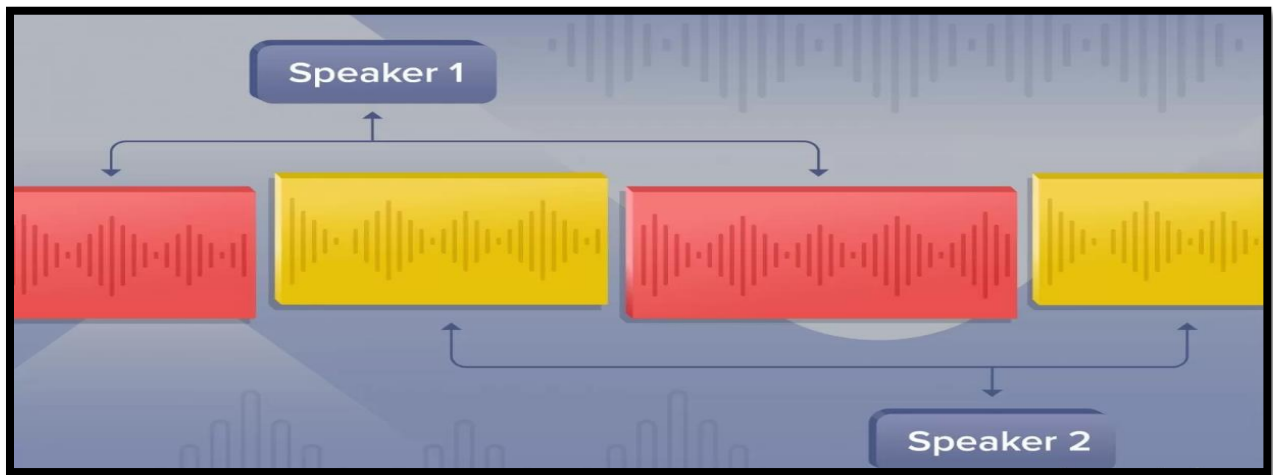


Fig15: Segmentation

5.6 Speaker Diarization

The technique of segmenting and labelling an audio recording into discrete segments based on the different speakers present in the recording is known as speaker diarization. Speaker diarization aims to recognize and differentiate between various speakers in a given audio stream, attributing each speech segment to a particular speaker.

The word "diarization" is derived from the word "diary," as the procedure is comparable to recognizing and differentiating between various "voices" or "speakers" in an audio recording, similar to recognizing various individuals in a conversation.

Segmentation: The audio recording needs to be divided up into more manageable, relevant chunks as the initial step in speaker diarization. Techniques like Voice Activity Detection (VAD), which recognizes speech in the audio, can be used for this. Segmentation aids in separating sounds into chunks that most likely belong to different speakers.

Extracting Features: Relevant features are taken from the audio for each segment. Mel-frequency cepstral coefficients (MFCCs), which indicate the spectral content of the audio, are typical features. Pitch, timbre, and rhythm are captured by these traits, which are helpful for differentiating across voices.

Clustering: Following the extraction of the characteristics, clustering methods are used to combine related segments. The segments in each cluster are most likely from the same speaker. The well-known clustering techniques are k-means.

The function `speakerdiarisationdf` appears to be a part of a larger speaker diarization process, aiming to convert the results of speaker diarization into a structured DataFrame containing information about the identified speakers and their speech segments. Let's break down the code step by step to understand its purpose and functionality:

Input Parameters:

- `hyp`: An array containing the hypothesis of speaker labels (likely generated through speaker diarization).
- `frameRate`: The frame rate of the audio processing.
- `wavFile`: The path or name of the audio file being processed.

Output:

`spdatafinal`: A DataFrame that contains information about the speaker segments, including audio name, speaker label, start time, and end time.

Explanation:

- **Initialization**: Initialize lists (`audioname`, `starttime`, `endtime`, `speakerlabel`) to hold information about each speaker segment.
- **Finding Speaker Change Points**: Identify indices where the speaker label changes (`spkrChangePoints`) by comparing consecutive elements of the `hyp` array.
- **Identifying Speaker Labels**: Create a list of speaker labels (`spkrLabels`) based on the identified change points. This list will represent the unique speakers detected in the audio.
- **Creating Speaker Segment Information**: Iterate through the `spkrLabels` list. For each unique speaker, gather information about their segments:
 - Append the audio name derived from `wavFile`.
 - Calculate the start time by converting the frame index to time using the `frameRate`.
 - Calculate the end time by finding the difference between the next speaker's start time and the current speaker's start time.
 - Create a speaker label string ("Speaker X") for the DataFrame.
- **Creating the DataFrame `speakerdf`**: Create a DataFrame named `speakerdf` using the collected information (audio name, start time, end time, speaker label).
- **Creating the Final DataFrame `spdatafinal`**: Initialize an empty DataFrame named `spdatafinal` to hold the final structured speaker segment information.
- **Iterate through the rows of `speakerdf` using `itertuples`**: If it's the first row, initialize variables to track speaker label, start time, and end time.
- If the current speaker label matches the previous one (`spfind`), update the end time.

- If the speaker label changes, add a row to `spdatafinal` containing the previous speaker's information.
- Increment the row index (`k`).
- Update `spfind`, `stime`, and `etime` for the new speaker segment. Update iteration counters (`i`, `j`).
- Returning the Final DataFrame: Return the `spdatafinal` DataFrame containing the structured speaker segment information. purpose of this function is to build a Data Frame that represents speaker diarization by processing the clustering findings. Based on the results of the clustering, it determines speaker change points, separates the audio into speaker segments, and generates a Data Frame containing the name of the audio file, the start and end times, and the speaker label.

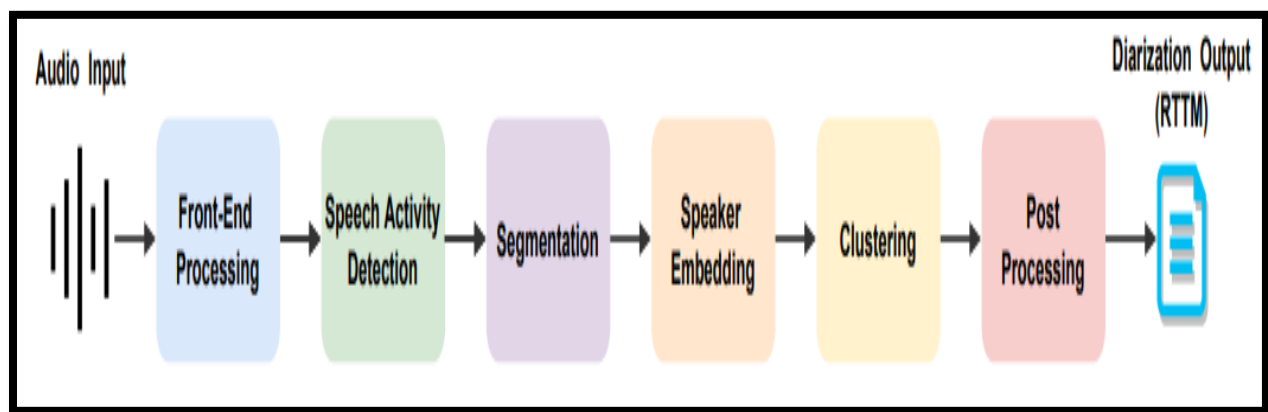
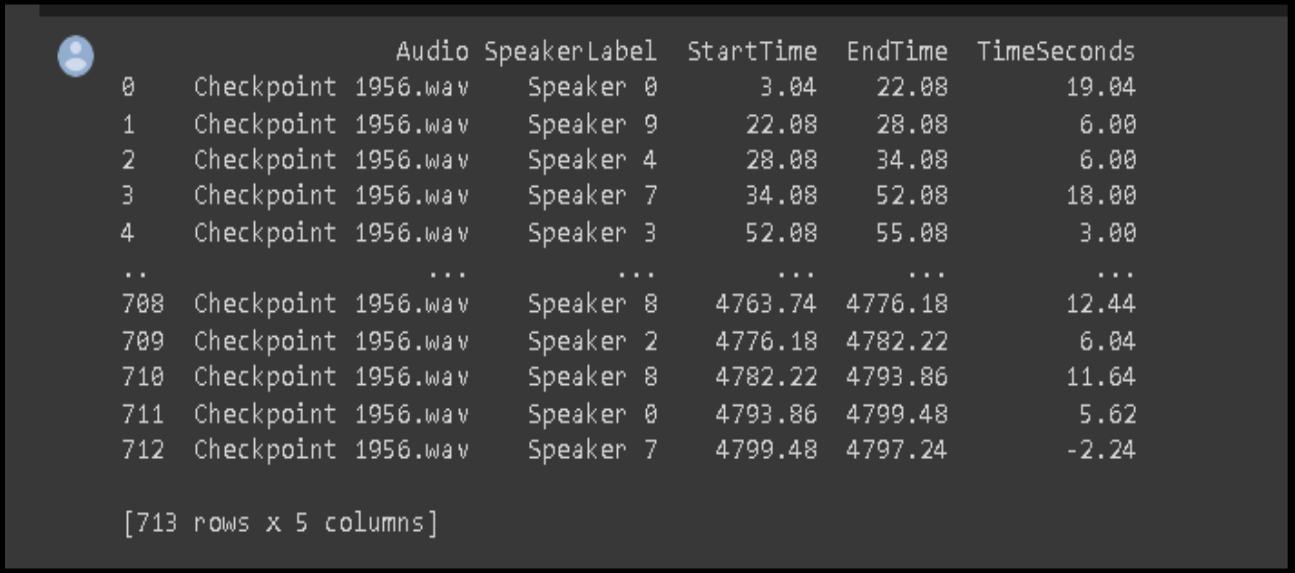


Fig 16: Speaker Diarization System

5.7 Summary

The speaker diarization technique has been implemented segmenting an audio recording into unique segments depending on speaker identity. To accomplish this, it makes use of a number of approaches, including feature extraction, clustering, segmentation and speech activity detection.



A terminal window with a dark background and a blue user icon in the top-left corner. It displays a table of predicted output for speaker diarization. The table has 7 columns: an index, a segment type, an audio file path, a speaker label, start and end times, and duration. The data shows segments for 'Checkpoint 1956.wav' assigned to various speakers (0, 9, 4, 7, 3, 8, 2, 8, 0, 7) with corresponding time intervals. The last row shows a negative duration, likely indicating a segment ending before it started. At the bottom, it states the total dimensions of the data as [713 rows x 5 columns].

		Audio	SpeakerLabel	StartTime	EndTime	TimeSeconds
0	Checkpoint	1956.wav	Speaker 0	3.04	22.08	19.04
1	Checkpoint	1956.wav	Speaker 9	22.08	28.08	6.00
2	Checkpoint	1956.wav	Speaker 4	28.08	34.08	6.00
3	Checkpoint	1956.wav	Speaker 7	34.08	52.08	18.00
4	Checkpoint	1956.wav	Speaker 3	52.08	55.08	3.00
..	
708	Checkpoint	1956.wav	Speaker 8	4763.74	4776.18	12.44
709	Checkpoint	1956.wav	Speaker 2	4776.18	4782.22	6.04
710	Checkpoint	1956.wav	Speaker 8	4782.22	4793.86	11.64
711	Checkpoint	1956.wav	Speaker 0	4793.86	4799.48	5.62
712	Checkpoint	1956.wav	Speaker 7	4799.48	4797.24	-2.24

[713 rows x 5 columns]

Fig 17: Predicted Output

Chapter 6

Fusion of Audio Visual Model

6.1 Introduction

This chapter will give the overall view of how audio classification model, speaker change model and video model are integrated to get the desired output. The output is determined when the scores of audio model is fused with video model. This chapter will define the steps that has been taken to integrate the models to determine whether the speaker is active or not.

6.2 Integration of Audio Models

Integration of audio models begins by importing all the necessary libraries. Then the pre-trained audio classification model is loaded that is stored in a separate file. The audio input file is loaded, that is preprocessed and its features are extracted to determine whether the audio is audible or not. The audio class is predicted using the pre-trained model and the class with highest probability is identified. The predicted class label is then returned determining the output.

If the predicted class is not 'NO_SPEECH', then speaker diarization technique is applied that segments the audio into multiple segments that corresponds to different speakers. The resulting speaker segments and their duration of speech are displayed as an output. If the predicted class is 'NO_SPEECH' then the message indicating that no speech is detected is displayed.



Fig 18: Result of Audio Model

6.3 Score Based Fusion

Score-based fusion, also known as decision-level fusion, is a method for combining the results or decisions from various individual classifiers, models, or sources to arrive at a final judgment or prediction. It is used in a variety of fields, including pattern recognition, machine learning, and information retrieval. By utilizing the varied information offered by many sources, score-based fusion aims to increase the overall performance, accuracy, or robustness of a system.

Once the audio models are integrated, the predicted class with the highest probability along with its label is displayed. The speaker change model is implemented that determines the starting and ending time of the speakers. The score of the audio classification model is then calculated that is fused with the score obtained by the video model.

6.4 Integration of Audio Visual Model

Integration of audio model and video model begins when the scores obtained by both the models are fused. The obtained scores are then compared with the threshold value to find the active speaker, in order to eradicate the conflict between the two active speakers. The max active speaker score is then calculated and the camera is panned on that predicted active speaker based on the max score.

6.5 Summary

The main goal of audio visual models is to find the active speaker with the highest probability so that the conflict between the two active speakers is resolved. For this purpose, audio classification model and speaker change model is implemented and their scores are calculated. Then the visual model is implemented and its score is calculated which is then fused with the score of audio model. The maximum of the scores helps us to determine the final active speaker.



Fig 19: Desired Outputs

6.6 Predicted Outputs





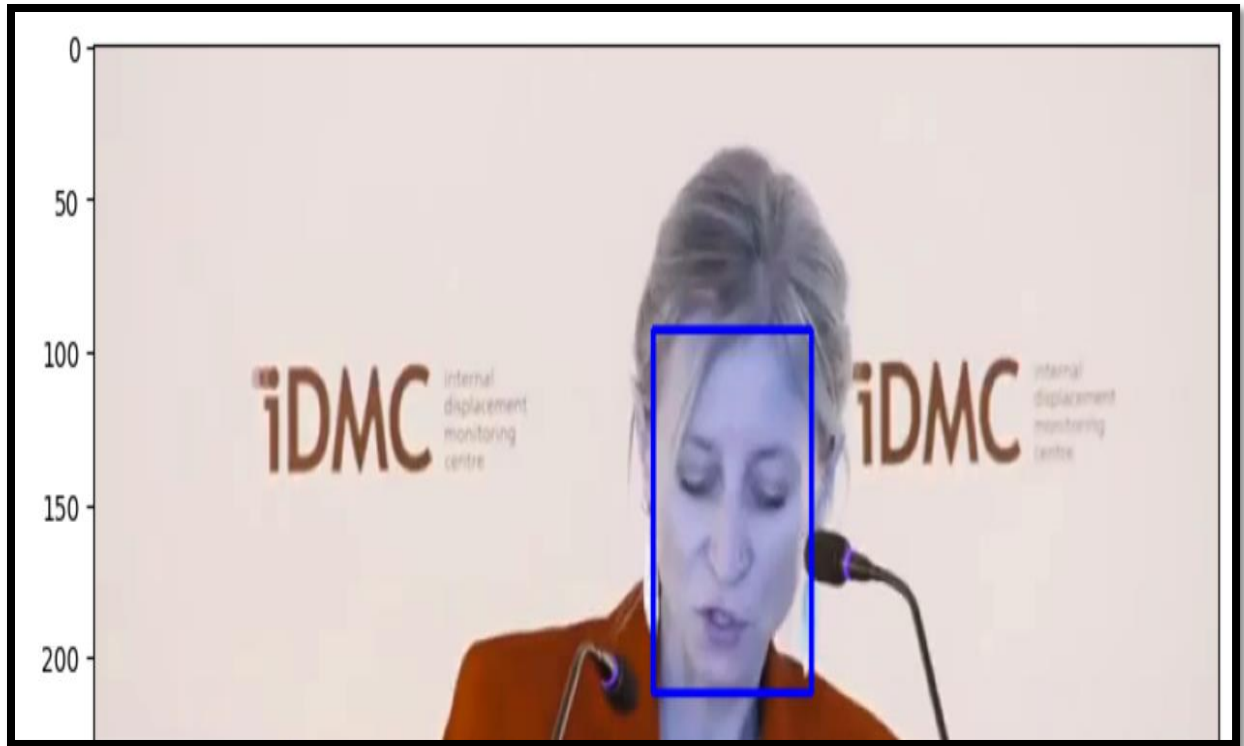


Fig 20: Predicted Output



```
{ 'score': 0.9997042417526245, 'facial_area': [495, 88, 577, 203], 'landmarks': { 'right_eye': [506.2542, 131.24667], 'left_eye': [537.6214, 132.37804], 'nose': [509.06854, 152.53859]
{ 'score': 0.9993885159492493, 'facial_area': [314, 92, 390, 211], 'landmarks': { 'right_eye': [360.69482, 139.94434], 'left_eye': [381.32208, 137.59882], 'nose': [386.85214, 158.882]
{ 'score': 0.9992890357971191, 'facial_area': [76, 101, 160, 218], 'landmarks': { 'right_eye': [115.4335, 145.84204], 'left_eye': [150.26459, 146.15878], 'nose': [145.48767, 166.6697]
{ 'score': 0.9972785711288452, 'facial_area': [475, 351, 487, 368], 'landmarks': { 'right_eye': [477.22302, 357.38828], 'left_eye': [479.01352, 357.69574], 'nose': [475.1915, 360.732]
{ 'score': 0.9947344064712524, 'facial_area': [702, 116, 754, 214], 'landmarks': { 'right_eye': [712.4679, 157.22998], 'left_eye': [712.5336, 157.24727], 'nose': [704.0568, 174.95497]
{ 'score': 0.9821575284004211, 'facial_area': [541, 347, 554, 369], 'landmarks': { 'right_eye': [545.91626, 354.6892], 'left_eye': [545.91547, 354.9656], 'nose': [542.33765, 358.5287]
{ 'score': 0.979114294052124, 'facial_area': [357, 360, 367, 373], 'landmarks': { 'right_eye': [363.3419, 365.439], 'left_eye': [365.9355, 364.58536], 'nose': [366.4306, 367.38132],
{ 'score': 0.9997042417526245, 'facial_area': [495, 88, 577, 203], 'landmarks': { 'right_eye': [506.2542, 131.24667], 'left_eye': [537.6214, 132.37804], 'nose': [509.06854, 152.53859]
{ 'score': 0.9993885159492493, 'facial_area': [314, 92, 390, 211], 'landmarks': { 'right_eye': [360.69482, 139.94434], 'left_eye': [381.32208, 137.59882], 'nose': [386.85214, 158.882]
{ 'score': 0.9992890357971191, 'facial_area': [76, 101, 160, 218], 'landmarks': { 'right_eye': [115.4335, 145.84204], 'left_eye': [150.26459, 146.15878], 'nose': [145.48767, 166.6697]
{ 'score': 0.9972785711288452, 'facial_area': [475, 351, 487, 368], 'landmarks': { 'right_eye': [477.22302, 357.38828], 'left_eye': [479.01352, 357.69574], 'nose': [475.1915, 360.732]
{ 'score': 0.9947344064712524, 'facial_area': [702, 116, 754, 214], 'landmarks': { 'right_eye': [712.4679, 157.22998], 'left_eye': [712.5336, 157.24727], 'nose': [704.0568, 174.95497]
{ 'score': 0.9821575284004211, 'facial_area': [541, 347, 554, 369], 'landmarks': { 'right_eye': [545.91626, 354.6892], 'left_eye': [545.91547, 354.9656], 'nose': [542.33765, 358.5287]
{ 'score': 0.979114294052124, 'facial_area': [357, 360, 367, 373], 'landmarks': { 'right_eye': [363.3419, 365.439], 'left_eye': [365.9355, 364.58536], 'nose': [366.4306, 367.38132],
```





Fig 21: Predicted Outputs

Chapter 7

Libraries

7.1 Introduction

Libraries refer to pre-built collections of functions, algorithms, and tools that facilitate the development, implementation, and experimentation of machine learning models and techniques. These libraries provide a set of ready-to-use functionalities, making it easier for developers and researchers to work on complex machine learning tasks without having to implement everything from scratch.

7.2 Libraries used in the Project

Libraries play a crucial role in accelerating the development and deployment of machine learning projects. The libraries used in our project are as followed:

- **Pandas (import pandas as pd):** Pandas is a cornerstone library for data manipulation and analysis. It introduces two essential data structures: **Series** and **DataFrame**. **Series** represents a one-dimensional labeled array, while **DataFrame** is a two-dimensional labeled table. These structures enable intuitive handling and manipulation of structured data, including operations like filtering, grouping, joining, and reshaping. Pandas is instrumental for data preprocessing, exploration, and transformation in various machine learning and data analysis projects.
- **Numpy (import numpy as np):** Numpy is a fundamental numerical computing library that provides support for arrays and matrices. It forms the foundation for scientific and mathematical computations in Python. Numpy arrays are homogeneous and efficiently handle large datasets. The library includes a wide range of mathematical functions for array operations, linear algebra, Fourier transforms, and more. Its core data structure, the **ndarray**, supports both element-wise operations and advanced array manipulations.

- **Scipy (import scipy):** Built upon Numpy, Scipy extends its capabilities into various scientific domains. The library encompasses modules for optimization, integration, interpolation, signal processing, statistics, and more. Scipy's extensive toolkit is invaluable for tackling advanced mathematical and scientific problems. For instance, the integration module allows numerical integration of functions, and the signal processing module offers functions for filtering, convolution, and spectral analysis.

- **Scipy's wavfile submodule (from scipy.io import wavfile):** The wavfile submodule in Scipy enables reading and writing audio files in the WAV format. WAV files are widely used to store uncompressed audio data. The submodules read function loads WAV files, returning sample rate and audio data as a Numpy array. This is critical for processing audio data, such as in speech recognition or sound analysis.

- **Scipy's fftpack submodule (import scipy.fftpack as fft):** The fftpack submodule facilitates discrete Fourier transform computations. The Discrete Fourier Transform (DFT) transforms a signal from the time domain to the frequency domain, enabling analysis of frequency components. The Fast Fourier Transform (FFT), a faster implementation of DFT, allows efficient computation of large transforms, used in applications like audio spectrum analysis and signal filtering.

- **Scipy's get_window function (from scipy.signal import get_window):** The get_window function aids in designing windows for signal processing. Windows are used to taper or shape a signal before applying spectral analysis techniques like the FFT. Common windows include Hamming, Hanning, and Blackman. Proper windowing minimizes spectral leakage and enhances the accuracy of frequency domain analysis.

- **IPython's display submodule (import IPython.display as ipd):** IPython's display submodule enhances interactive computing environments by enabling multimedia content presentation directly within the interface. For instance, using ipd.Audio allows the playback of audio files, making it a useful tool for audio analysis and debugging.

- `Matplotlib` (`import matplotlib.pyplot as plt`): `Matplotlib` is a comprehensive visualization library, capable of creating a wide range of plots, graphs, and charts. Its flexible API empowers users to craft interactive and publication-quality visualizations. From simple line plots to complex 3D figures, `Matplotlib` supports visualization tasks across various scientific disciplines and data analysis projects.
- `TensorFlow` (`import tensorflow as tf`): `TensorFlow` is a powerful deep learning framework developed by Google. It provides a versatile ecosystem for building and training machine learning models, especially deep neural networks. `TensorFlow` offers both high-level APIs for rapid model development and low-level operations for fine-grained control.
- `Keras` from `TensorFlow` (`from tensorflow import keras`): `Keras`, an integral component of `TensorFlow`, simplifies the creation and training of neural networks. Its high-level API allows developers to define neural network architectures using intuitive building blocks called layers. `Keras` abstracts many complexities, making deep learning more accessible.
- `load_model` from `Keras` (`from tensorflow.keras.models import load_model`): `Keras`' `load_model` function enables the loading of pre-trained neural network models. This is crucial for reusing models, performing evaluations, and making predictions without retraining.
- `os` (`import os`): The `os` module provides functionalities for interacting with the operating system. It supports tasks such as managing directories, creating files, and navigating file paths. It's indispensable for handling files and directories in machine learning pipelines.
- `pickle` (`import pickle`): The `pickle` module facilitates serialization and deserialization of Python objects. It's used to store and retrieve complex objects, such as trained models or data structures. Pickling allows objects to be saved to disk and restored later.
- `warnings` (`import warnings`): The `warnings` module helps manage warnings generated during program execution. It can suppress or filter out certain warnings, ensuring that developers can focus on relevant information while ignoring less critical messages.

- **Librosa (import librosa):** Librosa is an audio analysis library designed for tasks like feature extraction, signal manipulation, and music information retrieval. It simplifies the process of loading audio data, computing audio features (e.g., MFCCs, chroma features), and performing transformations. Librosa is valuable for various audio analysis applications, from speech recognition to music genre classification.

- **AgglomerativeClustering from sklearn.cluster:** Agglomerative Clustering is a hierarchical clustering algorithm available in scikit-learn. It works by iteratively merging the closest clusters or data points based on a linkage criterion (e.g., "ward", "average", "complete"). This creates a hierarchical tree of clusters. The `AgglomerativeClustering` class provides the implementation for this algorithm. It takes parameters like the number of clusters to be formed and the linkage criterion to use.

- **StandardScaler from sklearn.preprocessing:** `StandardScaler` is a preprocessing technique to scale and standardize features by removing the mean and scaling to unit variance. This process ensures that features have similar scales, which can improve the performance of various machine learning algorithms. The `StandardScaler` class provides this functionality in scikit-learn.

- **Normalize from sklearn.preprocessing:** The `normalize` function from scikit-learn's preprocessing module scales individual samples to have a unit norm. This is particularly useful when you want to ensure that the magnitude of each sample vector is the same, which can be important for some distance-based algorithms.

- **GaussianMixture from sklearn.mixture:** Gaussian Mixture Model (GMM) is a probabilistic model often used for clustering and density estimation. It assumes that data points are generated from a mixture of several Gaussian distributions. The `GaussianMixture` class in scikit-learn provides tools for fitting GMMs to data. It allows you to specify the number of components (Gaussians) and the type of covariance matrix to use.

- `silhouette_score` from `sklearn.metrics`: Silhouette Score is a metric used to evaluate the quality of clusters. It measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). Higher values indicate well-separated clusters. The `silhouette_score` function computes the silhouette score for a given clustering.
- `davies_bouldin_score` from `sklearn.metrics`: Davies-Bouldin Score is another metric for evaluating clustering results. It measures the average similarity between each cluster and its most similar cluster, considering both the distance and scatter within the clusters. Lower values indicate better clustering. The `davies_bouldin_score` function calculates this score.
- `pyplot` from `matplotlib`: The `pyplot` module from the `matplotlib` library provides an interface similar to MATLAB for creating various types of plots and visualizations. It's commonly used for data visualization in a wide range of scientific and engineering applications. Functions from this module allow you to create line plots, scatter plots, histograms, bar plots, and more.
- `Retinaface`: Retinaface is a deep learning-based face detection model specifically designed to detect faces in images and videos. It is built on top of popular deep learning frameworks like PyTorch and TensorFlow. The uniqueness of Retinaface lies in its accuracy and ability to handle complex scenarios, such as faces of varying sizes, angles, and occlusions. It uses convolutional neural networks (CNNs) to process images and identify facial regions. This library is particularly useful for tasks like face recognition, emotion analysis, and human-computer interaction. Retinaface provides a ready-to-use interface for developers to integrate advanced face detection capabilities into their applications with ease.
- `CV2 (OpenCV)`: CV2, also known as OpenCV (Open Source Computer Vision Library), is a popular and powerful library for computer vision and image processing tasks. It offers a wide range of functionalities, from basic image manipulation (resizing, cropping, filtering) to advanced computer vision tasks (object detection, image segmentation, motion tracking). OpenCV supports various programming languages, including Python, C++, and Java. It provides efficient algorithms and tools for image and video processing, making it a go-to choice for researchers, developers, and engineers working on computer vision projects.

- **Matplotlib:** Matplotlib is a widely-used Python library for creating static, animated, and interactive visualizations in various formats. It provides an object-oriented interface for embedding plots into applications and generating publication-quality figures. With Matplotlib, you can create line plots, scatter plots, bar plots, histograms, pie charts, 3D plots, and more. Its customization options allow you to control every aspect of the visualization, including colors, labels, legends, and annotations.

These libraries collectively contribute to various stages of computer vision and image processing tasks. Retinaface specializes in accurate face detection, CV2 provides a comprehensive suite of computer vision tools, and Matplotlib empowers users to create visually appealing and informative visualization.

Chapter 8

Conclusion

8.1 Summary

Active speaker detection is a crucial task in audio processing and multimedia analysis that involves identifying and distinguishing the active speaker or speakers within an audio recording or a video. This technology finds applications in various domains, such as video conferencing, surveillance, automatic transcription, and content indexing. This report provides an in-depth overview of active speaker detection, covering its importance, methods, challenges, and potential enhancements.

Active speaker detection plays a pivotal role in improving the efficiency of various communication and multimedia systems. It enables the following:

- **Enhanced Communication:** Active speaker detection significantly enhances communication, especially in scenarios like video conferencing. In a video conference with multiple participants, identifying the active speaker allows the participants to visually focus on the person who is currently speaking. This improves the clarity of communication and reduces confusion among participants. By highlighting the speaker, it becomes easier for participants to follow the conversation, understand the context, and engage effectively.
- **Transcription and Captioning:** For automatic transcription and captioning of audio or video content, accurate identification of the active speaker is crucial. When transcribing spoken content, associating the correct text with the corresponding speaker is essential for generating accurate and coherent transcripts. Active speaker detection helps ensure that the transcribed text is properly attributed to the individuals speaking, allowing for more accurate and meaningful transcripts. This is particularly valuable for accessibility purposes and for creating searchable and understandable content.

- **Content Indexing:** Active speaker detection can be used to index and organize multimedia content based on the speakers present in the recording. By identifying the speakers at different timestamps, it becomes possible to create an index that indicates who was speaking when. This indexing can facilitate efficient content retrieval, enabling users to navigate through a recording and locate specific sections where a particular speaker was active. It's particularly useful for large audio or video databases, where searching for specific speakers or segments could otherwise be challenging.

- **Security and Surveillance:** In security and surveillance systems, active speaker detection can be a valuable tool for monitoring and analyzing events. For example, in CCTV footage, identifying who is speaking during certain incidents or conversations can provide context and insights into the situation. This can be crucial for investigations, incident analysis, and real-time monitoring. By knowing who the active speakers are in surveillance footage, security personnel can better understand interactions, conversations, and potential threats.

- **Interactive Virtual Assistants:** In the realm of virtual assistants and chatbots, accurate active speaker detection is paramount for creating a seamless and intuitive user experience. When users interact with these AI-driven systems through voice commands, it's essential that the virtual assistant can discern who is speaking. This ensures that the responses are directed to the right person, making the interaction feel personalized and natural. For instance, in a smart home environment, where multiple family members might interact with a virtual assistant, active speaker detection ensures that each family member's requests are correctly identified and addressed.

- **Education and E-Learning:** Active speaker detection holds potential for transforming remote learning experiences. In virtual classrooms or online lectures, being able to identify the current speaker is instrumental for students to follow the discussion effectively. It aids educators in emphasizing key points and maintaining student engagement. Moreover, in collaborative online learning settings, knowing who is speaking facilitates tracking individual contributions, enhancing accountability, and encouraging participation.

- **Media Production and Editing:** The applications of active speaker detection extend to the domain of media production and editing. Consider scenarios where interviews, panel discussions, or group conversations are recorded. During the editing process, automatic identification of speakers streamlines the task. Video editors can efficiently segment and organize content based on who is speaking at any given moment. This capability significantly accelerates the production workflow, resulting in well-structured, coherent, and professionally edited content.

- **Emotion Analysis and Sentiment Mining:** Active speaker detection has the potential to enrich emotion analysis and sentiment mining in multimedia content. By associating speakers with their emotional expressions, it becomes possible to analyze the sentiments conveyed by different individuals. This is particularly valuable in contexts such as debates, discussions, or customer interactions. Understanding the emotional context of each speaker provides deeper insights into the overall sentiment and mood of the conversation, enabling more nuanced analysis.

- **Public Speaking and Presentation Analysis:** In the realm of public speaking and presentations, active speaker detection offers a unique advantage. Analyzing speaking patterns, frequency, and duration of different speakers during a presentation can yield valuable insights. Speakers can gain feedback on their delivery style, interaction with the audience, and overall presentation dynamics. This information assists presenters in refining their pacing, emphasizing key points effectively, and fostering better audience engagement.

- **The multifaceted applications of active speaker detection underscore its transformative potential across various domains.** It not only enhances communication clarity and transcription accuracy but also revolutionizes content creation, sentiment analysis, and the overall quality of interactions. As technology continues to advance, active speaker detection stands as a testament to the power of AI-driven solutions in reshaping the way we engage with multimedia content and information.

8.2 Methods to Implement the Project

Active speaker detection can be accomplished through various methods, each leveraging different cues from audio and video data to determine who is currently speaking. These methods play a vital role in enhancing communication, transcription accuracy, content indexing, and security in various domains like video conferencing, surveillance, and content analysis.

- **Audio-Based Approaches:** In audio-based methods, the analysis revolves around key audio characteristics associated with speech. Voice energy, or the loudness of the audio signal, is a distinguishing factor. When someone speaks, their voice generates higher energy levels compared to background noise. This energy is often concentrated within specific frequency ranges linked to speech. Analyzing these energy levels helps identify segments with active speakers. Additionally, the pitch (fundamental frequency) of the voice varies from person to person. By tracking pitch variations, shifts between speakers can be discerned. Speaking rate, or the speed at which speech occurs, is another crucial factor. Different speakers' exhibit varying speaking rates, and detecting sudden changes in rate can indicate speaker transitions.
- **Video-Based Approaches:** Video-based methods capitalize on visual cues captured from video recordings. These cues include lip movement and facial expressions. When a person speaks, their lips move in sync with their speech. By tracking lip movement through image processing techniques, it becomes possible to identify the active speaker based on whose lips are moving. Moreover, facial expressions can provide additional insights. Analyzing facial features such as mouth and jaw movements, eyebrow positions, and eye blinks helps determine the speaker. These methods are particularly valuable when audio quality is compromised, and visual information is available.
- **Audio-Visual Fusion:** Audio-visual fusion involves combining cues from both audio and video sources to enhance accuracy and robustness. By integrating results from audio-based and video-based methods, the overall identification becomes more reliable. Fusion aims to overcome limitations of individual methods and leverage their complementary strengths. For instance, while audio-based methods may struggle in noisy environments, visual cues from video can still provide useful information. Similarly, video-based methods can aid in scenarios

where audio quality is poor. The process involves aligning audio and video data properly to ensure synchronized and accurate fusion.

In summary, active speaker detection methods are essential in multimedia analysis. Audio-based approaches leverage voice energy, pitch, and speaking rate to determine speakers based on audio cues. Video-based methods analyze lip movement and facial expressions from video data to identify the active speaker using visual cues. Audio-visual fusion combines information from both modalities to achieve heightened accuracy and robustness. The choice of method depends on factors like available data quality, the environment, and the specific requirements of the application. These techniques collectively contribute to creating more efficient communication systems, accurate transcriptions, organized content indexing, and improved security in various multimedia contexts.

Active speaker detection methods, whether audio-based, video-based, or through audio-visual fusion, are essential tools in multimedia analysis. They leverage distinctive audio and visual cues to identify the active speaker, enhancing communication, transcription accuracy, content organization, and security in diverse multimedia scenarios. The choice of method depends on the context and available data, and collectively, these techniques contribute to shaping more effective and insightful multimedia systems.

Flow of the project is as followed:

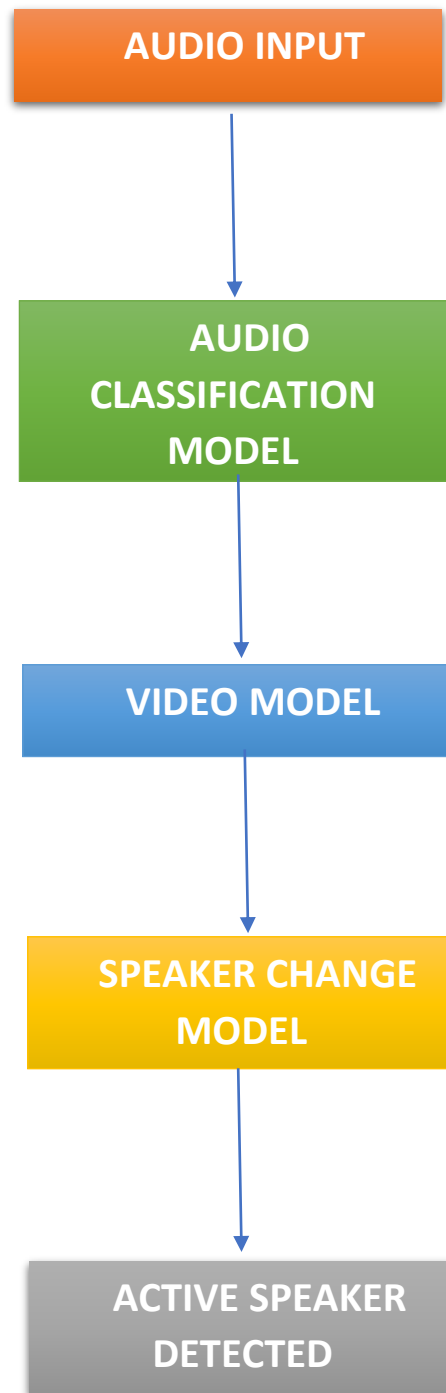


Fig 21: Flow of the Project

8.3 Challenges

Active speaker detection, while an invaluable tool in multimedia analysis, faces several challenges that must be overcome for accurate and robust performance. One of the most pressing issues is the presence of noise and variability within audio data. Background noise, ambient sounds, and variations in speaking styles can significantly hinder the process of identifying the active speaker. The task becomes even more intricate in environments with multiple speakers engaged in conversation simultaneously. The challenge lies in distinguishing the target speaker's voice amidst a cacophony of sounds, emphasizing the need for sophisticated algorithms capable of isolating the desired signal.

- A critical consideration in audio-visual fusion approaches is the precise synchronization of audio and visual data. Aligning these two modalities correctly is paramount for the success of fusion-based methods. Inaccurate synchronization can lead to erroneous conclusions about the active speaker, which can undermine the reliability of the entire system. Achieving effective synchronization requires meticulous attention to detail, calibration, and the integration of both timing and visual cues. Failure in synchronization can result in fusion errors and degrade the overall accuracy of the detection process.
- The demand for real-time processing in active speaker detection, particularly for applications like video conferencing, adds another layer of complexity. Achieving real-time capabilities necessitates the development of algorithms that can swiftly analyze audio and visual data streams to identify the active speaker promptly. The challenge arises from striking a balance between accuracy and speed. While accurate detection is paramount, processing delays can lead to disjointed communication experiences. Thus, algorithms must be finely tuned to provide both real-time responsiveness and high accuracy, a delicate equilibrium to achieve.
- Another hurdle in active speaker detection emerges when multiple speakers overlap in their speech. The occurrence of simultaneous speech further complicates the task of isolating and identifying individual speakers. Distinguishing overlapping speakers requires advanced signal processing techniques capable of discerning different voice characteristics and patterns in a complex acoustic environment. Addressing speaker overlap is crucial for ensuring that all speakers are accurately identified, regardless of the intricacies of their interactions.
- **Speaker Identification Variability:** A significant challenge in active speaker detection arises from the natural variability in how individuals speak. People have distinct speech patterns, pitches, accents, and speaking rates. This variability complicates the task of creating generalized models that can accurately identify speakers across diverse populations.

Training models that can effectively handle this variability requires extensive and diverse training data. Moreover, the challenge extends to adapting these models to new speakers in real-time scenarios, where the model needs to quickly adjust to an unfamiliar voice and still ensure accurate detection.

- **Ambiguous Visual Cues:** While video-based methods rely on visual cues like lip movement and facial expressions, these cues are not always straightforward to interpret. Lip movement might be obscured due to various reasons, such as facial hair, lighting conditions, or camera angles. Additionally, facial expressions can be ambiguous, and a person's expression might not always correlate with whether they are actively speaking. Accurate interpretation of these cues requires sophisticated computer vision techniques capable of handling varying conditions and accurately associating visual cues with the active speaker.
- **Incorporating these challenges into active speaker detection solutions** necessitates the continuous advancement of signal processing techniques, machine learning algorithms, and computer vision methodologies. By addressing these complexities, the field can achieve more accurate and robust active speaker detection systems across diverse scenarios and applications. The challenges mentioned, such as noise and variability, speaker overlap, and audio-visual synchronization, highlight the intricacies of processing audio and visual data for active speaker detection. Innovations in filtering, feature extraction, and noise reduction are pivotal to enhancing the accuracy and reliability of active speaker detection algorithms.
- **Environmental Noise and Acoustic Conditions:** The presence of environmental noise and varying acoustic conditions poses a significant challenge to accurate active speaker detection. Background noise, reverberation, and changes in acoustic environments can distort the audio signal, making it difficult to differentiate between speakers and noise. Robust algorithms are required to filter out unwanted noise and enhance the clarity of speech signals. Additionally, adapting to different acoustic environments in real-time, such as a quiet room versus a noisy street, demands advanced noise reduction and adaptation techniques to ensure reliable detection.
- **Adverse Speaking Scenarios:** Active speaker detection can encounter scenarios where speakers don't follow typical conversational patterns. For example, whispering, speaker interruptions, or speakers deliberately lowering their voice to avoid detection can confound detection algorithms. Whispered speech might have significantly different acoustic characteristics compared to normal speech, and interruptions might lead to sudden changes in the audio signal. Handling such diverse speaking scenarios necessitates algorithms that can accommodate atypical speech patterns and abrupt changes in audio patterns.

- **Limited Training Data and Speaker Diversity:** Developing accurate speaker identification models requires substantial amounts of labeled training data. However, obtaining labeled data for every possible speaker across different demographics, accents, and languages is a daunting task. Limited training data can lead to models that struggle with speaker diversity, causing reduced accuracy when encountering speakers not well-represented in the training set. Ensuring that models are capable of handling a wide range of speakers, even those with limited training data, requires techniques like transfer learning and data augmentation.
- By acknowledging and addressing these challenges, researchers and engineers in the field of active speaker detection can contribute to the development of more sophisticated and robust algorithms. These algorithms should be capable of handling adverse speaking scenarios, diverse speaker populations, challenging acoustic conditions, and limited training data. As technology evolves, the field continues to strive towards achieving accurate and reliable active speaker detection systems across various real-world applications.

In conclusion, active speaker detection, despite its transformative potential, is not without challenges. Noise and variability in audio, accurate audio-visual synchronization, real-time processing demands, and the presence of speaker overlap all pose substantial hurdles. Overcoming these challenges requires the development of innovative algorithms that can effectively separate voices from noise, achieve precise synchronization, deliver real-time capabilities, and differentiate speakers even in challenging scenarios. By addressing these challenges, the field of active speaker detection can advance and provide more accurate and reliable solutions for various multimedia applications. Active speaker detection methods have far-reaching implications in multimedia analysis. They contribute to the creation of more efficient communication systems, enabling clearer interactions in contexts like video conferencing.

Accurate transcriptions benefit from correctly attributing text to the right speaker, resulting in more coherent and meaningful transcripts. Organized content indexing, achieved by identifying speakers at different timestamps, enhances content retrieval and navigation. Additionally, in security contexts, knowing who is speaking in surveillance footage provides valuable insights for monitoring and analysis. Active speaker detection methods, whether audio-based, video-based, or through audio-visual fusion, are essential tools in multimedia analysis. They leverage distinctive audio and visual cues to identify the active speaker, enhancing communication, transcription accuracy, content organization, and security in diverse multimedia scenarios. The choice of method depends on the context and available data, and collectively, these techniques contribute to shaping more effective and insightful multimedia systems.

8.4 Recommendations for Future Work

Future enhancements that can be done are as followed:

- **Subtitle Support:** Adding subtitle support to active speaker detection involves generating real-time subtitles that display the text of what the detected active speaker is saying. This enhancement can significantly improve user experience and accessibility. By providing subtitles, not only is the active speaker identified, but their speech is also transcribed in real-time, allowing participants to understand the content even if they have hearing impairments or if the audio quality is poor. Subtitles can enhance comprehension, especially in situations where accents or background noise might hinder understanding.
- **Real-Time Application in Video Conferencing:** Expanding the active speaker detection system to real-time applications like live video conferencing is a valuable enhancement. In live video scenarios, where conversations are happening in real time, identifying the active speaker as they talk ensures that participants can follow the ongoing conversation seamlessly. This enhancement improves the overall flow of communication, reducing interruptions and creating a more natural interaction environment.
- **Noise-Resistant Models:** Incorporating noise-resistant models is a critical advancement for accurate detection in environments with significant background noise or cluttered audio conditions. Noise reduction techniques and models trained to filter out disturbances can improve the system's ability to accurately identify the active speaker, even in challenging acoustic conditions. Such models can enhance the reliability of detection, contributing to a consistent user experience across different environments.
- **Privacy Measures:** As with any technology that involves audio and video data, privacy is a paramount concern. Future enhancements should include robust privacy measures to ensure that sensitive audio and visual data are handled securely. Implementing encryption, anonymization techniques, and adhering to data protection regulations can help build user trust and ensure that data privacy is maintained throughout the active speaker detection process.
- **User-Friendly Real-Time Interface:** Creating a user-friendly real-time interface is essential for easy visualization of active speaker detection results. A well-designed interface could include visual indicators, such as highlighting the current active speaker in a video grid or overlaying a speaker label. The interface should be intuitive and easy to navigate, allowing users to quickly grasp who is speaking at any given moment. Intuitive visualization enhances the user experience, making the technology more accessible and user-friendly.

- **Deep Learning Architectures (RNNs, CNNs):** Integrating deep learning architectures like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can lead to enhanced speaker detection accuracy. RNNs are adept at modeling sequential data, making them suitable for capturing temporal patterns in speech. CNNs excel at feature extraction from visual data, which can be advantageous in video-based speaker detection. Combining these architectures with large datasets can lead to more sophisticated and context-aware detection models.
- **Emotion and Sentiment Analysis:** Integrating emotion and sentiment analysis into active speaker detection can provide deeper insights into communication dynamics. By analyzing vocal tones, facial expressions, and speech patterns, the system can detect emotions such as happiness, anger, or sadness. This information can be valuable for understanding the emotional context of a conversation, which has applications in customer service, therapy sessions, and market research.
- **Multilingual Support:** Expanding active speaker detection to support multiple languages enhances its versatility and usability in global contexts. This entails training models on diverse language datasets and incorporating language-specific features. Multilingual support enables accurate speaker identification and transcription across different languages, facilitating cross-cultural communication and accessibility.
- **Adaptive Learning and Personalization:** Introducing adaptive learning mechanisms allows the system to adapt to individual speakers' characteristics over time. By learning each speaker's voice, speech patterns, and style, the system can enhance accuracy and adaptability. Personalized models for individual users or frequent participants in a conference can lead to improved identification performance.
- **Gesture Recognition and Body Language Analysis:** Incorporating gesture recognition and body language analysis adds another layer of context to active speaker detection. Visual cues like hand gestures, posture, and body movements can offer insights into the speaker's engagement level, confidence, and interaction dynamics.
- **Environmental Context Integration:** Consider integrating environmental context data, such as location and activity level, into the active speaker detection process. This can provide additional context for identifying the active speaker. For instance, in a noisy coffee shop, the system could prioritize detecting the speaker closer to the microphone, improving accuracy in challenging settings.

- **Real-Time Feedback and Coaching:** Enhancing the active speaker detection system to provide real-time feedback to speakers can help improve communication skills. The system could offer insights on speaking pace, clarity, and engagement levels. Such feedback is valuable for public speaking training, conference presentations, and interviews, contributing to effective communication development.
- **Integration with Virtual Assistants and AI:** Integrating active speaker detection with virtual assistants and AI systems can create more interactive and responsive environments. Virtual assistants can dynamically respond to the active speaker, addressing their queries, and tailoring responses based on the ongoing conversation. This integration blurs the lines between active speaker detection and conversational AI.
- **Long-Form Content Analysis:** Extending active speaker detection to long-form content, such as recorded lectures or podcasts, can offer enhanced content indexing and navigation. The system could automatically generate speaker-based content summaries or highlight key sections, aiding in efficient content consumption and review.
- **Social Interaction Analysis:** Applying active speaker detection to social interactions captured in videos or recordings can yield valuable insights into group dynamics and interpersonal communication. Analyzing the active speaker patterns within group discussions can reveal information about leadership roles, influence dynamics, and conversational turn-taking.
- These additional future enhancements broaden the horizons of active speaker detection. By incorporating emotion analysis, multilingual support, adaptive learning, gesture recognition, and more, the technology becomes more sophisticated, contextual, and beneficial across diverse scenarios and applications. These advancements align with the evolving nature of communication, technology, and user expectations, driving the field toward greater accuracy, versatility, and meaningful insights.

In conclusion, these future enhancements represent exciting directions for active speaker detection. By introducing subtitle support, real-time applications, noise-resistant models, privacy measures, user-friendly interfaces, and advanced deep learning architectures, the field can achieve even greater accuracy, usability, and versatility. These enhancements align with the evolving needs of communication systems, accessibility, and data privacy concerns, making active speaker detection more valuable and effective across various domains.

REFERENCES:

- [1] Tae Jin Parka, Naoyuki Kandab, Dimitrios Dimitriadis. (2021). “A review of Speaker Diarization with recent advances”.
- [2] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher (2019). “An Audio Visual Dataset for Active Speaker Detection”.
- [3] Juan Leon Alc ´ azar, Fabian Caba Heilbron , Ali K. Thabet1 & Bernard Ghanem (2021). “Multi-modal Assignment for Active Speaker Detection”.
- [4] Junhua Liao, Haihan Duan2, Kanghui Feng1, Wanbing Zhao, Yanbing Yang, Liangyin Chen (2020). “A Light Weight Model for Active Speaker Detection”.
- [5] Yogesh Virkar, Brian Thompson, Rohit Paturi, Sundararajan Srinivasan, Marcello Federico (2020). “Speaker Diarization of Audio-Visual Content”.
- [6] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, Stefanos Zafeiriou (2018). “Retina Face for face detection”.
- [7] Baptiste Pouthier, Laurent Pilati, Leela K. Gudupudi, Charles Bouveyron, Frederic (2019). “Active Speaker Detection as a Multi-Objective Optimization.