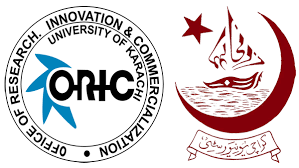
Voice Processing System

(VPS)



**Session 2017-2021**

**Project Supervisor**

**Sir Muhammad Saeed**

**Submitted By**

**ALI ZIA UDDIN (B16101021)**

**KAZIM HUSSAIN (B16101061)**

**AZMEER ABRAR KHAN (B16101029)**

**MUHAMMAD AMMAR AAMIR (B16101081)**

**A report submitted to the**

**Department of Computer Science**

**In partial fulfillment of the requirement for the**

**Degree**

**Bachelor of Science in**

**Computer Science**

**By**

**Ali Zia Uddin**

**Kazim Hussain**

**Azmeer Abrar Khan**

**Muhammad Ammar Aamir**

**UNIVERSITY OF KARACHI**

**DATE 7th April 2021**

We truly acknowledge the cooperation and help by Sir Muhammad Saeed, Professor , Department of Computer Science University of Karachi. He has been a constant source of guidance throughout the course of this project.

(Signed)

**Contents**

1. **ABSTRACT** 7
2. **CHAPTER 01  
   INTRODUCTION**

1.1 PROBLEM STATEMENT 9

* 1. GOALS 9
  2. CHALLENGES 9

1. **CHAPTER 02  
   SPEAKER DIARIZATION**

[2.1](#_heading=h.3dy6vkm) INTRODUCTION 12

* 1. ANATOMY OF A SPEAKER DIARIZATION SYSTEM 12
     1. SPEECH DETECTION
     2. SPEECH SEGMENTATION
     3. AUDIO EMBEDDING EXTRACTION
     4. CLUSTERING
     5. RESEGMENTATION
  2. IMPLEMENTATION 14
     1. mp3 TO wav
     2. SPEECH DETECTION
     3. SPEECH SEGMENTATION AND AUDIO EMBEDDING EXTRACTION
     4. SPECTRAL CLUSTERING
     5. CONTINUOUS SEGMENTS

1. **CHAPTER 03  
   SPEAKER RECOGNITION**
   1. INTRODUCTION 19
   2. DATA COLLECTION & PREPARATION 21  
        
      3.2.1 AUDIO RECORDING

3.2.2 AUDIO PREPROCESSING

* 1. FEATURE EXTRACTION 22
  2. MODEL TRAINING 23
  3. MODEL TESTING 24

1. **CHAPTER 04  
   INTEGRATION**

4.1 TECHNOLOGIES 26

4.1.1 PYTHON

4.1.2 FLASK

4.1.3 FLUTTER & DART

4.1 IMAGES 26

**ABSTRACT**

**Speaker Recognition And Diarization**

Speaker recognition is a task of identifying persons from their voices. Recently, deep learning has dramatically revolutionized speaker recognition. However, there is a lack of comprehensive reviews on the exciting progress. In this document, we review two major subtasks of speaker recognition, including identification and diarization, with a focus on deep-learning-based methods. Because the major advantage of deep learning over conventional methods is its representation ability, which is able to produce highly abstract embedding features from utterances, we first pay close attention to deep-learning-based speaker feature extraction, including the inputs, network structures,and objective functions respectively, which are the fundamental components of many speaker recognition subtasks. Then, we make an overview of speaker diarization, with an emphasis of recent supervised, end-to-end diarization.

**CHAPTER 01**

**INTRODUCTION**

# PROBLEM STATEMENT

In the age of IoT and automation there is a growing need to have work done as effortlessly as possible. One such area which has a great role to play in automation is Voice Processing. Through voice processing commands can be given to machines to carry out different tasks. Another Area for Voice Processing is speech-to-text for example writing transcripts in a conference of different speakers and being able to distinguish between the speakers.

* 1. **GOALS**

To develop a model that uses assisted recognition techniques to discern between different speakers. For this purpose supervised learning should be combined with speaker diarization so as to help in distinguishing speakers of similar features.

* 1. **CHALLENGES**
* Collection of clean audio data
* Preprocessing to reduce ambient noise
* Choice of model for optimal results
* Integration

# CHAPTER 02: SPEAKER DIARIZATION

**2.1 INTRODUCTION**

Speaker diarization is the process of partitioning an input audio

stream into homogeneous segments according to the speaker identity. It answers the question “who spoke when” in a multi-speaker environment. It has a wide variety of applications including multimedia information retrieval, speaker turn analysis, and audio processing. In particular, the speaker boundaries produced by diarization systems have the potential to significantly improve acoustic speech recognition (ASR) accuracy.

**2.2 ANATOMY OF A SPEAKER DIARIZATION SYSTEM**

A typical speaker diarization system usually consists of the following subsystems:

**2.2.1 SPEECH DETECTION**

where the speech is filtered out from non-speech. This is required to trim out silences and non-speech parts from audio recording.

**2.2.2 SPEECH SEGMENTATION**

where the input audio is segmented into short sections that are assumed to have a single speaker, and the non-speech sections are filtered out.

**2.2.3 AUDIO EMBEDDING EXTRACTION**

where specific features such as MFCCs, speaker factors, or i-vectors are extracted from the segmented sections.

**2.2.4 CLUSTERING**

where the number of speakers is determined, and the extracted audio embeddings are clustered into these speakers.

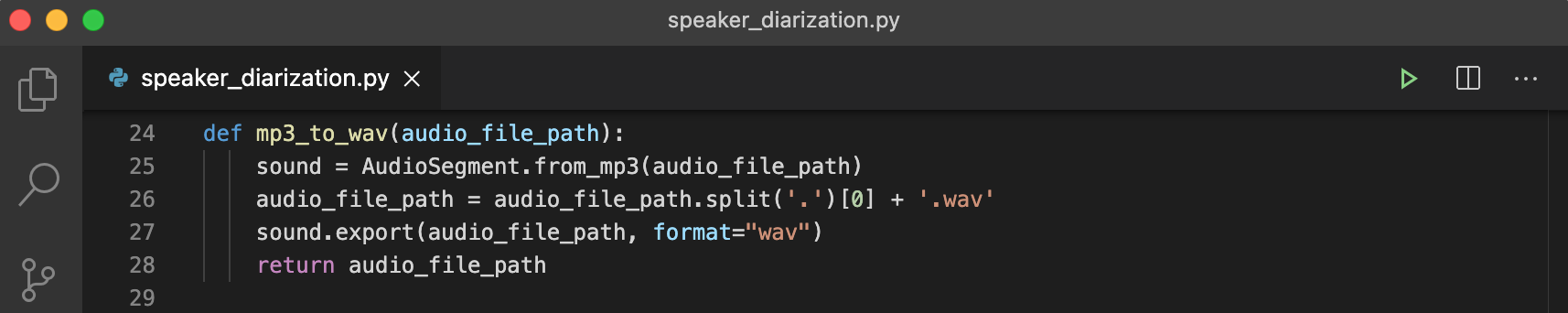
**2.2.5 RESEGMENTATION (OPTIONAL)**

where the clustering results are further refined to produce the final diarization results.

**2.3 IMPLEMENTATION**

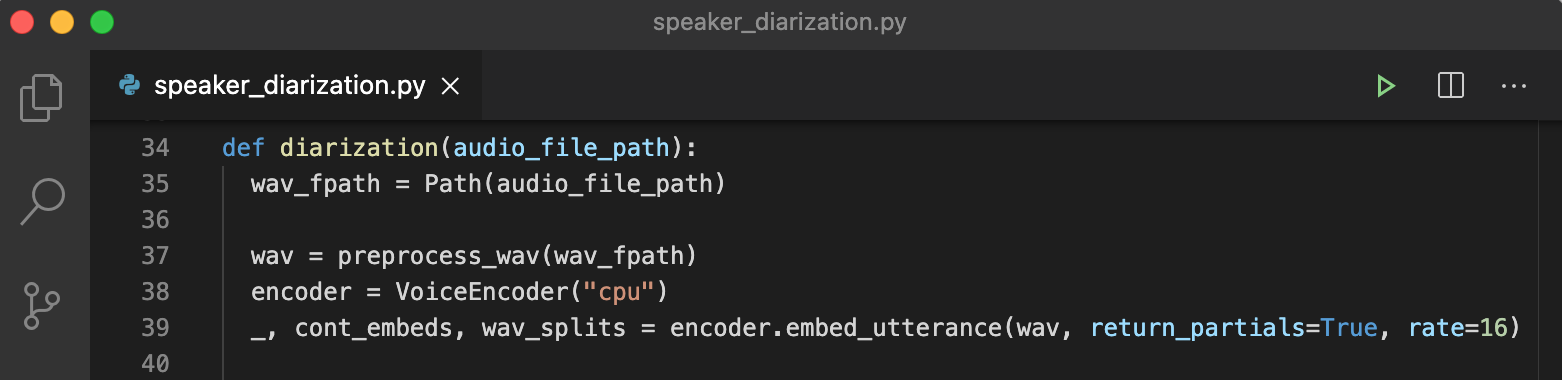
We implemented the Speaker Diarization module by using the pre-trained model provided by the Resemblyzer repository for creating speaker embeddings.

**2.3.1 mp3 TO wav**



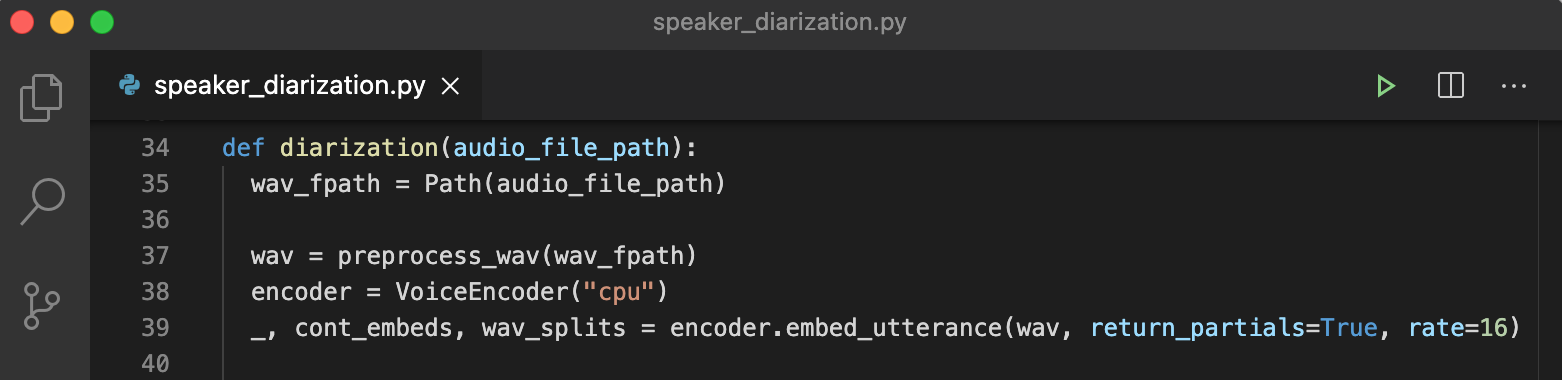
**2.3.2 SPEECH DETECTION**

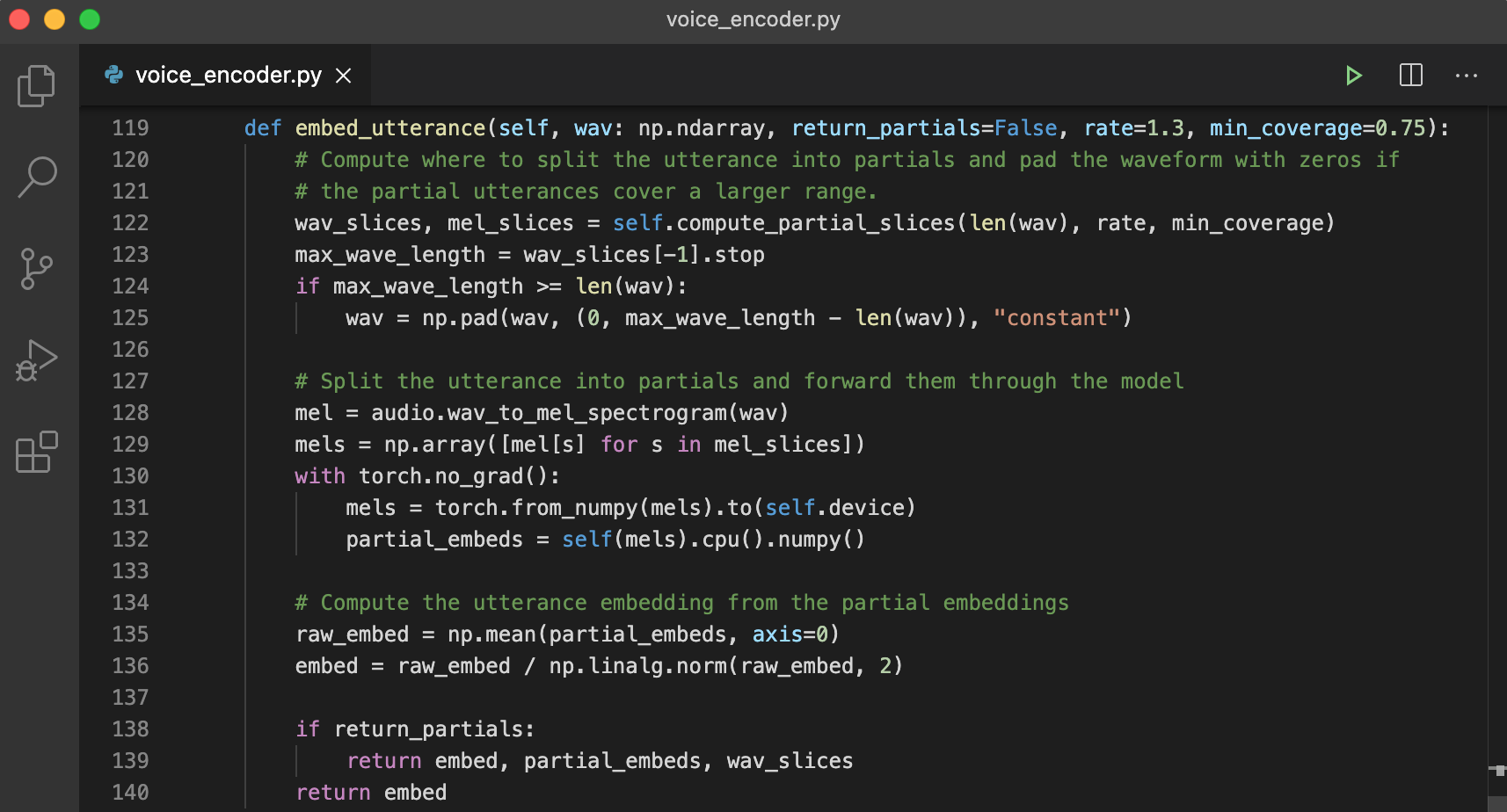
In this step, we use a Voice Activity Detector (VAD) module to separate out speech from non-speech.



We pass the file path to the preprocess\_wav function, which internally uses a VAD to trim out the silences in the audio file and also normalizes the decibel level of audio.

**2.3.3 SPEECH SEGMENTATION AND AUDIO EMBEDDING EXTRACTION**



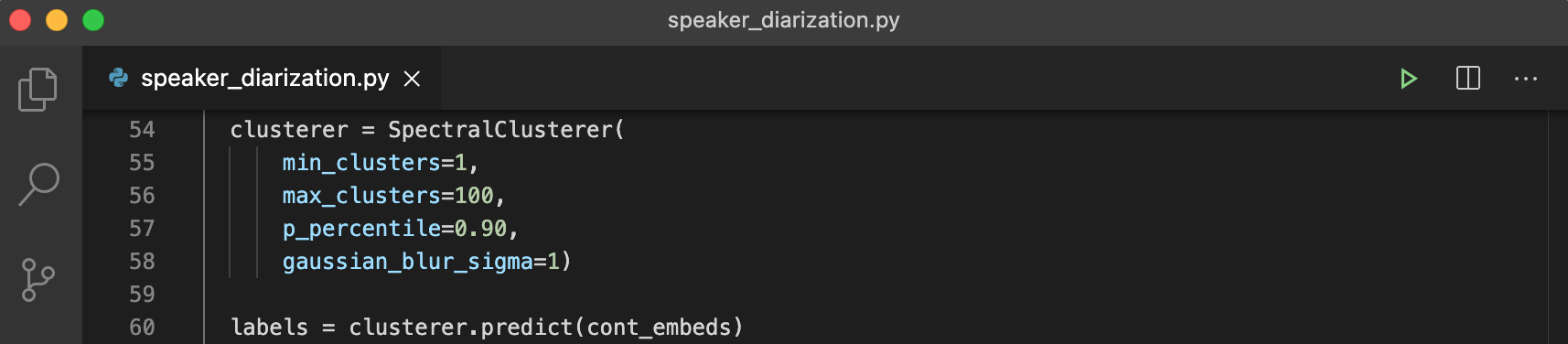


We create an instance of the VoiceEncoder class name encoder, passing cpu as default device. The embed\_utterance function of this instance takes in the processed wav file, segments it out into windows , makes MFCCs of these segments and finally creates d-vectors of these audio segments.

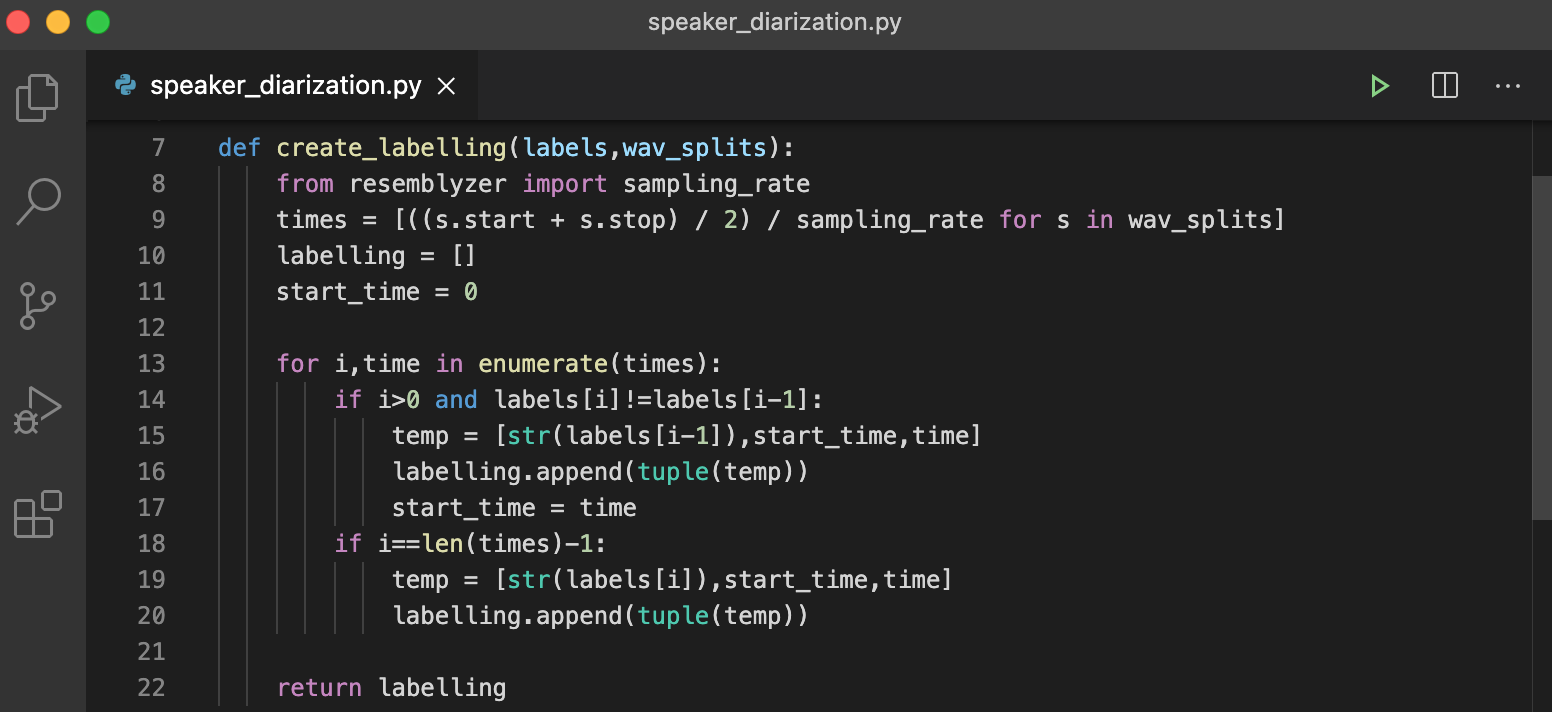
The cont\_embeds is a N by D matrix, where N is the number of segments created (which is equal to the number of d-vectors) and D is the dimension of each d-vector, which by default is 256. wav\_splits is a list with the start and end time of each window for which a d-vector has been created.

**2.3.4 SPECTRAL CLUSTERING**

Spectral clustering works by first creating an affinity matrix of the data ( a matrix whose (i,j)th element is obtained by performing some mathematical operation on ith and jth embedding), and then after performing some matrix operations on this affinity matrix, we use K-means to get the final labels.

****

**2.3.5 CONTINUOUS SEGMENTS**

We have got labels for our d-vectors in the previous step. However, each d-vector corresponds to a small window frame, and using them directly for any further task is not feasible. We need to join continuous windows which have a common speaker together.

We pass the previously created labels and wav\_splits to the create\_labelling function, which finds a continuous segment of windows having a common label (or a common speaker) and merges them together. labelling is a list of tuples with values in order (speaker\_label, start\_time, end\_time)

# CHAPTER 03: SPEAKER RECOGNITION

**3.1 INTRODUCTION**

It is known that a speaker’s voice contains personal traits of the speaker, given the unique pronunciation organs and speaking manner of the speaker, e.g. the unique vocal tract shape, larynx size, accent, and rhythm . Therefore, it is possible to identify a speaker from his/her voice automatically via a computer. This technology is termed as automatic speaker recognition. Speaker recognition is a fundamental task of speech processing, and finds its wide applications in real-world scenarios. For example, it is used for the voice-based authentication of personal smart devices, such as cellular phones, vehicles, and laptops. It guarantees the transaction security of bank trading and remote payment. It has been widely applied to forensics for investigating a suspect to be guilty or not-guilty , or surveillance and automatic identity tagging. It is important in audio-based information retrieval for broadcast news, meeting recordings and telephone calls. It can also serve as a frontend of automatic speech recognition (ASR) for improving the transcription performance of multi-speaker conversations.

The research on speaker recognition can be dated back to at least the 1960s. In the following forty years, many advanced technologies promoted the development of speaker recognition. For example, a number of acoustic features ( dictive cepstral coefficients, the perceptual linear prediction coefficient, and the mel-frequency cepstral coefficients) and template models (e.g. vector quantization, and dynamic time warping) have been applied. Later on, D. A. Reynolds, T. F. Quatieri, R. B. Dunn proposed the Gaussian mixture model based universal background model (GMM-UBM), which has been the foundation of speaker recognition for more than ten years since year 2000. Several representative models based on GMM-UBM have been developed, including the applications of support vector machines and joint factor analysis. Among the models, the GMMUBM/i-vector frontend with probabilistic linear discriminant analysis (PLDA) backend [10, 11] provided the state-of the-art performance for several years, until the new era of deep learning based speaker recognition. Recently, motivated by the powerful feature extraction capability of deep neural networks (DNNs), a lot of deep learning based speaker recognition methods were proposed right after the great success of deep learning based speech recognition, which significantly boosts the performance of speaker recognition to a new level, even in wild environments.

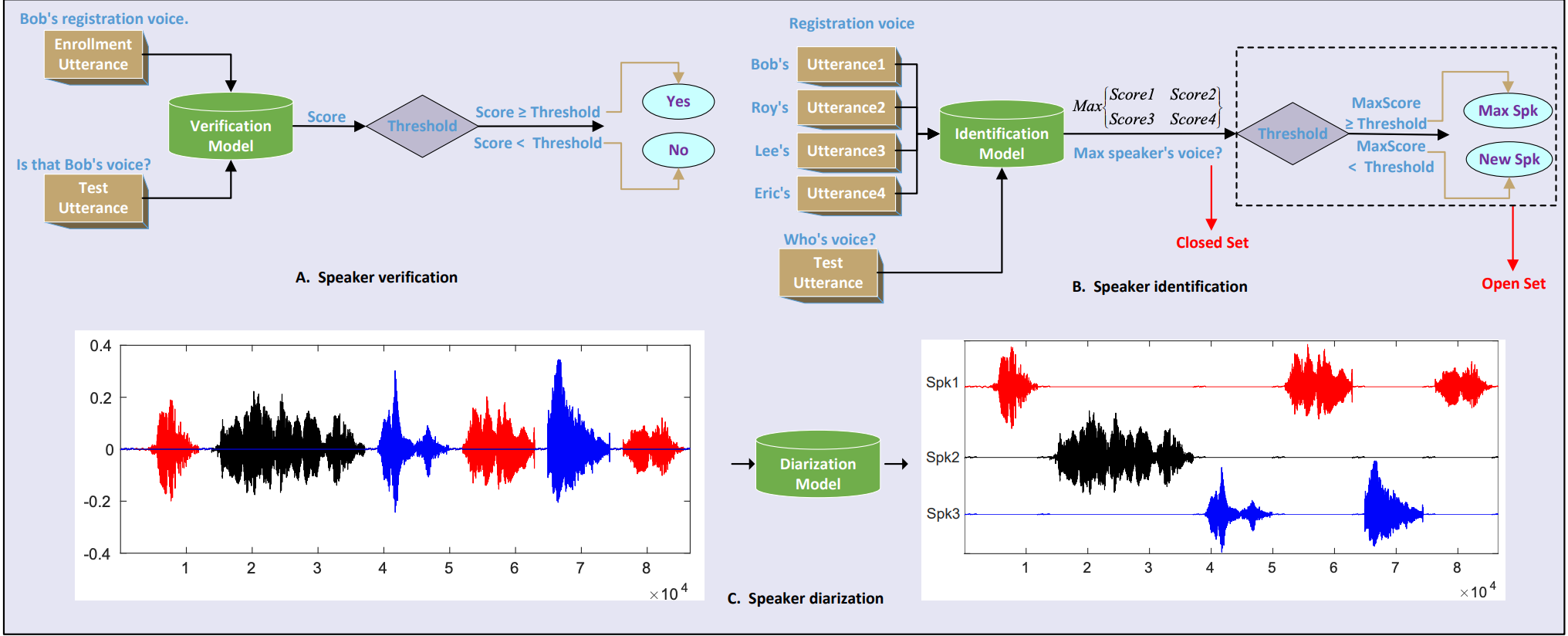


Figure 1: Flowcharts of speaker verification, speaker identification, and speaker diarization. Fig. A describes speaker verification, which is a task of verifying whether a test utterance and an enrollment utterance are uttered by the same speaker via comparing the similarity score of the utterances with a predefined threshold. Fig. B describes speaker identification, which is a task of determining the speaker identity of a test utterance from a set of speakers. If the utterance must be produced from the set of the speakers, then it is a closed set identification problem; otherwise, it is an open set problem. Fig. C describes speaker diarization, which addresses the problem of “who spoke when”, i.e., partitioning a conversation recording into several speech recordings, each of which belongs to a single speaker.

**3.2 DATA COLLECTION & PREPARATION**

Data preparation is the process of gathering, combining, structuring, and organizing data so it can be used in supervised learning. We need to prepare a custom data set for training purposes according to our requirements.

**3.2.1 AUDIO RECORDING**

Audio is recorded in wav format, minimally a 60 seconds clip in which the speaker can say anything and is not bound to a specific script. This audio clip is then broken down into 10 smaller clips of equal length.

**3.2.2 AUDIO PREPROCESSING**

Prior to training, the audio data is preprocessed by variance scaling, to help with largely sparse datasets. The issue with sparsity is that it is very biased or in statistical terms skewed. So, therefore, scaling the data brings all your values onto one scale eliminating the sparsity. In mathematical terms, this follows the same concept of Normalization and Standardization.

**3.3 FEATURE EXTRACTION**

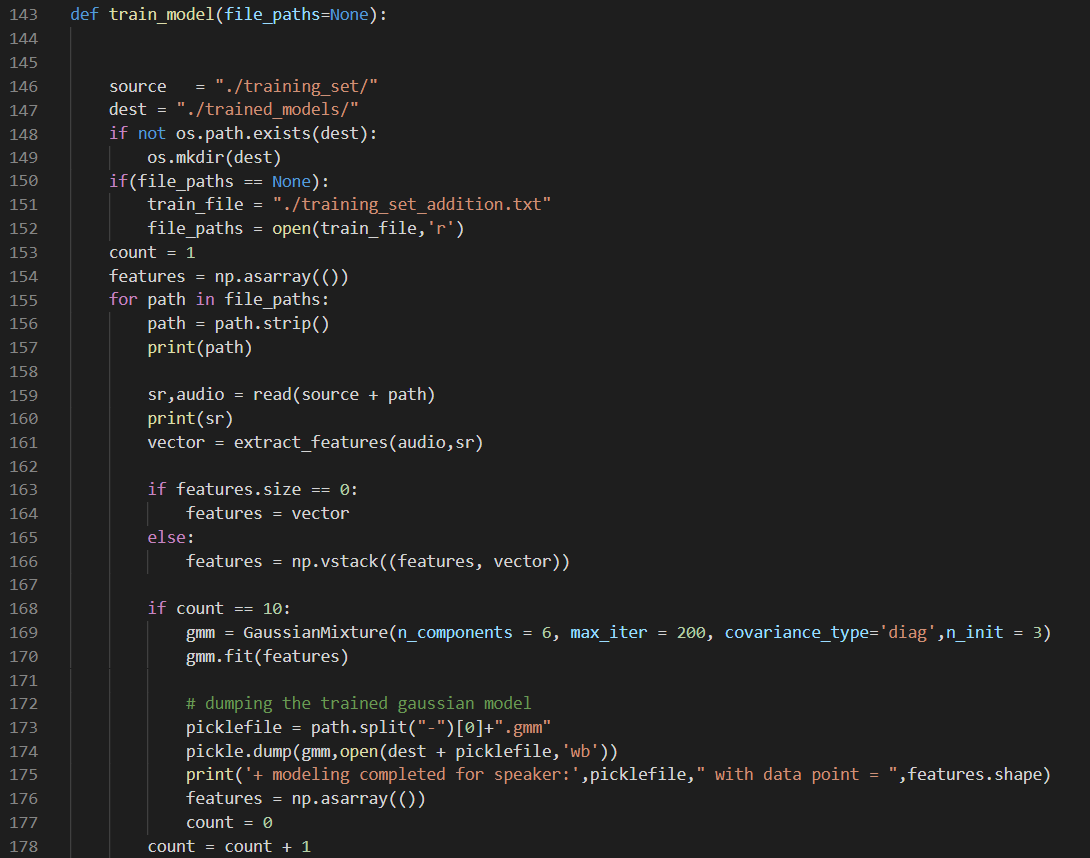
The idea is to make the correct identification of the speaker by using the Gaussian mixture model. The first step while dealing with an audio sample is to extract the features from it i.e. to identify components from the audio signal. We are using the Mel frequency cepstral coefficient (MFCC) to extract the features from the audio sample. MFCC which maps the signal onto a non-linear Mel-Scale that mimics the human hearing and provides the MFCC feature vectors which individually describes the power spectral envelope of a single frame.



We considered the MFCC with tuned parameter as a primary feature and delta MFCC which also known as differential and acceleration coefficients which are used to deal with speech information which is related to dynamics i.e. trajectories of MFCC coefficient over time it turns out to calculation of these trajectories.

**3.4 MODEL TRAINING**

Gaussian Mixture Model (GMM) is one of the most popular models used for training while dealing with audio data so we used MFCC and GMM together to achieve the target of identifying the speaker correctly. GMM is used to train the model on (MFCC+Delta MFCC) extracted features.



**3.5 MODEL TESTING**

Now, the trained model can be tested by supplying it with an audio file. Features are extracted from the audio file and then fed to the model, which tests it on the trained classes and returns the name of the speaker.

# CHAPTER 04: INTEGRATION

**4.1 TECHNOLOGIES**  
   
The technologies used in the project are:

**4.1.1 PYTHON**

Python is the programming Language used by the speaker diarization and speaker identification applications.

**4.1.2 FLASK**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. In this project Flask is used to create a REST api for communication between the user application and the speaker diarization and speaker identification applications.

**4.1.3 FLUTTER & DART**

Flutter is a Dart Framework used to write frontend applications for android, iOS and web. The user application of this project is written in dart on the Flutter Framework.

# CHAPTER 05: IMAGES

# 