# Voice Conversion using classical machine learning techniques

**Group No. 31**

Vishal Bharti (BTech/ECE)
Shashank G. Sharma (MTech (CSE)/Gen)
Aditi Roy (MTech (CSE)/Gen)
VijayKrishna Pamula(BTech/ECE)
Ritwik Kashyap(BTech/CSB)

## Contents

Preprint. Under review.

# Abstract

Voice conversion is a significant field in speech processing and authentication, offering operations such as speaker imitation and speaker verification.It plays vital role in many applications like speaker adaptation,voice cloning, multilingual text-to-speech, voice enhancement. In this report, we will provide an analysis of the metadata associated with the "Common Voice Corpus 14.0" dataset. This dataset contains a total of large number of MP3 audio files. We focused on the metadata files and their contents, preprocessing steps, and the key columns present in the metadata.Understanding the structure of the dataset, checking for missing values, and obtaining relevant insights from the data were all part of our analysis.The findings help to improve speech conversion and speaker verification approaches utilising conventional machine learning, making them a significant resource for applications in security, voice modulation, and human-computer interface.

## 1 Introduction

Voice conversion, an essential component of speech processing and authentication, has received considerable interest due to its many applications, which include speaker impersonation, speaker verification, and numerous human-computer interaction situations. The extensive and diversified Link English Corpus V14.0 dataset serves as the cornerstone for our research, offering a solid platform for voice-related investigations. We intend to facilitate voice conversion by employing conventional machine learning techniques such as Gaussian Mixture techniques and Hidden Markov Models, with the goal of transforming one speaker's speech into the manner of another while keeping its naturalness and quality. We will use traditional machine learning approaches in speech conversion and speaker verification, providing useful insights into the field of voice processing. This project has practical significance in a variety of sectors, notably in strengthening security measures, voice modulation, and improving human-computer interface systems. The research will also demonstrate the importance and potential of traditional machine learning approaches in solving the issues provided by voice conversion and speaker verification, as well as the crucial function of spectrogram comparison as an objective performance indicator [10].

## 2 Dataset (English Corpus V14.0)

The metadata for the "Common Voice Corpus 14.0" dataset is distributed across eight files, comprising seven TSV (Tab-Separated Values) files and one TXT file. These files include: 'train.tsv,' 'dev.tsv,' 'test.tsv,' 'validated.tsv,' 'invalidated.tsv,' 'reported.tsv,' 'other.tsv,' and 'times.txt.' Notably, the 'validated.tsv' file stands out with an extensive collection of rows, totaling 1,724,056 entries. This file is a valuable resource, housing a repository of validated audio files that can be employed for data augmentation in future applications. On the other hand, the 'invalidated.tsv' and 'reported.tsv' files are excluded from the analytical process as they contain audio files of low quality or those that have been flagged for various reasons. The primary TSV files for machine learning tasks are 'train.tsv,' 'test.tsv,' and 'dev.tsv,' which serve as the central components for data analysis and model development.

### 2.1 Columns in dataset

In the "Common Voice Corpus 14.0," consistency is maintained across all TSV files with a shared set of 11 columns that collectively describe the speaker and speech-related details, as well as essential information regarding the audio file and the speaker's client ID. These uniform columns include: 'client_id,' which serves as the identifier for the speaker or client; 'path,' specifying the file path to the corresponding audio; 'sentence,' containing the transcribed sentence found within the audio; 'up_votes,' representing the count of upvotes received for the particular audio; 'down_votes,' accounting for the number of downvotes garnered; 'age,' denoting the age of the speaker; 'gender,' indicating the gender of the speaker; 'accents,' providing insights into the speaker's accent; 'variants,' typically containing variant information (often recorded as NaN); 'locale,' which consistently registers as "en" for English; and finally, 'segment,' an additional column typically marked as NaN, used to convey segment-related information. This standardized column structure facilitates efficient data management and analysis throughout the dataset.

### 2.2 Preprocessing On Metadata Dataframe

In our data preparation process, we performed a series of essential preprocessing steps on the 'train.tsv,' 'dev.tsv,' and 'test.tsv' files. These steps were designed to enhance the quality and relevance of the dataset for subsequent analysis. Firstly, we carefully examined and filtered out rows with missing values in the columns representing age, gender, or accents. This action was crucial as these attributes play a pivotal role in speaker recognition, and the absence of such information could hinder the accuracy of our models.

Another important aspect of our preprocessing involved the evaluation of the quality of audio files. We specifically retained audio files with more up_votes than down_votes, as this served as a reliable indicator of good audio quality. By doing so, we ensured that our dataset primarily consisted of high-quality audio recordings.

Lastly, we streamlined the data by removing the 'variant' and 'segment' columns from the dataframe. These columns predominantly contained NaN values and contributed no meaningful information for our analysis. By eliminating them, we streamlined the dataset, making it more efficient for subsequent machine learning processes. Overall, these preprocessing steps were instrumental in refining the dataset and ensuring its suitability for our speaker recognition and analysis endeavors.

| | train.tsv | dev.tsv | test.tsv |
|---|---|---|---|
| Total Rows | 1,046,353 | 16,379 | 16,380 |
| Total Columns | 11 | 11 | 11 |

Table 1: Before removal of NaN values

| NaN value in Each Column | train.tsv | dev.tsv | test.tsv |
|---|---|---|---|
| client_id | 0 | 0 | 0 |
| path | 0 | 0 | 0 |
| sentence | 0 | 0 | 0 |
| up_votes | 0 | 0 | 0 |
| down_votes | 0 | 0 | 0 |
| age | 303,874 | 13,774 | 14,055 |
| gender | 303,627 | 13,783 | 14,063 |
| accents | 407,135 | 14,585 | 14,600 |
| variant | 1,046,353 | 16,379 | 16,380 |
| locale | 0 | 0 | 0 |
| segment | 1,046,353 | 16379 | 16,366 |

Table 2: Analysis with NaN value

| | train.tsv | dev.tsv | test.tsv |
|---|---|---|---|
| Total Rows | 578,446 | 1333 | 1132 |
| Total Column | 9 | 9 | 9 |

Table 3: After removal of NaN values

- training dataframe

  This contains information on the audio data used to train a machine learning model. This dataset is essential for developing and tailoring speech recognition or voice-related models in the context of the English language.

  These gender, accent, and age graphs are useful for analysing the demographic and linguistic variety of the speakers in the English Corpus V14.0 training dataframe. By analysing these graphs, we can gain a better understanding of the dataset's properties and modify their machine learning models or analyses appropriately.

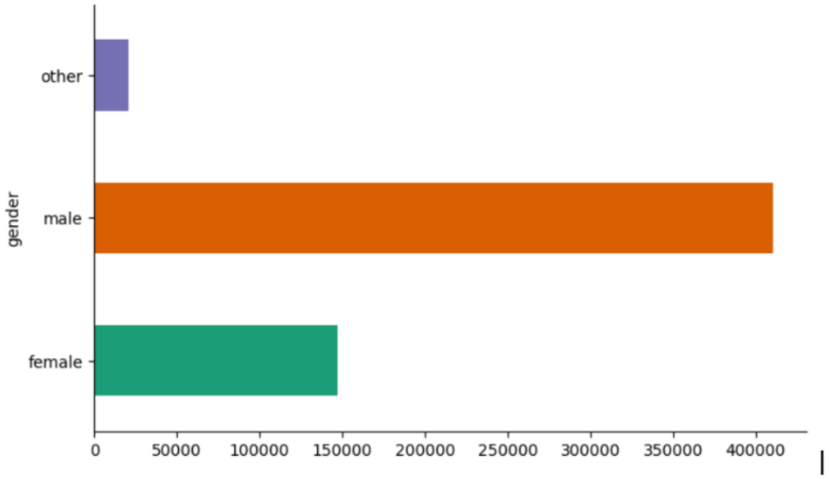


Figure 1: Distribution of ‘gender in training dataframe

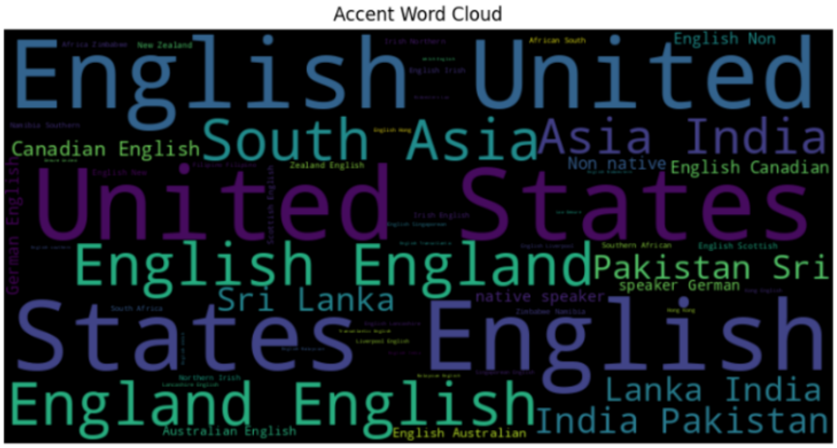


Figure 2: Distribution of ‘accents in training dataframe

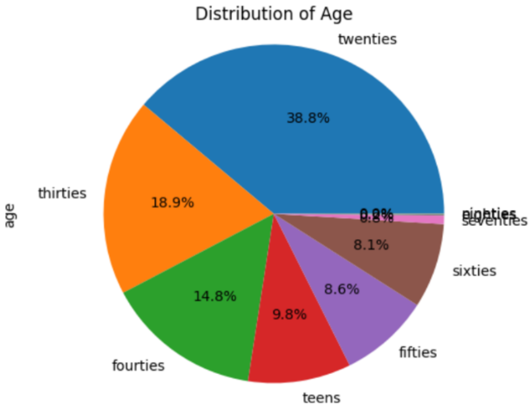


Figure 3: Distribution of ‘age’ in training dataframe

- dev dataframe
  The graphs of gender, accents, and age in the dev dataset provide important insights into the

demographic and linguistic variety of speakers contributing to the dev data in the English Corpus V14.0.

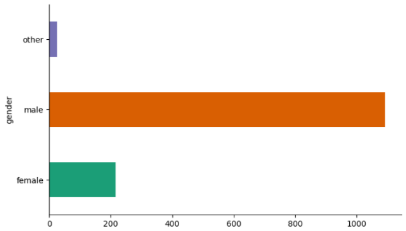


Figure 4: Distribution of ‘gender dev dataframe

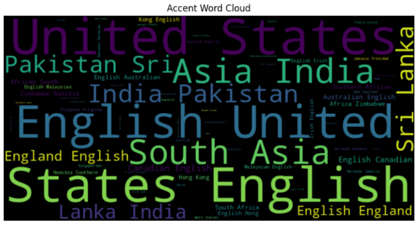


Figure 5: Most common ‘accents’ dev dataframe

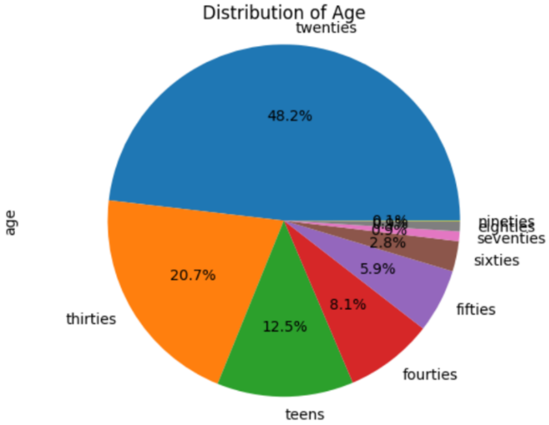


Figure 6: Distribution of ‘age’ dev dataframe

- times dataframe
  The "times.txt" file contains information on the length of audio snippets. The overall dataset’s mean length is $5184.14ms$, with a variation of $98,437,548.83ms^2$.

- test dataframe
  The graphs of gender, accents, and age in the English Corpus V14.0 test dataframe allow academics and data analysts to acquire a full knowledge of the demographic features contained in the test dataset. They help evaluate the diversity and features of the speakers in the test data for various language processing and voice recognition tasks.

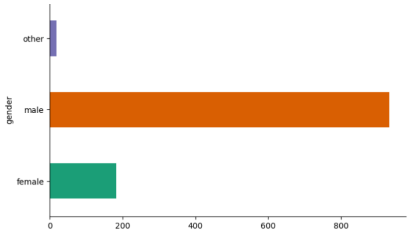


Figure 7: Distribution of ‘gender’ in test dataframe

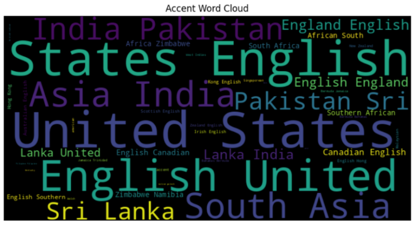


Figure 8: Most common ‘accents’ in test dataframe

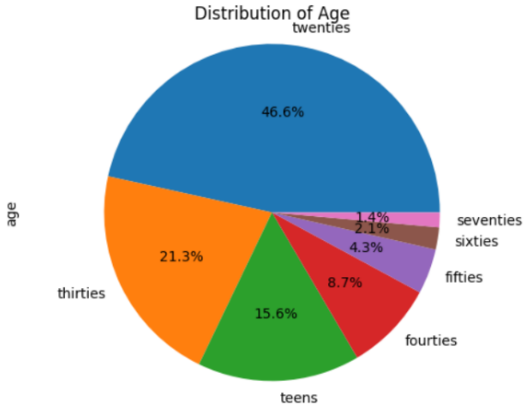


Figure 9: Distribution of ‘age’ in test dataframe

These graphs give critical insights into the demographic makeup of the dataset, allowing researchers and developers to adjust their speech-related models and technologies to specific speaker traits or demographic characteristics.

## 3 Visualization

Visualizing audio waveforms is essential for understanding audio characteristics. Key insights include amplitude (louder vs. quieter), duration (time axis), transients (sharp changes), periodicity (repeating patterns), silence (flat lines for pauses), and noise/distortion (irregular patterns). Analyzing these aspects helps reveal crucial audio traits and potential areas for improvement.

The total number of files in our dataset is 1503676. It is very hard to analyse such a large number of files. So, as a solution to this, first in order to analyse the waveform and spectogram of the audio files in our dataset, we have loaded 20 files from the dataset for the analysis. The files that we have considered for visualization are:

| Audio files | Audio files |
|---|---|
| common_voice_en_100155.mp3 | common_voice_en_9334.mp3 |
| common_voice_en_2125.mp3 | common_voice_en_2528.mp3 |
| common_voice_en_8840.mp3 | common_voice_en_7.mp3 |
| common_voice_en_3478.mp3 | common_voice_en_8034.mp3 |
| common_voice_en_1592.mp3 | common_voice_en_2918.mp3 |
| common_voice_en_8450.mp3 | common_voice_en_10127.mp3 |
| common_voice_en_9139.mp3 | common_voice_en_3829.mp3 |
| common_voice_en_1839.mp3 | common_voice_en_1426.mp3 |
| common_voice_en_10699.mp3 | common_voice_en_100155.mp3 |
| common_voice_en_3139.mp3 | common_voice_en_9750.mp3 |

Table 4: Audio files taken for visualization

Among the 20 audio files,we selected one file "common_voice_en_100155.mp3" and performed visualization analysis, after confirming the code and results we proceeded with other files. We get a waveform [Fig.10] and spectrogram [Fig.11] of this file as given in the figure. Spectrograms are essential for visualizing the frequency content of audio signals over time. They represent frequencies on the vertical axis (Hz) and time on the horizontal axis (s) using color intensity to depict amplitude. Bright areas indicate higher energy, while dark areas represent lower energy. Patterns, including continuous horizontal lines and vertical transients, reveal key audio characteristics. Harmonic content appears as evenly spaced lines above the fundamental frequency. Analyzing changes over time offers insights into the audio’s dynamics, making spectrograms indispensable for understanding complex audio features [3].

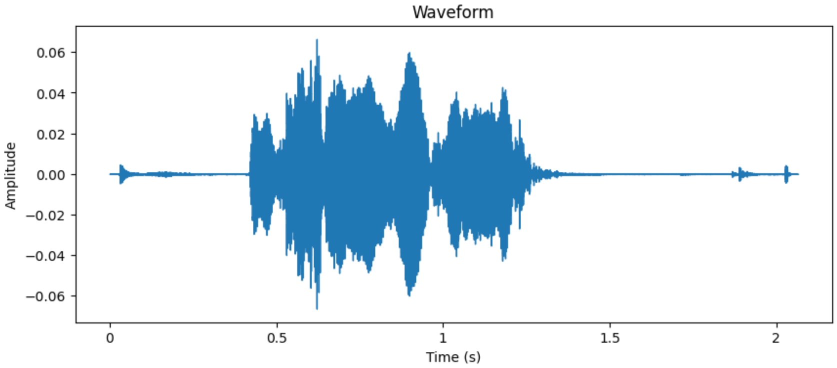


Figure 10: Audio Waveform of common_voice_en_100155.mp3

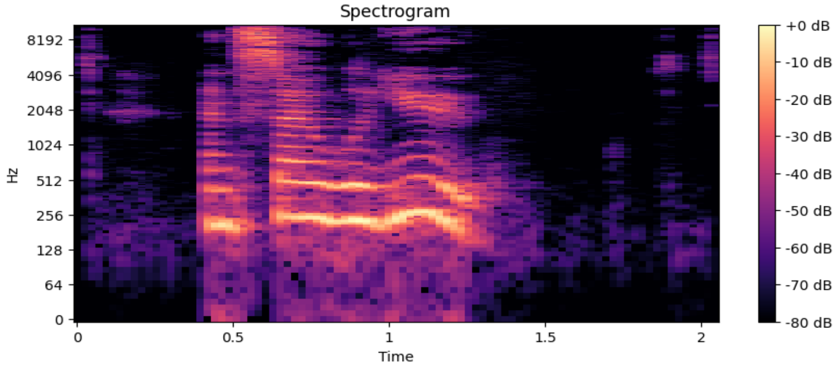


Figure 11: Spectrogram of common_voice_en_100155.mp3

We have plotted the frequency spectrum of "common_voice_en_100155.mp3" is given in the figure[Fig.12].

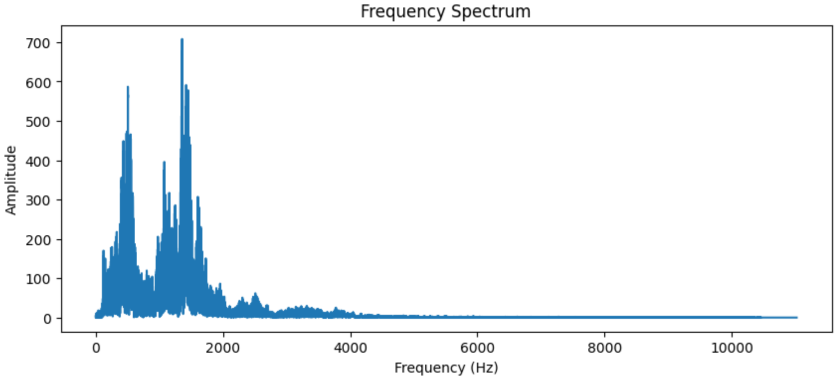


Figure 12: frequency spectrogram

Histograms of audio signals are valuable tools for discerning amplitude distribution, shedding light on critical audio characteristics.In the histogram , the x-axis represents amplitude levels, while the y-axis shows the frequency of occurrence for each amplitude level. A broader amplitude spread signifies a more extensive dynamic range, reflecting loudness variations.Peaks in the histogram reveal the audio's overall loudness by highlighting common amplitude values. Louder audio features more occurrences of higher amplitudes.Sharp peaks at the histogram's edges may suggest audio clipping, which distorts quality.A wider amplitude range, especially in the lower end, might indicate background noise.The histogram's portrayal of the range between the lowest and highest amplitudes offers insight into the audio's dynamic range.High frequency at the amplitude extremes could indicate audio saturation or heavy compression, impacting fidelity and dynamic expressiveness. In essence, audio signal histograms offer vital insights into amplitude distribution and essential audio characteristics.We are plotting the histograms of all the 20 files based on their amplitude distribution and saving them in a seperate directory. So, one of these "common_voice_en_100155.mp3" has its histogram as given in [Fig.13][8]. The average of all histograms is shown in the figure [Fig.14].

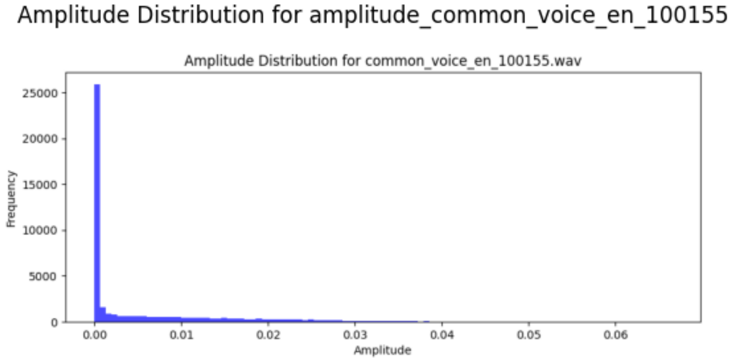


Figure 13: Histogram for 'common_voice_en_100155.mp3'

The Fast Fourier Transform (FFT) stands as a potent tool for delving into the frequency content of audio signals. Analyzing FFT plots reveals essential insights into the audio's frequency characteristics, with key elements to consider.Within the FFT plot, peaks emerge, serving as indicators of dominant frequencies in the audio signal, representing primary tones and harmonics. The amplitude of these peaks directly reflects the strength of the components at their respective frequencies.The x-axis on the plot signifies frequency in Hertz (Hz), offering a glimpse into the audible spectrum present in the audio. Unwanted noise or artifacts may manifest as smaller peaks, distinguishable by their frequency and amplitude.Should the audio contain harmonics, they will manifest as additional peaks at regular intervals. By examining the FFT plot, one can discern whether the audio's frequency distribution is narrow (tonal) or broad (noisy) [Fig.12].FFT plays a pivotal role in scrutinizing the frequency components of audio signals, facilitating the identification of dominant frequencies, their amplitudes, spectral characteristics, and the presence of noise or harmonics [6].

The frame energy histogram, derived from the distribution of energy levels in segmented frames of an audio signal, finds versatile applications. It distinguishes between voiced and unvoiced segments in speech recognition, aids in event detection through abrupt energy changes, and contributes to audio quality assessment by identifying anomalies. The histogram's role in segmentation assists in tasks like music analysis, while its insights into energy characteristics are crucial for efficient audio

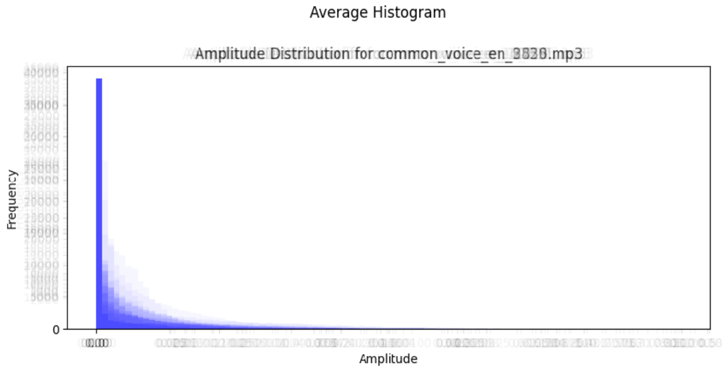


Figure 14: Average of all the histogram

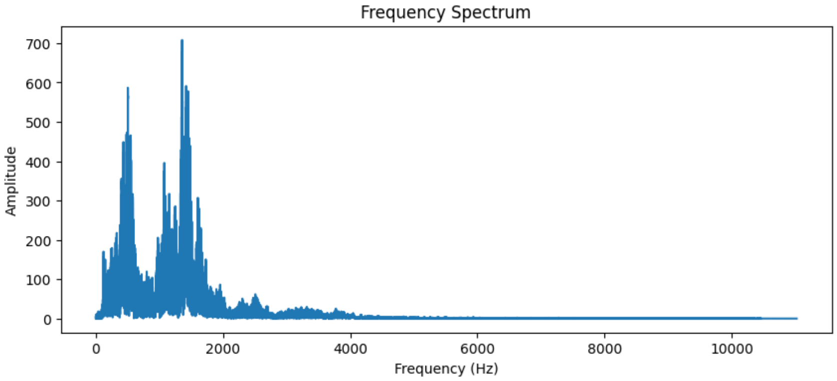


Figure 15: Frequency Spectrum of ’common_voice_en_100155.mp3’

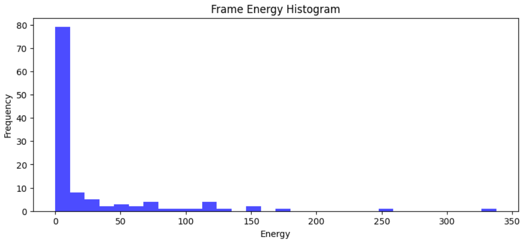


Figure 16: Frame energy histogram

compression and coding. Overall, the frame energy histogram serves as a foundational tool with diverse applications in audio signal analysis and processing [fig.16][8].

The average sample rate for the audio data is 48,000 Hz, representing the number of samples captured per second. This parameter is fundamental in defining the fidelity and resolution of the audio signal.

After that we iterated through a list of MP3 files in the directory, loading each file with a given average sample rate using Librosa. For each MP3 file, it generates two visualizations: a waveform and a spectrogram. The waveform plot illustrates the amplitude of the audio signal over time, while the spectrogram visualizes the frequency content. Both visualizations are saved as PNG files in their respective directories. The code concludes by printing a ’Processing complete’ message, indicating the successful generation of visualizations for all MP3 files in the specified directory.

We also iterated through all selected audio filed for calculating spectral power density. After that calculated average spectral power to understand how the power of a signal is distributed across different frequency components [fig.17].

For understanding the temporal length of the recording and facilitating precise time-based analyses duration of each audio file in seconds is calculated:

The mean duration of a set of 20 files is 3.70 seconds, indicating the central tendency of their individual durations.

A Mel spectrogram visually represents the frequency spectrum of an audio signal, emphasizing human pitch perception through the use of a mel scale. Its application is crucial in tasks like speech and music analysis, where capturing perceptually relevant features matters. Mel spectrograms address limitations in traditional spectrograms, providing a more human-centric representation of sound.

In processing 20 audio files, Mel spectrograms were generated and stored in a directory for analysis or machine learning applications. This step is essential for extracting meaningful features, contributing to tasks like speech recognition. The Mel spectrogram of "common_voice_en_100155.mp3" exemplifies this transformation of audio data into a visual format [Fig.18].

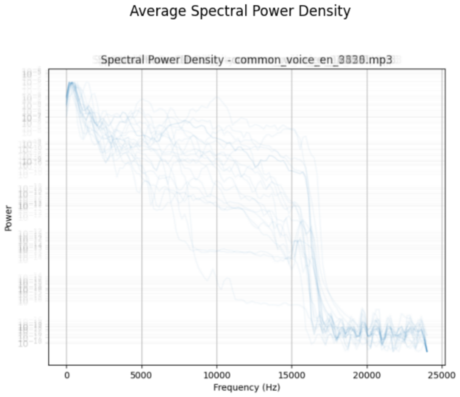


Figure 17: Average spectral power density

| Audio file | Duration(s) | Audio file | Duration(s) |
|---|---|---|---|
| common_voice_en_7.mp3 | 2.47 | common_voice_en_100155.mp3 | 2.06 |
| common_voice_en_2918.mp3 | 4.01 | common_voice_en_3829.mp3 | 3.46 |
| common_voice_en_8450.mp3 | 3.94 | common_voice_en_9750.mp3 | 4.49 |
| common_voice_en_9334.mp3 | 1.94 | common_voice_en_10127.mp3 | 3.82 |
| common_voice_en_3478.mp3 | 4.01 | common_voice_en_8840.mp3 | 3.26 |
| common_voice_en_1839.mp3 | 5.81 | common_voice_en_2125.mp3 | 6.94 |
| common_voice_en_8034.mp3 | 5.38 | common_voice_en_10699.mp3 | 3.79 |
| common_voice_en_3139.mp3 | 2.06 | common_voice_en_1592.mp3 | 5.26 |
| common_voice_en_1426.mp3 | 0.58 | common_voice_en_9139.mp3 | 3.46 |
| common_voice_en_2528.mp3 | 3.26 | common_voice_en_3478 (1).mp3 | 4.01 |

Table 5: Durations of the audios

Zero crossing rate, a feature in audio signal processing, quantifies the signal's rate of sign changes. It's useful for characterizing noisiness or percussiveness. A higher rate indicates a more complex signal, while a lower rate suggests a smoother one. This feature is valuable in distinguishing between sounds and identifying patterns in speech signals.

The total zero crossings in the provided audio files vary widely, ranging from 1320 in "common_voice_en_1426.mp3" to 30465 in "common_voice_en_2125.mp3." This metric reflects the frequency of signal sign changes, offering insights into the complexity and noisiness of the respective audio signals. At the same time the average zero-crossing rate across all files is 1028.55.

Chromagrams visually represent the energy distribution across musical pitch classes.High-intensity regions indicate specific pitches or notes in the audio.Abrupt color intensity shifts reveal changes in harmonic content, marking transitions in the musical structure. Chromagrams aid in recognizing chord progressions by highlighting recurring patterns.Sudden chromagram shifts signal alterations in the musical key.In addition to pitch analysis, chromagrams can offer insights into timbre when

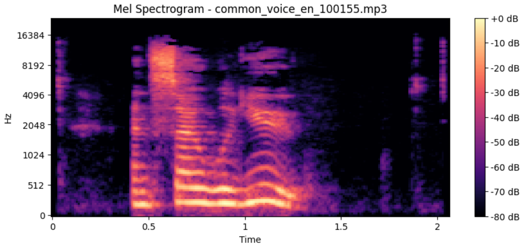


Figure 18: Mel spectrum of 'common_voice_en_100155.mp3'

| Audio file | Total zero crossings | Audio file | Total zero crossings |
|---|---|---|---|
| common_voice_en_7.mp3 | 6390 | common_voice_en_100155 | 9716 |
| common_voice_en_2918.mp3 | 13102 | common_voice_e_3829.mp3 | 14201 |
| common_voice_en_8450.mp3 | 19566 | common_voice_en_9750.mp3 | 5938 |
| common_voice_en_9334.mp3 | 5004 | common_voice_en_10127.mp3 | 15828 |
| common_voice_en_3478.mp3 | 22628 | common_voice_en_8840.mp3 | 6828 |
| common_voice_en_1839.mp3 | 18630 | common_voice_en_2125.mp3 | 30465 |
| common_voice_en_8034.mp3 | 9358 | common_voice_en_10699.mp3 | 7171 |
| common_voice_en_3139.mp3 | 5751 | common_voice_en_1592.mp3 | 16613 |
| common_voice_en_1426.mp3 | 1320 | common_voice_en_9139.mp3 | 11891 |
| common_voice_en_2528.mp3 | 14134 | common_voice_en_3478 (1).mp3 | 22628 |

Table 6: Total zero crossings for each Audio file

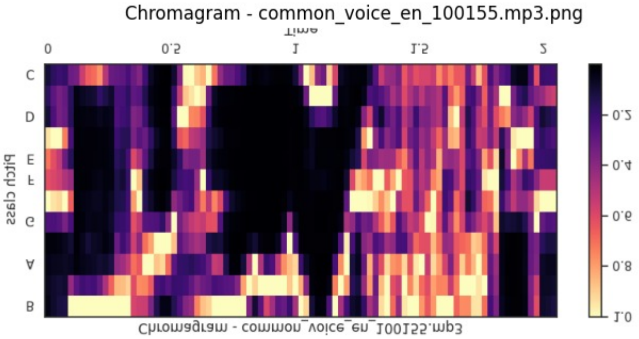


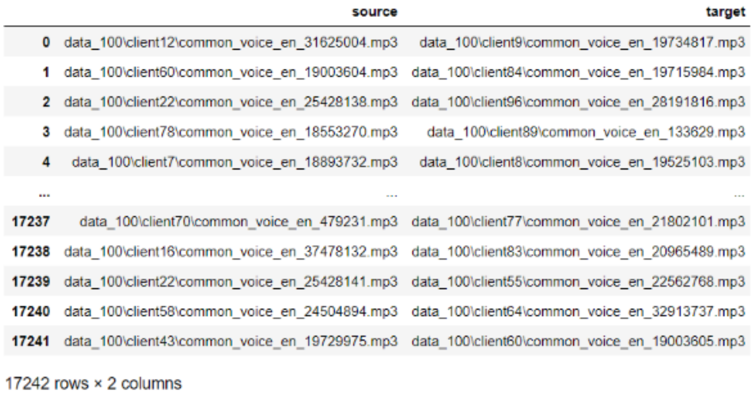
Figure 19: Chromagram of ’common_voice_en_100155.mp3’

combined with other features. This makes them a versatile tool for understanding audio characteristics [Fig.19].

## 4 Feature Extraction

The process of feature extraction plays a pivotal role in understanding and characterizing the audio data, enabling the identification of crucial attributes for further analysis [5].

- Source to Target Mapping: The initial step involved mapping source to target MP3 audio files. This process included the extraction of MP3 audio files from the top 100 client IDs in the ’train.tsv’ dataset. To ensure data quality, any empty audio files were removed, and rows with missing values in the ’age,’ ’gender,’ and ’accents’ columns were excluded. For comprehensive mapping, each client ID was mapped with the other client IDs, with a specific focus on avoiding duplication to eliminate redundancy. As a result, source and target MP3 audio files from different client IDs were effectively paired and stored together [Fig.20].
- Exraction of MFCC features : MFCC stands for Melfrequency ceptral coefficient of audio file.Initially, MFCC values were extracted from each MP3 file taken under consideration, capturing essential spectral features of the audio data. We are computing 13 MFCCs and then modifying the resulting matrix to ensure it has a fixed length of 300 frames by either padding or truncating. In our code , "librosa.feature.mfcc" is handling most of the steps involved in the mfcc feature extraction. It computes the MFCC directly from the audio signal ’y’ using a specified number of MFCC i.e n_mfcc=13 and sampling rate of sr.
- Prosodic Features: Prosodic features encapsulate the variations in pitch, duration, and rhythm of speech, elements that significantly influence the melody and intonation of spoken language. This category of features encompassed chroma features, zero crossings, pitch values, and intensity values. Chroma features are particularly relevant in the realm



| | source | target |
|---|---|---|
| 0 | data_100\client12\common_voice_en_31625004.mp3 | data_100\client9\common_voice_en_19734817.mp3 |
| 1 | data_100\client60\common_voice_en_19003604.mp3 | data_100\client84\common_voice_en_19715984.mp3 |
| 2 | data_100\client22\common_voice_en_25428138.mp3 | data_100\client96\common_voice_en_28191816.mp3 |
| 3 | data_100\client78\common_voice_en_18553270.mp3 | data_100\client89\common_voice_en_133629.mp3 |
| 4 | data_100\client7\common_voice_en_18893732.mp3 | data_100\client8\common_voice_en_19525103.mp3 |
| ... | ... | ... |
| 17237 | data_100\client70\common_voice_en_479231.mp3 | data_100\client77\common_voice_en_21802101.mp3 |
| 17238 | data_100\client16\common_voice_en_37478132.mp3 | data_100\client83\common_voice_en_20965489.mp3 |
| 17239 | data_100\client22\common_voice_en_25428141.mp3 | data_100\client55\common_voice_en_22562768.mp3 |
| 17240 | data_100\client58\common_voice_en_24504894.mp3 | data_100\client64\common_voice_en_32913737.mp3 |
| 17241 | data_100\client43\common_voice_en_19729975.mp3 | data_100\client60\common_voice_en_19003605.mp3 |

17242 rows × 2 columns

Figure 20: Source to target mapping

of music analysis as they excel in capturing tonal information, facilitating chord detection and harmonic assessment. Zero crossings contribute to the detection of abrupt signal shifts, proving valuable in audio processing for identifying onsets and silent periods within audio recordings. Pitch values hold a fundamental role in determining the fundamental frequency of audio signals, a parameter employed in various applications such as music transcription and voice analysis. Additionally, intensity values provide insights into the amplitude of audio signals, thereby aiding tasks related to speech processing, emotion recognition, and audio classification.

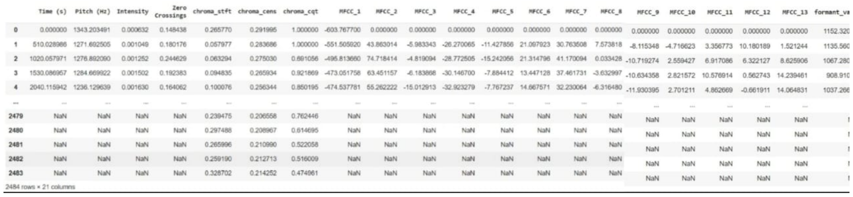
- Supra-Segmental Features: This category of features pertains to the modification of elements like stress, prosody, and speaking rate, which encompass multiple speech segments, ultimately influencing the overall expressive qualities of speech. Supra-Segmental features encompassed Mel-Frequency Cepstral Coefficients (MFCC) values and formant values. MFCC values are employed to describe the spectral characteristics of speech, making them invaluable for feature extraction and the identification of phonetic information. Formant values are instrumental in identifying the resonance frequencies within the vocal tract, a crucial aspect in comprehending vowel sounds and identifying distinct speech sounds and their articulatory features. These features collectively enhance our ability to comprehend and analyze the intricacies of spoken language and contribute to the broader understanding of audio data [9] [1].

The process of feature extraction from both source and target MP3 audio files is a crucial step in data preparation, enabling the extraction of valuable information for subsequent analysis and modeling. These extracted features were thoughtfully organized into a structured format. Specifically, each set of extracted features for a particular source and its corresponding target audio was meticulously paired within a list of tuples. Each tuple within this list represented a cohesive unit, housing the extracted features dataframe for the source and the corresponding target audio. This approach ensured that the relationship between the source and target data remained intact, making it easier to leverage these features for further analysis [Fig.20].

To safeguard and retain the integrity of this mapped feature data for future reference and analysis, the list containing these tuples was stored in a pickle file. This pickle file serves as a convenient and efficient means of data storage, allowing for easy access and retrieval of the extracted features in subsequent stages of the project. By storing this valuable data in a pickle file, it is readily available for future analysis, experimentation, and model development, streamlining the research process and promoting the reusability of this valuable dataset.

## 5 Pipeline used for voice conversion

The step by step process used in our project for voice conversion is shown in [fig.22] For the processing we have considered 20 files, and made the source-target pairs of the parallel audios. We have mapped these parallel audios and saved them for reuse in "mapping_df.pkl". So this pkl has



| | Time (s) | Pitch (Hz) | Intensity | Zero Crossings | chroma_stft | chroma_cens | chroma_cqt | MFCC_1 | MFCC_2 | MFCC_3 | MFCC_4 | MFCC_5 | MFCC_6 | MFCC_7 | MFCC_8 | MFCC_9 | MFCC_10 | MFCC_11 | MFCC_12 | MFCC_13 | formant_va |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.000000 | 1343.203491 | 0.000632 | 0.148438 | 0.265770 | 0.291995 | 1.000000 | -603.767700 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1152.320 |
| **1** | 510.028986 | 1271.692505 | 0.001049 | 0.180176 | 0.057977 | 0.283686 | 1.000000 | -551.505920 | 43.863014 | -5.983343 | -26.270065 | -11.427856 | 21.097923 | 30.763508 | 7.573818 | -8.115348 | -4.716623 | 3.356773 | 10.180189 | 1.521244 | 1135.560 |
| **2** | 1020.057971 | 1276.892090 | 0.001252 | 0.244629 | 0.063294 | 0.275030 | 0.691056 | -495.813660 | 74.718414 | -4.819094 | -28.772505 | -15.242056 | 21.314796 | 41.170094 | 0.033428 | -10.719274 | 2.559427 | 6.917086 | 6.322127 | 8.625906 | 1067.280 |
| **3** | 1530.086957 | 1284.669922 | 0.001502 | 0.192383 | 0.094835 | 0.265934 | 0.921869 | -473.051758 | 63.451157 | -6.183868 | -30.146700 | -7.884412 | 13.447128 | 37.461731 | -3.632997 | -10.634358 | 2.821572 | 10.576914 | 0.562743 | 14.239461 | 908.910 |
| **4** | 2040.115942 | 1236.129639 | 0.001630 | 0.164062 | 0.100076 | 0.256344 | 0.850195 | -474.537781 | 55.262222 | -15.012913 | -32.923279 | -7.767237 | 14.667571 | 32.230064 | -6.316480 | -11.930395 | 2.701211 | 4.862669 | -0.661911 | 14.064831 | 1037.266 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **2479** | NaN | NaN | NaN | NaN | 0.239475 | 0.206558 | 0.762446 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| **2480** | NaN | NaN | NaN | NaN | 0.297488 | 0.208967 | 0.614695 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| **2481** | NaN | NaN | NaN | NaN | 0.265996 | 0.210990 | 0.522058 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| **2482** | NaN | NaN | NaN | NaN | 0.259190 | 0.212713 | 0.516009 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| **2483** | NaN | NaN | NaN | NaN | 0.328702 | 0.214252 | 0.474961 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |

2484 rows × 21 columns

Figure 21: Extracted features

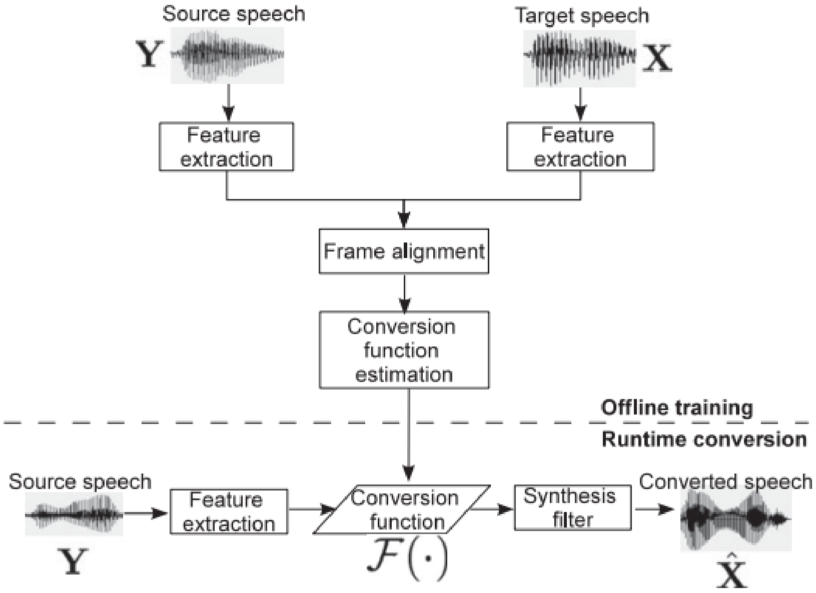
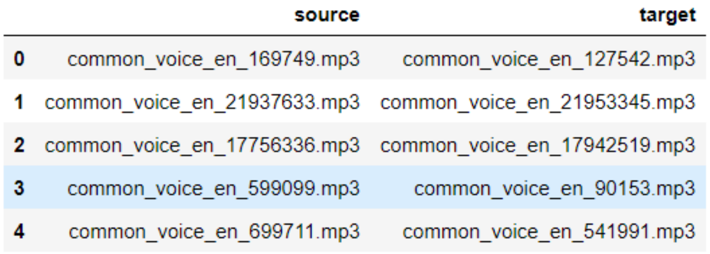


**Fig. 1.** Diagram of a typical voice conversion system.

Figure 22: Pipeline for voice conversion

5 pairs of parallel audios. As shown in [fig.23] From this we have selected a pair and made one source and one target. We have firstly, trimmed and normalised source and target audio files and then removed the noise for better processing. Next we have extracted 13 MFCCs for source and target audio file. We have noramlised the mfcc values of source and target audio and applied padding to it for DTW calculation. DTW stands for Dynamic Time Warping which is used to measure the similarity between two audio sequences that may vary in speed or duration. It aligns them in a way that minimises the difference between them [2]. The visualization of the alignment path after DTW calculation is shown in[fig.24]



| | source | target |
|---|---|---|
| **0** | common_voice_en_169749.mp3 | common_voice_en_127542.mp3 |
| **1** | common_voice_en_21937633.mp3 | common_voice_en_21953345.mp3 |
| **2** | common_voice_en_17756336.mp3 | common_voice_en_17942519.mp3 |
| **3** | common_voice_en_599099.mp3 | common_voice_en_90153.mp3 |
| **4** | common_voice_en_699711.mp3 | common_voice_en_541991.mp3 |

Figure 23: mapping of parallel audios

The warping path is added and aligned source is obtained. Next we append the delta features to aligned features of source and target file.

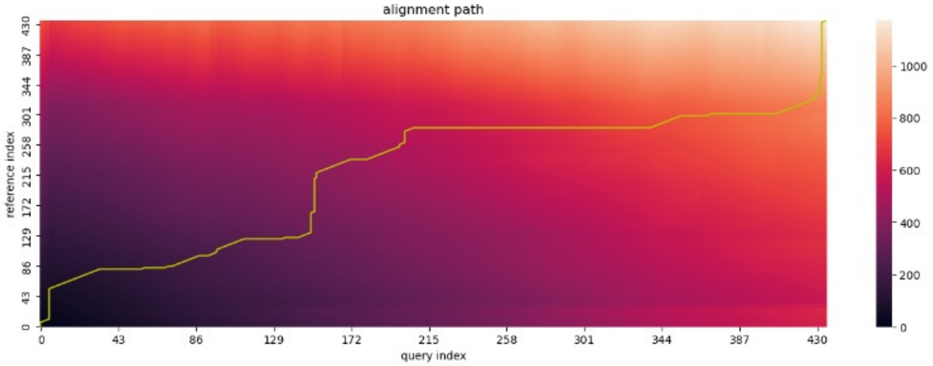


Figure 24: Alignment path

Then we combine the final aligned features of source and target audio files into single joint feature matrix[4].

We use this matrix named as ’XY’ in our code for training using GMM. The visualization of the mean of the 3 mixtures are given in [fig.25] The visualization of the diagonal part of covariance matrix is given in [fig.26] Next we have applied MLPG on the trained model object ’gmm’. MLPG stands for Maximum Likelihood Parameter Generation. MLPG is a technique used to generate maximum likelihood parameter estimates.The output of this process is likely a modified or smoothed version of the parameters estimated by the GMM, considering the windowing and differential features applied during MLPG. This is stored in ’paramgen’ object [8][7]. Next ’paramgen’ is used to transform the aligned features of source audio. The purpose of applying MLPG transformation is likely to modify or enhance the features in src_aligned based on the specified windows, differential features, or other configurations set up within the paramgen object. The transformed data is stored in the variable src_aligned.

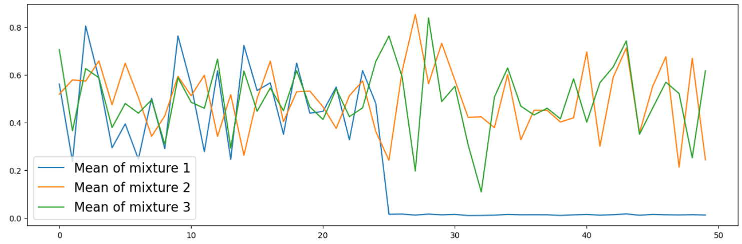


Figure 25: Mean of mixture

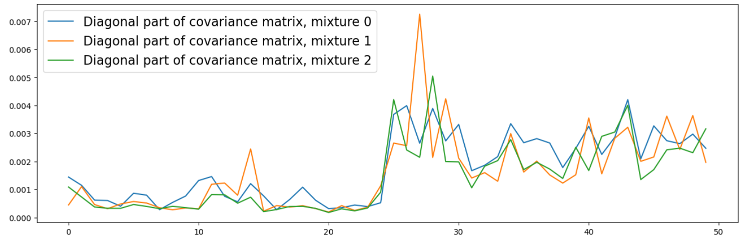


Figure 26: Diagonal part of covariance matrix

We use this transformed src_aligned features to convert it back to its original audio representation. So we apply the appropriate synthesis and get the converted audio.

## 6 Conclusion

This project explored the significance of voice conversion in speech processing, authentication, and human-computer interaction. Detailed analysis of the 'Common Voice Corpus 14.0' dataset provided valuable insights into demographic diversity, linguistic characteristics, and audio content.

After the entire process, we were supposed to get the converted audio similar to voice of target audio.

## 7 Result

In the converted audio after the conversion, we have obtained some random noise due to which the resulted audio is not that clear.

## References

[1] Ryo Aihara, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Gmm-based emotional voice conversion using spectrum and prosody features. *American Journal of Signal Processing*, 2(5):134–138, 2012.

[2] Hadas Benisty and David Malah. Voice conversion using gmm with enhanced global variance. 2011.

[3] Mostafa Ghorbandoost, Abolghasem Sayadiyan, Mohsen Ahangar, Hamid Sheikhzadeh, Abdoreza Sabzi Shahrebabaki, and Jamal Amini. Voice conversion based on feature combination with limited training data. *Speech Communication*, 67:113–128, 2015.

[4] Rabul Hussain Laskar, D Chakrabarty, Fazal Ahmed Talukdar, K Sreenivasa Rao, and Kalyan Banerjee. Comparing ann and gmm in a voice conversion framework. *Applied Soft Computing*, 12(11):3332–3342, 2012.

[5] Seyed Hamidreza Mohammadi and Alexander Kain. An overview of voice conversion systems. *Speech Communication*, 88:65–82, 2017.

[6] Imen Ben Othmane, Joseph Di Martino, and Kaïs Ouni. Improving the computational performance of standard gmm-based voice conversion systems used in real-time applications. pages 1–5, 2018.

[7] B Ramani, MP Actlin Jeeva, P Vijayalakshmi, and T Nagarajan. A multi-level gmm-based cross-lingual voice conversion using language-specific mixture weights for polyglot synthesis. *Circuits, Systems, and Signal Processing*, 35:1283–1311, 2016.

[8] Tomoki Toda, Alan W Black, and Keiichi Tokuda. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2222–2235, 2007.

[9] Zhizheng Wu and Haizhou Li. Voice conversion versus speaker verification: an overview. *APSIPA Transactions on Signal and Information Processing*, 3:e17, 2014.

[10] Ning Xu, Yibing Tang, Jingyi Bao, Aiming Jiang, Xiaofeng Liu, and Zhen Yang. Voice conversion based on gaussian processes by coherent and asymmetric training with limited training data. *Speech Communication*, 58:124–138, 2014.

["https://github.com/YashThakran/Voice-Conversion/tree/main"]