4. Proposed Framework & Voice Recognition Model for Vernacular Gujarati Dialects

4.1 Introduction

In any Voice recognition system generally following steps are performed i.e. Audio Input, Preprocessing, Feature extraction, Voice Print creation, Model Training, Speaker Identification/ Verification, Decision Making, post-processing. Based on this general model of voice recognition system, for proposed research work Voice Recognition Model for Vernacular Gujarati Dialects is proposed. Various components and sub-components of this model are described in detail in this chapter. This chapter also include the framework development for the model. The framework was created to efficiently develop a voice recognition model. The voice recognition model is a fundamental part of this model that takes in voice recordings, processes them, and then identifies the speaker. To identify the speaker, a number of processing tasks are needed. The development of the model's components and its application are covered in Chapter 5.

4.2 Proposed Framework for Vernacular Voice Recognition of Gujarati Dialects

The primary objective of the proposed framework is to support voice recognition for various regional dialects. The aim of framework is on speaker recognition using variation of the dialect. Carrying out high linguistic accuracy on translating spoken words into text, handling the specificity of pronunciation, pitch and intonation in the case of the regional Gujarati dialects such as Kathiyawadi, Standard Gujarati, Kutchhi and Surti.

As contributing to the efficient and accurate recognition of vernacular voice data, this work defines constituent proposed framework shows in Figure 4-1 is systematically divided into three distinct layers: Input Layer, Preprocessing Layer and Processing Layer.

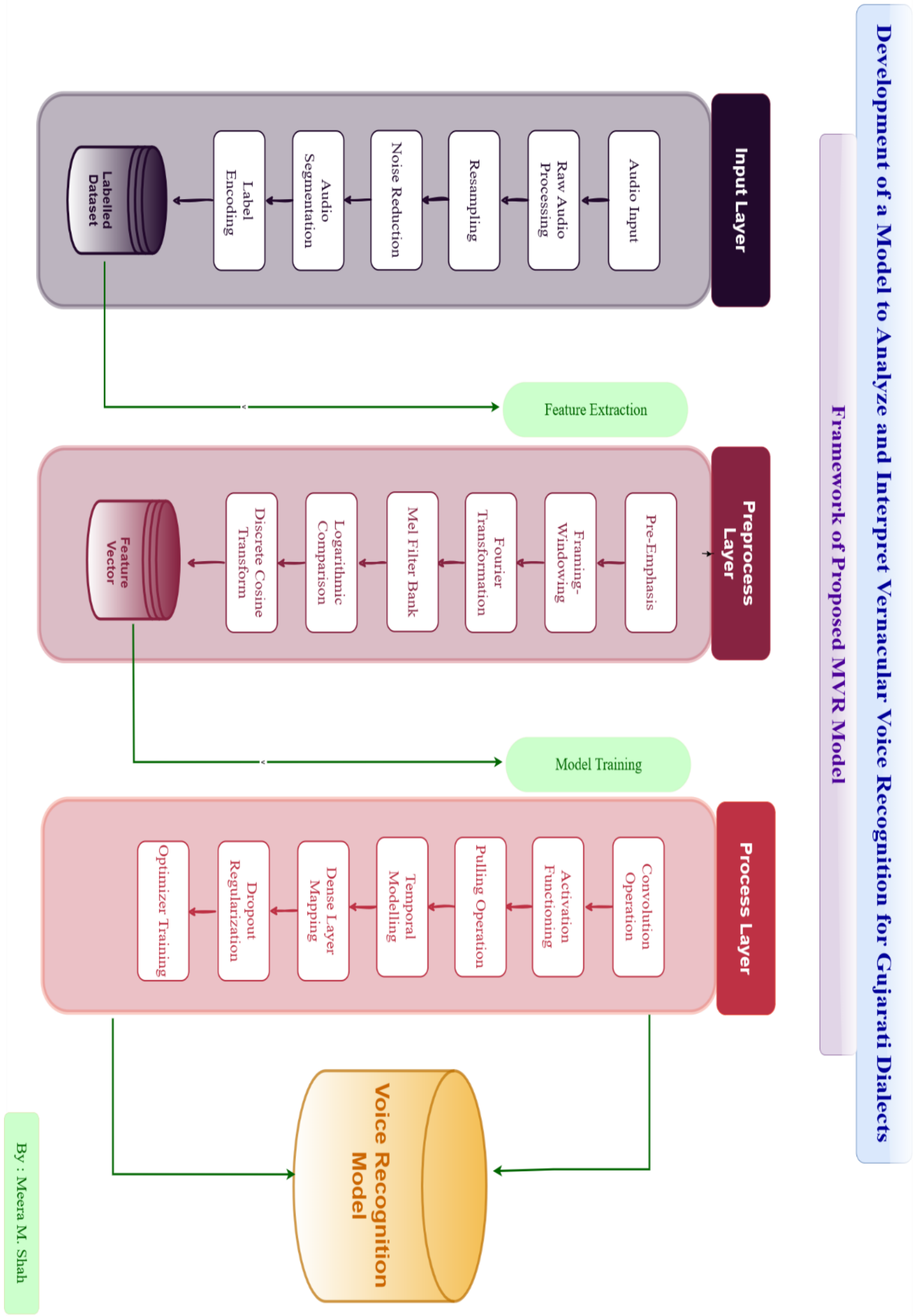

Figure 4- 2 Proposed Framework for Vernacular Voice Recognition Model of Gujarati Dialects

4.2.1 Input Layer

The Input Layer deals with raw audio and takes care to prepare them for further process. This layer contains many critical steps. Reprint is performed to make automated the sampling rates of the audio files, upholding uniformity across the dataset. Raw audio recordings are usually sampled at different rates, and resampling harmonizes these variances, usually setting a governmental rate like 16 kHz to be identical and consistent. We then apply noise reduction in order to remove unwanted ambient noise like background noise or static, while retaining the important speech characteristics. It improves the clarity and quality of the audio data giving an accurate recognition possible.

Second, the audio segmentation step splits a long audio file into shorter sized segments with the specified length, for example, a duration of 7 to 15 seconds. Through this segmentation, the system avoids being overloaded by long recordings and reduces its computational and performance overhead. Then we finally turn each audio file into labels, or machine-readable information, such as the speaker, or dialect, via label encoding. Furthermore, in the case of categorical labels we can transform this into a numerical format so that the classification and recognition tasks become simpler in the subsequent layers.

4.2.2 Preprocessing Layer

The Preprocessing Layer which delivers on the process of extracting meaningful structural features of the labelled dataset. It starts from this layer that transforms raw audio data into a format that can be used further to perform the further processing over extracted features, making sure that the extracted features hold the key characteristics of the audio signal. The preprocessing starts by pre-emphasizing the audio signal to amplify the high frequency components thereof. This reduces the noise and effectively improves the signal to noise ratio, so the noise will not distract you from that speech data. The audio signal is then Fourier Transformed, to switch between the time domain and the frequency domain. The system is able to analyse the frequency components of the signal in this transformation, an important feature for voice recognition. This is followed by filter banking as the frequency domain signal passes through series of overlapping filters.

This step helps in reducing noise and enhancing the signal-to-noise ratio, making the speech data more distinguishable.

This signal is then compared with a logarithm, so the dynamic range of the audio signal is compressed. This step further increases the distinctiveness of the speech features in the sense that the features are mimicking the logarithmic sensitivity of the human hearing. The signal then goes on Discrete Cosine Transformation (DCT), so it removes the redundancy by the frequency components being transformed into a compact set of coefficients. The obtained coefficients in such a reduced dimensional form are the most important information about signal.

This layer result in the final outcome, a feature vector, a condensed numerical description of the audio signal, that contains the structural and acoustic characteristics of the signal. Since this feature vector is needed for future tasks in the Processing Layer which is the case here in classification and recognition algorithms, it is essential. Through these narrowly designed preprocessing steps, the underlying framework prepares the audio data for further processing with an extremely informative and compact representation.

4.2.3 Processing Layer

The last stage of the proposed framework, called Processing Layer, uses the extracted feature vectors for implementing those extra operations to realize the perfect recognition of voice. The feature vectors are then processed by a suite of high-level machine learning and deep learning techniques in this layer, and from these, meaningful predictions are formed. The first upper layer of the proposed network first uses convolution operations with convolutional filters applied to the feature vectors for localized pattern extraction and spatial hierarchy of data. These operations can then be used to decide critical features, such as frequency and phonetic structures in the audio.

Once we apply convolution, we need to introduce non-linearity into the model, so we apply activation functions as they are: ReLU (Rectified Linear Unit). This step is done to let the network learn complex relationships and patterns in the data which are crucial to recognize correct. Then down sampled, dimensionally reduced feature maps are pool operations, while still retaining as much of the most significant information as possible.

This step improves efficiency of computations and keeps the model from over fitting to most important bits of the data.

Temporal modelling is also added, which captures the sequential nature of the audio data, in the layer. We deal with the temporal dependencies within the speech signals by analysing the temporal dependencies using temporal modelling techniques such as recurrent neural networks (RNNs)[1], [2], [3] or, in particular, long short-term memory (LSTM)[2]networks. Specifically, this allows the model to infer the time varying patterns in the audio that are fundamental for accurate voice recognition.

Then we process the data and pass it through dense layers, where the features extracted are mapped to the target classes i.e., speakers or dialects. High level feature abstraction and learning relationship of input features to output labels are performed by dense layers. During training, we apply dropout regularization, to gain generalization and remedy overfitting. The basic idea of this technique is that when applying this technique on the network, it randomly knocks out a portion of the neurons in the network with each iteration to avoid over-dependency in a specific feature.

Then the model is trained, after all the operations have finished, and the final voice recognition model is generated as output by the framework. The model can accurately tell apart speakers or dialect from the input audio. The Processing Layer to integrate these new advanced operations, guarantees that the framework is able to identify vernacular voices in high accuracy, robustness and efficiency.

4.3 Voice Recognition Model for Gujarati Dialects

Voice Recognition Model works by analysing the unique characteristics of a person's voice, such as pitch, tone, cadence, and speech patterns, to identify or verify their identity. It generally involves two major phases: enrolment and recognition. During the enrolment phase, the system records the voice of a user and extracts distinguishing features, often using techniques like Mel-frequency cepstral coefficients[4], [5] or deep learning models. The spoken features are thereby kept in the database. At the time of recognition, after the utterance from the user into the system, the incoming speech would be taken, features will be extracted and compared against stored data in one of two possible ways, either to verify the claimed identity of an individual (verification) or to determine the identity from a known group of voices.

- The proposed Voice Recognition Model is intended to do the following tasks shows in Figure 4-2:
    - Voice Data Collection
    - Organizing the collected audio files in global directory structure
    - Preprocessing the segmented data making it suitable for feature extraction.
    - Extracting Structural Features from the audio dataset
    - Voice Recognition Model
    - Classifications

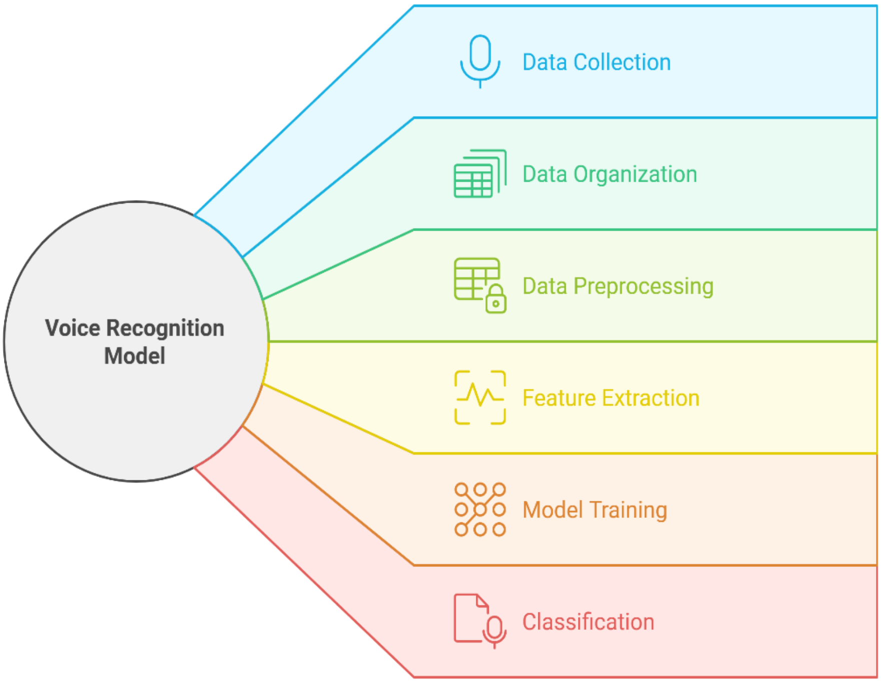

Figure 4- 5 Voice Recognition System for Vernacular Gujarati Dialects

4.3.1 Speech Data Collection

There are many ready to use dataset available for voice recognition in different languages which includes LDC-IL[6] Gujarati Raw Speech data set consists of different types of datasets that are made up of word lists, sentences, texts and date formats. Approximately 15 minutes of speech (per speaker) has taken from 96 female and 108 males from Gujarati's mother tongue speakers of different age groups. Each speaker recorded these datasets which are randomly selected from a master dataset[6].

Shrutilipi is a labelled ASR corpus obtained by mining parallel audio and text pairs at the document scale from All India Radio news bulletins for 12 Indian languages - Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Odia, Punjabi, Sanskrit, Tamil, Telugu, Urdu. The corpus has over 6400 hours of data across all languages[7].

The datasets offered by AI4Bharat [8] are accessible in multiple languages and cover a variety of speech processing tasks.

For the proposed model speech data for vernacular Gujrati language are collected for experimental work from different public resources and that organized based on experimental requirement that describe in detail in section 5.3.

4.3.2 Speech Data Organization

The data collected from different public resources are organized manually in order that it can be used easily for the further preprocessing tasks for Proposed Model. The collected data were arranged in directory formats according to dialect variations and gender.

The speech data is carefully organized manually from various public resources to make it usable for further preprocessing tasks required by the Proposed Model. The data is arranged methodically into a form that is consistent with the expected input of subsequent workflow stages, including feature extraction, normalisation and model input preparation.

To achieve this, the collected data is categorized and arranged in a directory-based structure based on two primary attributes: dialect variations and gender. Each dialect has its own directory with further separated folders for male and female speakers. The structure is hierarchical, thus allowing researchers to retrieve specific subsets of data and proceed with the recording, preprocessing and analysis. Say, for instance, researchers can fine tune their focus to a specific dialect or contrast male and female speech patterns within a given linguistic group — without other filtering.

But equally important as uploading the files to the proprietary website, is the organization process, which guarantees each audio file is given a meaningful filename and comes with metadata attached. The key required metadata includes dialect, gender, recording condition, and source for traceability and downstream model evaluation and cross referencing. The structured approach not only enhances efficiency of preprocessing tasks, but also reduces errors or inconsistency (if any) that could hamper the performance of the Proposed Model.

The manual organization puts forward a solid basis for follow on research and analysis by laying down an orderly and in an access key for further research. It facilitates the smooth run of preprocessing workflows, promotes reproducibility and generally improves the quality of the speech data which is being used in the course of model building. The critical importance of the data organization is clearly seen in this meticulous preparation.

4.3.3 Voice Recognition Model

The Voice Recognition Model will take audio as an input and will perform certain series of steps to identify the speaker. Main Objective of Voice Recognition Model is to identify the speaker based on given voice in audio format. To attain this objective the MVR Model is used which performs following tasks as shown in Figure 4-3. The model basically divides in two segments (1) Training Phase and (2) Recognition Phase. The Training phase involves the tasks like Data Organization, Preprocessing, Feature Extraction, Classification using deep learning modelling techniques and the Recognition phase involves to use the trained model to identify the speaker’s and its voice based on its characteristics that described in detail in following subsequent sections. This model will be able to identify the speaker which speaks Gujarati Language of any regional dialects from Kathiyawadi, Standard Gujarati, Kutchhi or Surti.

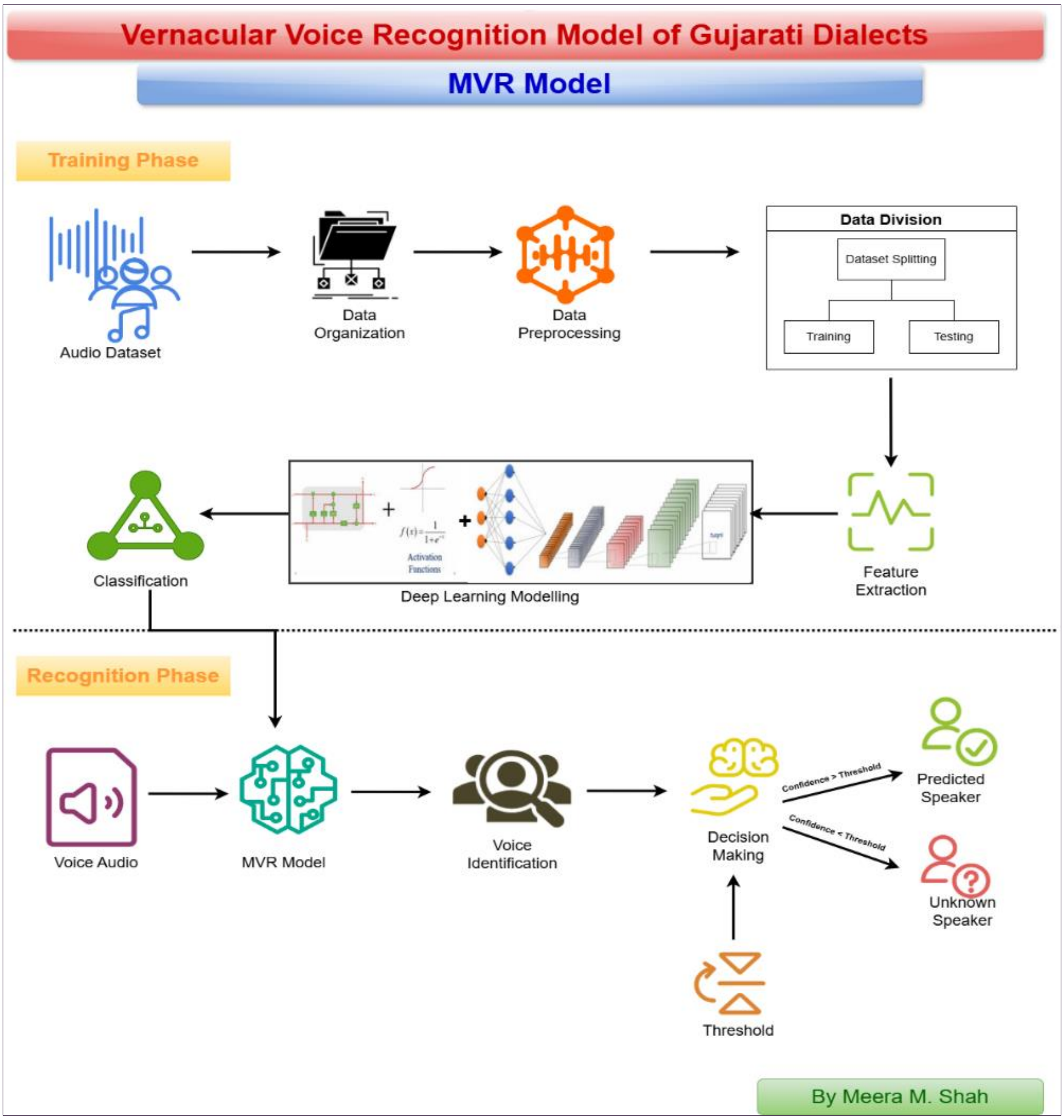

Figure 4- 8 Vernacular Voice Recognition Model of Gujarati Dialects (MVR Model)

4.3.3.1 Data Preprocessing

Preprocessing step is a preliminary step to be performed on acquired audio, which involves certain operations as shown in Figure 4-4 to provide a necessary base to perform further tasks of Voice Recognition. If audio recorded in noisy environment or it is not in a proper format then it directly affects the performance of voice recognition. Preprocessing does the task of enhancing audio making it suitable input for segmentation or feature extraction. Preprocessing involves series of steps to be performed as describe in following subsequent sections.

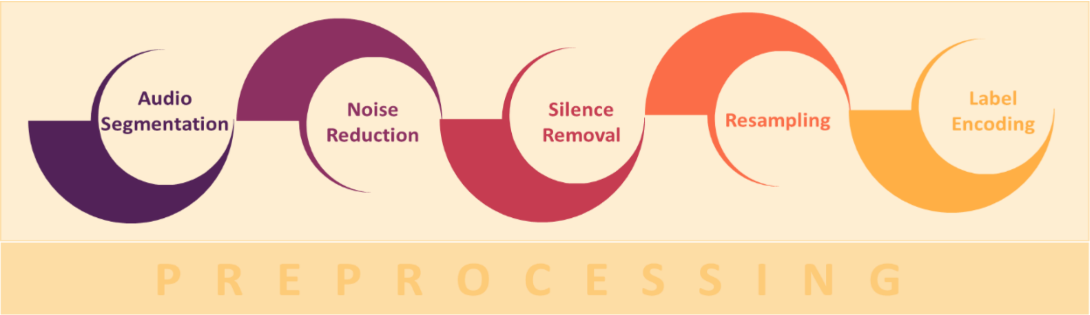

Figure 4- 11 Data Preprocessing Operations

4.3.3.1.1 Audio Segmentation

Audio segmentation in speaker recognition involves dividing audio recordings into smaller, meaningful segments to isolate individual speakers or speech utterances for analysis. This step is essential for efficient feature extraction and accurate model training. Technique of fixed-length segmentation for uniformity were used during the preprocessing task. Segmented audio is standardized, saved in structured directories, and often paired with metadata for further processing.

4.3.3.1.2 Noise Reduction

One of the most important aspects before using any speech data is to descend the noise from the audio preprocessing because it reduces unwanted background sounds like hums, static, environmental noise, etc. which helps maintaining the clarity of the speaker's voice. For example, techniques of spectral subtraction measure and remove noise from the audio signal by subtracting the noise profile from the audio, and bandpass filtering only within human speech range to 16 kHz. By reducing noise, we eliminate background noise and are left with clearer audio files so that features used can be extracted properly and the model as a whole works better.

4.3.3.1.3 Silence Removal

Types of audio preprocessing include silence removal whereby useful data consisting only of meaningful speech content is extracted from a spoken audio segment. This process is important for those applications like speaker recognition, speech to text and emotion detection, as it helps to reach dimensionality reduction of the dataset, filtering, and improves the model efficiency by focusing on the characteristics of the speech. Silence removal can be done using techniques like energy thresholding where by low energy segments below a specified threshold is eliminated. Optimal suppression of silence enhances the speed of computation, retains higher quality data for feature extraction and enhances performance of subsequent audio processing tasks.

4.3.3.1.4 Resampling

Resampling in audio preprocessing involves converting the sample rate of an audio file to a desired frequency, ensuring consistency across a dataset and compatibility with processing tools or machine learning models. The sample rate refers to the number of audio samples captured per second, typically measured in Hertz (Hz). Common sample rates include 16kHz widely used in speech recognition and 8kHz for telephone-quality audio.

Resampling is essential for tasks like speaker recognition, where inconsistent sample rates can lead to inaccurate feature extraction or model training. During resampling, the audio is interpolated to maintain fidelity while adjusting the number of samples per second.

4.3.3.1.5 Label Encoding

Label encoding in audio preprocessing involves converting categorical labels, such as speaker names or identities, into numerical values for compatibility with machine learning models. Each unique category is mapped to a distinct integer, ensuring consistent and structured data representation. This process is crucial for tasks like speaker recognition, where each speaker is assigned a unique ID for model training and evaluation. Label encoding ensures the dataset is ready for multi-class classification, enabling efficient and accurate processing in downstream tasks.

4.3.3.2 Feature Extraction

The First step is loading of the dataset for the proposed model which includes audio files with corresponding labels about the gender of the speakers. This will ensure that the audio files are correctly matched with their labels and follow a uniform format for the smooth conduct of the process in subsequent stages.

Feature extraction is the next step after pre-processing, in which the important features are elicited from the audio. It would transform raw audio signals into numerical representations that capture important aspects of the sound and eventually make them amenable for analysis or classification.

The data will be prepared after feature extraction by combining features with their labels in structured form, and then encoding these labels to numerical values that are compatible with machine learning. This will make sure the data is well-structured for training and testing.

4.3.3.3 Classification

By using the voice recognition engine, the audio data were classified by labelling, based on what it has learned. Results are measured by the model's performance in terms of accuracy, among other relevant metrics that show the effectiveness of the model. Classification defines classifying speaker to some group which is having unique characteristics. For proposed approach to classify speakers, the deep learning algorithms were used. After classifying a speaker in particular group based on some local features extracted speaker is identified.

4.3.3.4 Display the Results

Final outcome of the Voice Recognition Model for Vernacular Gujarati Dialects is to identify the speaker based on their unique vocal characteristics and Gujarati dialects. After recognition the next step is to display or store recognized speaker. For proposed approach Voice recognition for Gujarati Dialects is considered so representation is used to display editable text of recognized speaker.

References

[1] R. Ranjan, S. K. Singh, R. Kala, and R. Kumar, “Multilingual Speaker Recognition Using Neural Network Static hand gesture recognition using Deep Learning View project Expert System for Speaker Identification Using Lip Features with PCA View project MULTILINGUAL SPEAKER RECOGNITION USING NEURAL NETWORK,” 2009. [Online]. Available: https://www.researchgate.net/publication/272086352

[2] N. N. Prachi, F. M. Nahiyan, M. Habibullah, and R. Khan, “Deep Learning Based Speaker Recognition System with CNN and LSTM Techniques,” in 2022 International Conference on Interdisciplinary Research in Technology and Management, IRTM 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IRTM54583.2022.9791766.

[3] D. Sztahó, G. Szaszák, and A. Beke, “Deep learning methods in speaker recognition: a review.”

[4] M. A. Hossan, S. Memon, and M. A. Gregory, “A novel approach for MFCC feature extraction,” in 4th International Conference on Signal Processing and Communication Systems, ICSPCS’2010 - Proceedings, 2010. doi: 10.1109/ICSPCS.2010.5709752.

[5] M. Singh, “Speaker Identication using MFCC Feature Extraction ANN Classication Technique,” 2023, Doi: 10.21203/rs.3.rs-2407488/v1.

[6] Ramamoorthy L. et al., “Gujarati Raw Speech Corpus. Central Institute of Indian Languages, Mysore.” 2021.

[7] K. S. Bhogale et al., Effectiveness of Mining Audio and Text Pairs from Public Data for Improving ASR Systems for Low-Resource Languages. arXiv. doi: 10.48550/ARXIV.2208.12666.

[8] K. S. Bhogale et al., “Effectiveness of Mining Audio and Text Pairs from Public Data for Improving ASR Systems for Low-Resource Languages,” arXiv.org, Aug. 26, 2022. https://arxiv.org/abs/2208.12666

[9] S. T. Aung, H. Myo Tun, Z. M. Naing, and W. K. Moe, “Analysis of Speech Features Extraction using MFCCs and PLP Extraction Schemes.” [Online]. Available: www.ijsetr.com