Synopsis

Speech recognition methodologies for Gujarati language

Submitted by

Shardav Umang Bhatt
(FOTE/943)

Towards the partial fulfilment of the requirement of
Doctor of Philosophy in
Applied Mathematics

Under the Supervision of

Dr. Purnima K. Pandit

Department of Applied Mathematics
Faculty of Technology and Engineering
The Maharaja Sayajirao University of Baroda
Vadodara 390001, Gujarat, India.
February 2023

Contents

Chapter 1

Introduction and Literature review

1.1 Introduction

1.1.1 Speech recognition

Automatic Speech Recognition (ASR) is a procedure of designing an intelligent machine that can automatically recognise a natural human speech using the information contained in the digital speech signal. It is an interdisciplinary task [1]. Research in this area requires knowledge of signal processing, acoustics, pattern recognition, linguistics, physiology, computer science and applied mathematics [1]. Challenge to do ASR is that the recorded or otherwise real time speech signals include the surrounding noise. Moreover, there may be variations in pronunciations and accent also differ over geography. Hence, more research is required in this field to make an accurate ASR system which can recognise human speech without any error [2].

1.1.2 Types of Speech recognition

The speech recognition is separated in different classes, based on the type of utterance or the ability that they have to recognize, as below

- Isolated word recognition: It requires a single utterance at a time. It set necessary condition that each utterance having little or no noise on both sides of sample window.
- Connected word recognition: It requires minimum pause between utterances to make speech flow smoothly. They are similar to isolated words.
- Continuous speech recognition: It requires continuous speech by human speech, without silent pauses between words. It is much more difficult as compared to isolated word recognition.
- Spontaneous speech recognition: It can recognize a speech that is natural sounding and not tried out before. It has ability to handle a diversity of natural speech features such as words being run at the same time.

1.1.3 Applications of Speech recognition

Some of the most famous ASR systems are Google Assistant by Google, Cortana by Microsoft, Siri by Apple and Alexa by Amazon. Few of these are also available in regional languages. Such systems are useful for day-to-day tasks like hands–free computing, giving commands to domestic appliance to get our work done, voice dialling and many more.

1.1.4 Speech recognition of Indian languages

According to census of India, there are 22 scheduled languages and 99 unscheduled languages spoken in India. Indian languages contain more phonemes as compared to other languages. They contain more number of retro-flex consonants and fricatives. The accents are non-uniform within same language [3]. Proper articulation is required for correct pronunciation. Hence, recognizing a speech in an Indian regional language is a challenging task. Most of the robust ASR systems are not accurate for Indian languages.

Gujarati language

Gujarati is a language with 13 vowels and 36 consonants. The language is phonetically different from other Indian languages because it has retroflex lateral flap like ળ, which is unique in pronunciation. Moreover, the pronunciation of conjunct consonant જ્ઞ is different as compared to other Indian languages. The vowels એ and ઓ each represents two different pronunciations each [4].

1.1.5 Motivation

ASR in Indian regional languages can be very useful. It can facilitate people who can communicate using their mother tongue only. It can be useful for disabled people who cannot use input devices. Such people can give commands to computer, mobile or other smart gadgets using their speech. The speech user interface in regional Indian languages would be greatly beneficial to such people.

1.2 Literature Review

There are many methods and algorithms applied for designing an ASR system. In 1950s, early ASR systems were based on acoustic–phonetic methods, spectral measurements and analogue filter banks. These systems were updated in 1960s using filter bank spectral analyser, time normalisation methods and dynamic programming methods. Decade of 1970s added more advancement in it by introducing methods based on pattern recognition, clustering algorithms and Linear Predictive Coding (LPC). 1980s gave a revolutionary change in this field with the replacement of template–based methods by statistical modelling method like Hidden Markov Models (HMM) [2]. Artificial Neural Networks (ANN), expert systems and wavelets begun to use for ASR in 1990s. In the 21st century, these systems advanced further due to evolution of Machine Learning (ML) algorithms and improvisation in ANN due to Deep Learning (DL).

1.2.1 Indian languages:

Researchers across India are actively working in the research area of ASR for different Indian regional languages. Several authors has surveyed progress on Speech recognition in Indian languages [5], [6]. Most recent survey can be found in [7]. Survey for Gujarati language is summarised in [8]. From these surveys, it can be observed that very less amount of research work is done for the Gujarati language. One of the major challenges of ASR in Indian languages is the lack of speech corpora [3]. Gujarati is one of the low-resourced languages for speech recognition. Several researchers are working on generating speech corpora for Gujarati and other Indian languages [9], [10]. Also, Speech corpus in Gujarati language for emotional analysis is generated [11].

1.2.2 Gujarati language:

For speech recognition in the Gujarati language, many researchers are using various techniques. Approaches like Vector Quantisation [12], Gaussian Mixture Models [13], Artificial Neural Networks [14] [15] and Support Vector Machines [16] has been used. In recent works, Hidden Markov Models approach is used [17] [18]. Hidden Markov Models together with Artificial Neural Network is used in [19]. Fast bootstrapping was used in [20]. Some authors have developed the phonetic engine to convert speech

sound units into symbols [21]. Syllable boundary detection methods are summarised in [22]. The application "Avaaj Otalo" was developed for people having limited literacy, limited familiarity with technology and knowing Gujarati language only [23]. Speech recogniser of 60 words was prepared for smart-phone operations like calling, sending SMS and other smart-phone operations using Hidden Markov Models [24]. Due to the evolution of Deep Learning (DL) techniques, multilingual speech recogniser for various Indian languages are developed. Some researchers have prepared a multilingual speech recogniser, which can recognise a speech spoken in various Indian languages [25], [26]. End-to-end systems are proposed [27]. Several researchers have proposed multilingual speech recognition using different DL techniques [28], [29], [30], [31]. From the literature review, it can be observed that Gujarati language is a low-resourced language and a limited amount of work is done for the ASR in the Gujarati language. Most of the research works for the Gujarati language are based on feature extraction using Mel-frequency cepstral coefficient. The wavelet based coefficients are yet to be explored for the Gujarati language, for the feature extraction. Moreover machine learning technique like Radial basis function network is also not used for the speech recognition of Gujarati language.

1.3 Objectives

In our work, Speech Recognition in Gujarati language is done pertaining to two objectives

- ASR of isolated words spoken in Gujarati
- ASR for continuous sentences spoken in Gujarati

To achieve these objectives, firstly the existing techniques like Dynamic Time Warping, Machine learning techniques like Artificial Neural Networks, Radial Basis function network were applied on the Mel-frequency Cepstral Coefficient features obtained for speech signal in Gujarati. Here, we have proposed use of wavelets to obtain features for recognition of Gujarati speech. Next objective is done using Hidden Markov Models and Machine Learning models. As a part of this work, a Graphical user interface is also developed using open source platform 'Python'. The steps are summarized in the block diagram as shown in the 1.1.

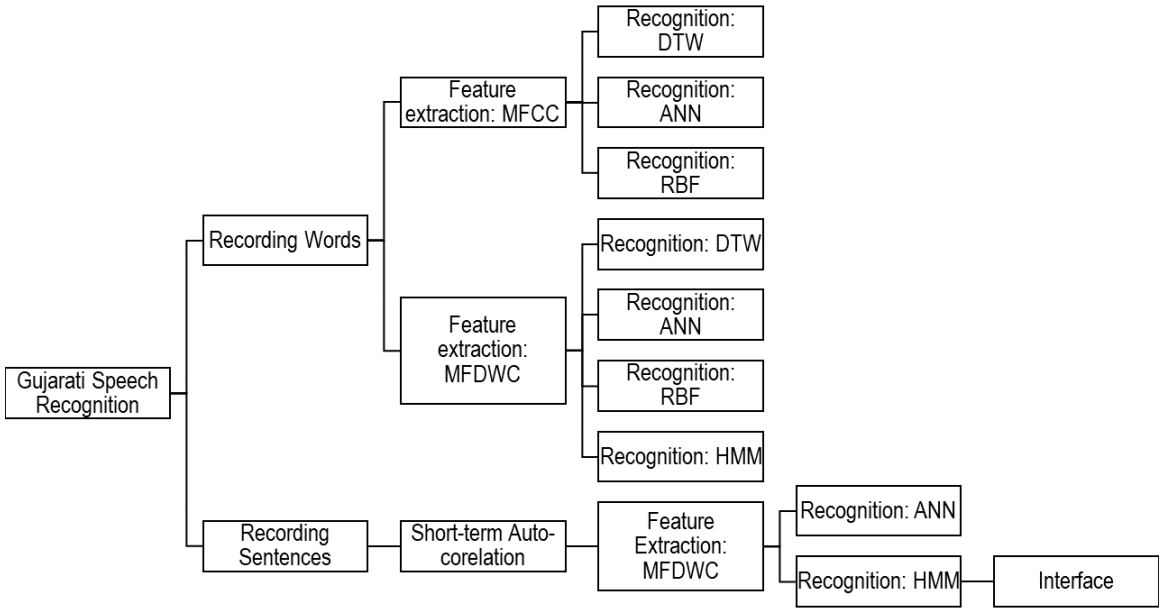

Figure 1.1: Block diagram showing objectives of our research work

Chapter 2

Preliminaries

The methodologies of the speech recognition are mainly divided into three parts: Pre-processing, feature extraction and classification. Following sections explains these three parts in details.

2.1 Pre-processing

This step mainly involves recording of continuous sentences, noise removal and extracting the words from the sentences.

2.1.1 Recording

The recording is usually done using usual mic of PC or Mobile. Common sampling rates of the recording are 8000 or 16000 samples per seconds. The digital format to store speech recording is 16-bit PCM WAV format. The recordings are done in two ways. In the first way, the recordings are done for each word, spoken by each speaker. This corresponds to isolated words. In the other way, whole sentences are recorded per speaker. This corresponds to the continuous speech.

2.1.2 Noise removal

It is important to remove noise from the speech signal so that the features extracted from it are only based on the speech data and not on the noise. A care has to be taken during the recording to tackle noise. After recording an open-source software Audacity is used for the noise removal. Using this, the noise is filtered using the noise reduction algorithm which can reduce the background sounds like hum, whistle, whine, buzz, "hiss", fan noise, FM carrier noise. The algorithm is based on Fourier analysis which finds the spectrum of pure tones which generate the background noise in the quiet sound. This algorithm finds the frequency spectrum of each short segment of audio. Any pure tones that are not sufficiently louder than their average levels in the background noise are reduced in volume.

2.1.3 Short-term autocorrelation

This method is useful to extract words from the sentences [32]. The general formula for determining auto-correlation between two shifted parts of sequence $x(m)$ is

$$r_{xx}(k) = \sum_{m=-\infty}^{\infty} x(m)x(m+k) \tag{2.1}$$

Here, $r_{xx}(k)$ is the k^{th} coefficient of auto-correlation between original signal $x(m)$ and same signal shifted by k samples $x(m+k)$. The value k gives summation of products of original signal with shifted version of itself by k samples. Speech signals are non-stationary. So it is required to divide the signal into several number of short frames. This makes signal stationary within particular short frame. Then

the auto-correlation for each short frame with different values of k is determined using equation (2.1). Hence, it is known as short term auto-correlation [33]. To remove the silence part of speech signal, auto correlations with $k = 1$ are used. Some values of auto-correlation will be high and some are close to zero. From this the voiced frame and the unvoiced frame can be identified.

2.2 Feature extraction

Most commonly used method for feature extraction from speech is mel-frequency cepstral coefficients (MFCC) [2]. The features from the speech are obtained in following way. First the speech signal is pre-emphasised. Let $x(n)$ be the recorded digital speech signal, where x is amplitude and n is a sample number. From this, the pre-emphasised speech signal is obtained using

$$s(n) = x(n) - \alpha x(n), \qquad 0.9 \leqslant \alpha \leqslant 1 \tag{2.2}$$

After that, the pre-emphasised signal $s(n)$ is divided into frames s_i having N samples, where i is the frame index. There is an overlapping of M samples of each frame with its adjacent frame. This gives frames

$$s_i(t), \qquad 1 \leqslant t \leqslant N \tag{2.3}$$

Further, each pre-emphasised frame is multiplied by a Hamming window given by

$$w(t) = (1 - \beta) - \beta \cos\left(\frac{2\pi t}{N-1}\right) \tag{2.4}$$

This gives pre-emphasised windowed frames

$$s_i(t)w(t), \qquad \forall i \tag{2.5}$$

Further, the Fourier transform and power spectrum of each windowed frame are computed using following equations.

$$f_i(k) = \sum_{t=0}^{N-1} s_i(t)w(t)e^{\frac{-2j\pi kt}{N}} \qquad 1 \leqslant k \leqslant \frac{N}{2} \tag{2.6}$$

$$p_i(k) = \frac{1}{N}|f_i(k)|^2 \tag{2.7}$$

Here k represents frequencies. Before going further, these frequencies are converted to mel using

$$M(f) = 1125 \ln\left(1 + \frac{1}{700}\right) \tag{2.8}$$

After that, a filter-bank of triangular filters defined by (2.9) is applied on power spectrum. This gives periodogram estimates of the power spectrum given by (2.10).

$$F_j(k) = \begin{cases} \frac{k - M(j-1)}{M(j) - M(j-1)} & M(j-1) \leqslant k \leqslant M(j) \\ \frac{M(j+1) - k}{M(j+1) - M(j)} & M(j) \leqslant k \leqslant M(j+1) \\ 0 & otherwise \end{cases} \tag{2.9}$$

$$\hat{p}_m = \sum_{k=0}^{K/2} p_i(k)F_j(k) \tag{2.10}$$

Finally, the MFCC features are determined by taking discrete Fourier transform of logarithm of periodogram estimates.

$$\hat{c}_n = \sum_{n=1}^{K/2} \ln\left(\hat{p}_m\right) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right] \tag{2.11}$$

Thus MFCC features are obtained for each frame. However, the wavelets are better for analysing the non-stationary signal such as speech signal, wavelet coefficients can also be used as a part of feature extraction. This method is mel-frequency discrete wavelet coefficients (MFDWC)[34].

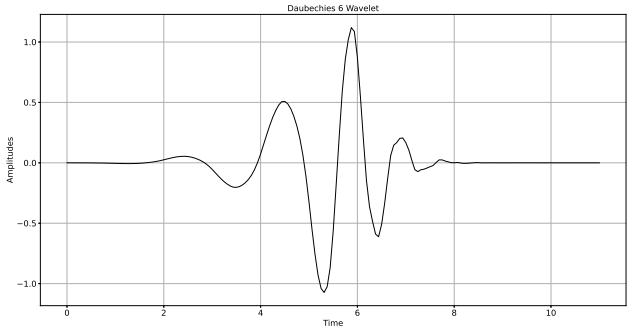

Figure 2.1: Daubechies-6 wavelet function

2.2.1 Wavelets

The word "wavelet" means a small wave. It is a windowed function of finite length. This function is oscillatory so it is called wave. Wavelets are characterized by compact support; it means that the signal does not last forever. Moreover, area underneath the wavelet is zero; it means that the energy is equally distributed in positive and negative directions. So wavelets are rapidly decaying wavelike oscillations that have zero mean and they exist for a finite duration. The Wavelet transform of signal $f(t)$ at scale a and location b is given by

$$W(f(t)) = \int_{-\infty}^{\infty} f(t)\Psi_{a,b}(t)dt \tag{2.12}$$

Here $\Psi_{a,b}(t)$ is is an analysing function which is given by

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-b}{a}\right) \tag{2.13}$$

In above equation, Ψ is a wavelet function satisfying following two properties.

$$\|\Psi(t)\| = 1 \tag{2.14}$$

$$\int_{-\infty}^{\infty} \Psi_t dt = 0 \tag{2.15}$$

A windowed function with zero mean, finite energy, compact support and fast-decaying nature can be a wavelet function. One example of such wavelet function is shown in Figure 2.1. The general expression of a discrete wavelet transforms (DWT) for a discrete signal $X[n]$, having M samples, is given by approximations (2.16) and details (2.17).

$$W_\phi[j_0, k] = \frac{1}{\sqrt{m}} \sum_n X[n]\phi_{j_0,k}[n] \tag{2.16}$$

$$W_\psi[j, k] = \frac{1}{\sqrt{m}} \sum_n X[n]\psi_{j,k}[n], j > j_0 \tag{2.17}$$

Here $\phi_{j_0,k}$ is a discrete scaling function and $\psi_{j,k}[n]$ is a discrete wavelet function having M components each [35].These approximations and details give the DWT of a given signal.

2.2.2 Mel-frequency discrete wavelet coefficients (MFDWC):

The steps of determining MFDWC from the input speech signals are as follows

- Pre–emphasis the digital speech signal

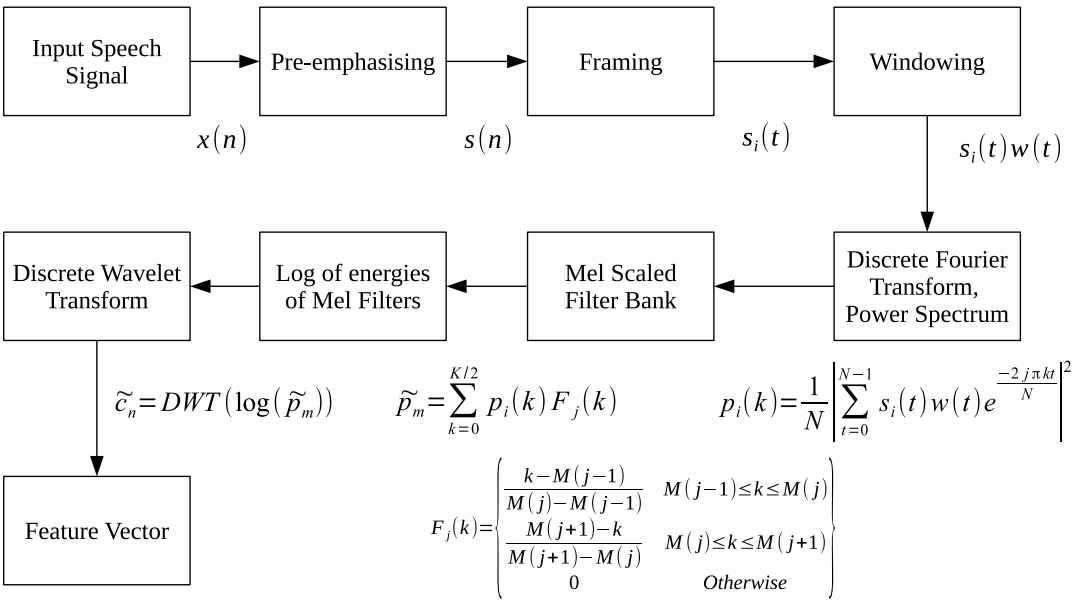

Figure 2.2: Steps of determining feature vectors.

- Frame the signal with overlapping frames
- Apply the Hamming window on each frame
- Obtain a discrete Fourier transform and power spectrum for each frame
- Find energy of applied Mel-scaled triangular filter-bank
- Apply Daubechies wavelets on logarithm of the filter–bank energies

These steps are summarized in the block diagram Figure 2.2.

2.3 Classification

2.3.1 Dynamic time warping (DTW)

DTW algorithm is useful to find the distance between the two sequences of unequal lengths. Let $X = \{x_1, x_2, \ldots, x_p\}$ and $Y = \{y_1, y_2, \ldots, y_q\}$ be two sequences of unequal lengths. The DTW distance between them is given by equation (2.18).

$$D(i,j) = |x_i - y_i| + K \tag{2.18}$$

In 2.18, $1 \le i \le p$, $1 \le j \le q$ and K is given by equation (2.19).

$$K = \min\{D(i, j-1), D(i-1, j-1), D(i-1, j)\} \tag{2.19}$$

The distance calculations are initialized with $D(1,1) = |x_1 - y_1|$. Here $|\cdot|$ is the Euclidean distance. Then other distances are determined using equations (2.18) and (2.19) iteratively.

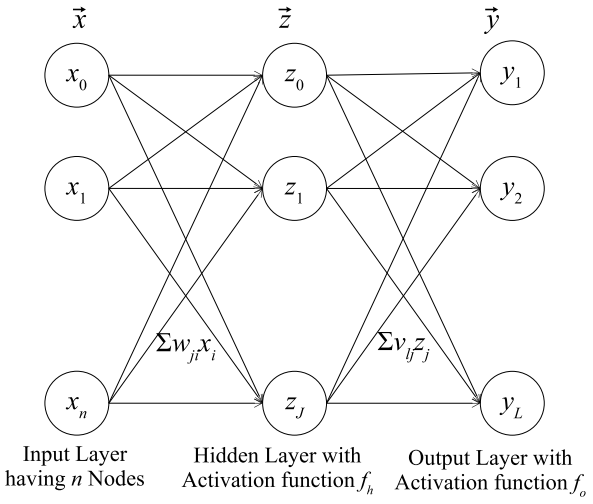

Figure 2.3: Multi-layer perceptron network architecture

2.3.2 Multilayered perceptrons (MLPs)

ANNs are useful for classification problems. MLPs are architecture of ANN. They learn generalisation from the patterns presented to it by changing weights. There are many algorithms for updating weights. Error Back–Propagation algorithm, introduced by [36], is useful for this purpose. The weights are changed in such a way that the error between the desired output and the output obtained from the network is minimised. The minimisation is achieved by taking gradients of errors with respect to the weights. To understand how it work, consider a two-layered MLP network architecture as shown in Figure 2.3. In this figure, let $\vec{x} = (x_0, x_1, \ldots, x_n) \in \mathbb{R}^{n+1}$ be the input vector having bias $x_0 = -1$ and input neurons $x_1, x_2, \ldots x_n$ and let $\vec{z} = (z_1, z_2, \ldots, z_J) \in \mathbb{R}^{J+1}$ be the hidden layer, having bias $z_0 = -1$ and hidden neurons $z_1, z_2, \ldots z_J$. Suppose we denote the connection between neurons x_i and z_j by w_{ji}. Let f_h be an activation function for a hidden layer. Then the values of the hidden neuron z_j for $0 \leqslant j \leqslant J$ is given by

$$z_j = f_h \left(\sum_{i=0}^{n} w_{ji} x_i \right) \tag{2.20}$$

Similarly, let $\vec{y} = (y_1, y_2, \ldots, y_L) \in \mathbb{R}^L$ be the output layer. Suppose we denote the weight connecting z_j and y_l by v_{lj}. Let f_o be an activation function for an output layer. Then the values of the output neuron y_l for $1 \leqslant l \leqslant L$ is given by

$$y_l = f_o \left(\sum_{j=0}^{J} v_{lj} z_j \right) \tag{2.21}$$

Let $\vec{d} = (d_1, d_2, \ldots, d_L) \in \mathbb{R}^L$ be the desired output. Then the sum of squares of all the errors by the current weights of the network are given by

$$E = \frac{1}{2} \sum_{l=1}^{L} (d_l - y_l)^2 \tag{2.22}$$

According to the Delta rule, we can update the weights by considering the partial derivative of error with respect to the weights. So for the output layer, the change in weights v_{lj} is given by

$$\begin{aligned}
\Delta v_{lj} &= \frac{\partial E}{\partial v_{lj}} \\
&= \frac{\partial}{\partial v_{lj}} \left[\frac{1}{2} \sum_{l=1}^{L} (d_l - y_l)^2 \right] \\
&= \frac{1}{2} \sum_{l=1}^{L} 2 (d_l - y_l)^{2-1} \frac{\partial}{\partial v_{lj}} (d_l - y_l) \\
&= \sum_{l=1}^{L} (d_l - y_l)^1 \frac{\partial}{\partial v_{lj}} \left[d_l - f_o \left(\sum_{j=0}^{J} v_{lj} z_j \right) \right] \\
&= \sum_{l=1}^{L} (d_l - y_l) \left[0 - f_o' \left(\sum_{j=0}^{J} v_{lj} z_j \right) z_j \right] \\
&= - \sum_{l=1}^{L} (d_l - y_l) f_o' \left(\sum_{j=0}^{J} v_{lj} z_j \right) z_j
\end{aligned} \tag{2.23}$$

Using Error Back-Propagation algorithm, the change in the weights w_{ji} between an input layer and a hidden layer is given by

$$\begin{aligned}
\Delta w_{ji} &= \frac{\partial E}{\partial w_{ji}} \\
&= \frac{\partial}{\partial w_{ji}} \left[\frac{1}{2} \sum_{l=1}^{L} (d_l - y_l)^2 \right] \\
&= \frac{1}{2} \sum_{l=1}^{L} 2 (d_l - y_l)^{2-1} \frac{\partial}{\partial w_{ji}} (d_l - y_l) \\
&= \sum_{l=1}^{L} (d_l - y_l)^1 \frac{\partial}{\partial w_{ji}} \left[d_l - f_o \left(\sum_{j=0}^{J} v_{lj} z_j \right) \right] \\
&= \sum_{l=1}^{L} (d_l - y_l) \left[0 - f_o' \left(\sum_{j=0}^{J} v_{lj} z_j \right) \frac{\partial}{\partial w_{ji}} \left(\sum_{j=0}^{J} v_{lj} z_j \right) \right] \\
&= - \sum_{l=1}^{L} (d_l - y_l) f_o' \left(\sum_{j=0}^{J} v_{lj} z_j \right) \sum_{j=0}^{J} v_{lj} \frac{\partial}{\partial w_{ji}} (z_j) \\
&= - \sum_{l=1}^{L} (d_l - y_l) f_o' \left(\sum_{j=0}^{J} v_{lj} z_j \right) \sum_{j=0}^{J} v_{lj} \frac{\partial}{\partial w_{ji}} \left[f_h \left(\sum_{i=0}^{n} w_{ji} x_i \right) \right] \\
&= - \sum_{l=1}^{L} (d_l - y_l) f_o' \left(\sum_{j=0}^{J} v_{lj} z_j \right) \sum_{j=0}^{J} v_{lj} f_h' \left(\sum_{i=0}^{n} w_{ji} x_i \right) x_i
\end{aligned} \tag{2.24}$$

Using equations (2.23) and (2.24), the updated weights at $(k+1)^{st}$ iteration are

$$v_{lj}^{k+1} = v_{lj}^{k} - \eta_o \Delta v_{lj}^{k} \tag{2.25}$$

$$w_{ji}^{k+1} = w_{ji}^{k} - \eta_h \Delta w_{ji}^{k} \tag{2.26}$$

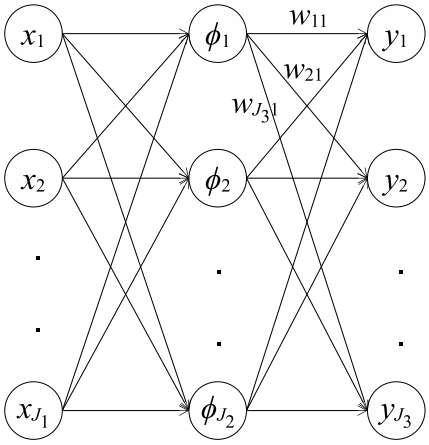

Figure 2.4: Radial basis function network (RBFN) architecture

In equations (2.25) and (2.26), η_h and η_o are learning rates usually having value between 0.0001 and 2. Several modifications of this algorithm occurred over the years and most recently the Adam algorithm is used to update weights, which is an adaptive momentum algorithm [37]. The momentum is updated iteratively and based on that, the weights are updated using equation (2.27) [38].

$$\Delta w = -\frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \tag{2.27}$$

Here,
$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ is updated momentum at step t,
$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla w_t$ is momentum at step t,
$\beta_1 = 0.9$,
$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$ is used to keep history of Gradients,
$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla w_t)^2$,
$\beta_2 = 0.999$,
$\eta = 0.001$ is a learning rate,
$\epsilon = 1 \times 10^{-8}$.

2.3.3 Radial basis function networks (RBFN)

The RBFN is shown in the Figure 2.4. It consists of an input layer, an output layer and one hidden layer consisting of RBFs $\phi_i(\vec{x}) = \phi(\vec{x} - \vec{c}_i)$, where $\vec{c}_i$ is a centre corresponding to the i^{th} neuron in a hidden layer [39]. In this architecture, the weights and the centres are updated for establishing relationship between inputs and outputs. Let $\{(\vec{x}_p, \vec{y}_p) | p = 1, 2, \ldots, N\}$, be given data where $\vec{x}_p$ contains inputs and $\vec{y}_p$ contains desired outputs. $W = [\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_{J_3}]$ is $J_2 \times J_3$ weight matrix with $\vec{w}_i = (w_{1i}, w_{2i}, \ldots, w_{J_2 i})^T$. For the RBF network with any suitable RBF, the error between desired output and output obtained by the network is given by

$$E = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni}^2 \tag{2.28}$$

Here e_{ni} is the error at i^{th} output neuron for the n^{th} pattern

$$e_{ni} = y_{ni} - \sum_{m=1}^{J_2} w_{mi} \phi(\|\vec{x}_n - \vec{c}_m\|) \tag{2.29}$$

Using the gradient descent method, the change in weights are obtained by

$$\begin{aligned}
\Delta w_{mi} &= \frac{\partial E}{\partial w_{mi}} \\
&= \frac{\partial}{\partial w_{mi}} \left[\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni}^2 \right] \\
&= \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} 2 e_{ni} \frac{\partial}{\partial w_{mi}} \left(e_{ni} \right) \\
&= \frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \frac{\partial}{\partial w_{mi}} \left(y_{ni} - \sum_{m=1}^{J_2} w_{mi} \phi \left(||\vec{x_n} - \vec{c_m}|| \right) \right) \\
&= \frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \left(0 - \sum_{m=1}^{J_2} \phi \left(||\vec{x_n} - \vec{c_m}|| \right) \right) \\
&= -\frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \sum_{m=1}^{J_2} \phi \left(||\vec{x_n} - \vec{c_m}|| \right)
\end{aligned} \tag{2.30}$$

Similarly, the change in centres is given by

$$\begin{aligned}
\Delta \vec{c_m} &= \frac{\partial E}{\partial \vec{c_m}} \\
&= \frac{\partial}{\partial \vec{c_m}} \left[\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni}^2 \right] \\
&= \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} 2 e_{ni} \frac{\partial}{\partial \vec{c_m}} \left(e_{ni} \right) \\
&= \frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \frac{\partial}{\partial \vec{c_m}} \left(y_{ni} - \sum_{m=1}^{J_2} w_{mi} \phi \left(\|\vec{x}_n - \vec{c}_m\| \right) \right) \\
&= \frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \left(0 - \sum_{m=1}^{J_2} w_{mi} \frac{\partial}{\partial \vec{c_m}} \phi \left(\|\vec{x}_n - \vec{c}_m\| \right) \right) \\
&= -\frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \sum_{m=1}^{J_2} w_{mi} \phi' \left(\|\vec{x}_n - \vec{c}_m\| \right) \frac{\partial}{\partial \vec{c_m}} \|\vec{x}_n - \vec{c}_m\| \\
&= -\frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \sum_{m=1}^{J_2} w_{mi} \phi' \left(\|\vec{x}_n - \vec{c}_m\| \right) \frac{\vec{x}_n - \vec{c}_m}{||\vec{x}_n - \vec{c}_m||} (-1) \\
&= \frac{2}{N} \sum_{n=1}^{N} \sum_{i=1}^{J_3} e_{ni} \sum_{m=1}^{J_2} w_{mi} \phi' \left(\|\vec{x}_n - \vec{c}_m\| \right) \frac{\vec{x}_n - \vec{c}_m}{||\vec{x}_n - \vec{c}_m||}
\end{aligned} \tag{2.31}$$

Using equations (2.30) and (2.31), the updated weights and centres at $(k+1)^{st}$ iteration are

$$w_{mi}^{k+1} = w_{mi}^{k} - \eta_1 \Delta w_{mi}^{k} \tag{2.32}$$

$$\vec{c_m}^{\,k+1} = \vec{c_m}^{\,k} - \eta_2 \Delta \vec{c_m}^{\,k} \tag{2.33}$$

In equations (2.32) and (2.33), η_1 and η_2 are learning rates usually having value between 0.0001 and 2. Such trained RBFN has capabilities of universal approximation. By selection of suitable RBF, we can approximate any continuous function with RBFN [40].

2.3.4 Hidden Markov models (HMM)

HMM are used to model problems having hidden states produced by observations. The model consists of N hidden states $S_1, S_2, \ldots, S_N$ and T observations $O_1, O_2, \ldots, O_T$. The parameters of HMM are initial state probabilities $\pi_i = P(S_i \text{ at time } t = 1)$, state transition probabilities $a_{ij} = P(S_j \text{ at time t+1} | S_i \text{ at time } t)$ and observation probabilities $b_j(O_k) = P(O_k \text{ at time t } | S_j \text{ at time } t+1)$. The initial state probabilities and the state transition probabilities are initialized randomly. But the observation probabilities are determined using Gaussian mixture models (GMM) [41].

Gaussian mixture models (GMM)

GMM uses a linear combination of multivariate Gaussian distributions to determine the observation probabilities $b_j(O_k)$ using equation 2.34 [42].

$$b_j(O_k) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sqrt{2\pi|\Sigma_{jm}|}} e^{-\frac{1}{2}(O_k - \mu_{jm})^T \Sigma_{jm}^{-1} (O_k - \mu_{jm})} \tag{2.34}$$

Here Σ_{jm} is the covariance matrix and μ_{jm} is the mean of m^{th} Gaussian probability density function for the j^{th} state. C_{jm}'s are the coefficients.

Baum-Welch algorithm

After initializing the parameters of HMM, they are trained iteratively using the Baum-Welch algorithm [43]. Let $\xi_t(i,j)$ be the joint probability of being in state S_i at time t and state S_j at time $t+1$ and $\gamma_t(i)$ be the probability of being in state S_i at time t. The algorithm is given by following equations.

$$\pi^\star = \gamma_1(i) \tag{2.35}$$

$$a_{ij}^\star = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{2.36}$$

$$b_j^\star(O_k) = \frac{\sum_{t=1 \text{ s.t. } O_t = S_K}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \tag{2.37}$$

The HMM with trained probabilities are then used for testing of unknown observations.

Chapter 3

Speech recognition of Isolated Gujarati words

For these experiments, isolated Gujarati words were recorded and then features were extracted from them using MFCC or MFDWC. Then the features were classified using four different approaches: Dynamic Time Warping, Artificial Neural Networks, Radial Basis Function Networks and Hidden Markov Models. The details of these experiments are summarized in the subsequent sections.

3.1 MFCC

3.1.1 Dynamic Time Warping

The recording of digits 1-10 was done using Audacity software. The recording sampling rate was 16000 Hz with Mono channel. For each speaker, digits 1-10 were extracted using zero-crossing. Then features were extracted from all speech signals. Features of all the digits by the template speaker was compared with that of the other speakers. The results for one such pair is shown in Figure 3.1. In Figure 3.1, each cell is the global DTW distance, found using DTW algorithm, between the features of a template speaker and features of other speaker. The column under title '1' represents the comparison of digit '1' by one speaker with the all digits of the template speaker. The highlighted cells represent the least DTW distance in each column. Figure 3.1 concludes that 8 out of 10 digits by the selected speaker are matching with the template. This process was repeated with all speakers keeping one person as a template. Similar tables were prepared for comparison of the spoken digits of template voice with all other speakers. Overall 84.44 % recognition success was achieved [44].

		SPEAKER									
TEMPLATE	Digits	1	2	3	4	5	6	7	8	9	10
	1	**2200**	4391	3083	4411	3082	3674	4453	3921	4293	2901
	2	3984	**2191**	2891	4489	3006	4025	3818	3538	2994	2453
	3	2645	4457	**1503**	5172	4044	3553	4551	4292	3321	3926
	4	6399	6857	3869	**3714**	4267	4294	**3128**	4301	6524	6391
	5	3356	4559	3084	3798	**2178**	3874	4227	**3065**	4185	2481
	6	3775	7193	3313	5305	5903	**3082**	5146	5925	4585	6579
	7	6932	6618	4736	4171	5277	5250	3520	4881	7302	7493
	8	4450	5364	2153	4908	4053	4111	4300	3313	4134	4038
	9	2900	5234	2748	7225	5942	4995	5975	5440	**2302**	4630
	10	3338	3100	3629	4838	2774	4569	4867	3650	4054	**1402**

Figure 3.1: DTW distances between two speakers using MFCC

3.1.2 Artificial Neural Networks

In this experiment [45], same dataset was used as that of the DTW experiment. After recording speech and pre-processing, the dataset was in form of vectors representing speech signals. Then MFCC features were extracted from them. The unequal lengths of these features were made equal using the insertion algorithm. The features obtained this way, were used as inputs to the Artificial Neural Network having 603 input neurons, 50 hidden neurons and 10 output neurons representing classes for each digit. The dataset was divided in training and testing with ratio 80:20. ANN was trained using Error Back-propagation rule. The output for the recognized digit was the image of corresponding digit in Gujarati language as well as its equivalent Sign language form as shown in the Figure 3.2. 100% train accuracy and 74% test accuracy were obtained for this experiment [45].

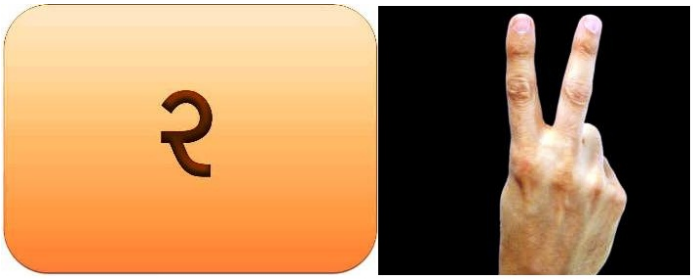

Figure 3.2: Output of the recognized digit 2

3.1.3 Radial Basis Function Networks

In this experiment [46], Radial Basis function (RBF) network was used for the classification of the equal-length MFCC feature vectors. The feature vectors were normalized before using them in the RBF network. The RBF network was constructed having 603 input neurons, 100 hidden neurons and 4 output neurons. The output neurons are binary representation of digits to be recognized. The output for the recognized digit is the image of corresponding digit in Gujarati language as well as its equivalent Sign language form as shown in the Figure 3.2. 100% train accuracy and 92% test accuracy were obtained for the first dataset [46].

3.2 MFDWC

3.2.1 Dynamic Time Warping

These experiments are similar to section 3.1.1. The main difference is that in this case, MFDWC features were used for the feature extraction of isolated digits 1-10. Here also, similar analysis was done and overall 86% accuracy was achieved [47]. There is a slight improvement in the accuracy due to involvement of wavelets in the feature extraction technique.

3.2.2 Artificial Neural Networks

In this experiment, two datasets were used. One dataset having recordings of digits 1-10 spoken by eight speakers. The other dataset was expanded dataset using the data augmentation methods. After recording speech and pre-processing, the dataset was in form of vectors representing speech signals. Then MFDWC features were extracted from them. The unequal lengths of these features were made equal using Cubic spline interpolation. The MFDWC features obtained this way, were used as inputs to the Artificial Neural Network having 603 input neurons, 78 hidden neurons and 10 output neurons representing classes for each digit. The dataset was divided in training and testing with stratified random sampling method. ANN was trained using Adam algorithm. The accuracy of a trained network was calculated using the confusion matrix as shown in the Figure 3.3. The network was trained several times for different wavelets.

Average accuracy, train time and number of iterations for various wavelets are summarised in Table 3.1. On an average 92% accuracy was achieved for Daubechies–6 wavelet, which is significantly better as compared to the MFCC approach. It took on an average 5.14 seconds and 168 iterations to achieve this accuracy.

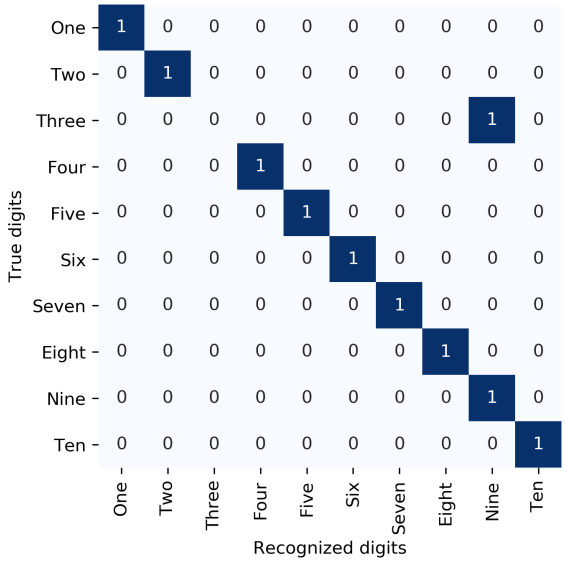

Figure 3.3: Confusion matrix

Name of wavelet	Average accuracy (%)	Average train time (second)	Average number of iterations
Daubechies – 1	47.50	2.95	100
Daubechies – 2	53.50	3.69	122
Daubechies – 3	58.00	3.60	121
Daubechies – 4	78.50	4.35	150
Daubechies – 5	85.00	4.74	158
Daubechies – 6	92.00	5.14	168

Table 3.1: Summary of results obtained for different wavelets

The second dataset was an expanded version of the first dataset. It was expanded using the data augmentation methods like amplifying the signal by some factor, de-amplifying the signal by some factor, adding a noise in the signal, shrinking the signal in time and stretching the signal in time. These variations were performed on all raw files and the dataset of total 1200 samples was created, 120 samples of each digit one to ten. Then MFDWC features were extracted from all of them. Then the dataset was divided into train set and test set. Train set consisted of 1000 samples and the remaining 200 samples were part of the test set. The Daubechies-6 wavelet was used for wavelet decomposition. Then the Artificial Neural Network consisting of three hidden layers of 80, 80 and 40 neurons each was trained using the Adam algorithm and ReLU activation function. It took 1625 iterations and 161 seconds to achieve 0.04 loss. Further, this network was tested with the remaining 200 patterns. The confusion matrix for the classification results as shown in Figure 3.4, suggests the 85% testing accuracy.

3.2.3 Radial Basis Function Networks

This experiment is similar to section 3.1.3. But the difference is that here features were extracted from the speech using MFDWC technique. The input layer consists of 603 neurons corresponding to

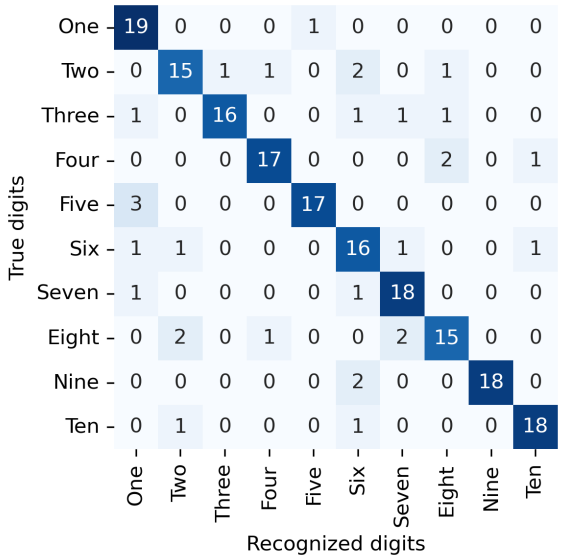

Figure 3.4: Confusion matrix for the augmented dataset

all MFDWC features shrunk or stretched to its median length of 603. The hidden layer consists of 70 neurons corresponding the Gaussian RBF. Output layer consists 4 neurons corresponding to the binary representation of digits 1–10 to be recognized. The data was normalised using maximum norm before training the network. Accuracy 90% was achieved for Daubechies–6 wavelet. It took on an average 0.54 second to achieve this accuracy.

3.2.4 Hidden Markov Models

In this experiment, for the recognition part, the HMM with GMM was used. The left-to-right HMMs were created for each word. In these HMMs, the hidden states were the speech units hidden inside the recording and the observations were the MFDWC features. Different number of states from 2 to 7 were considered and the accuracy was determined in each case. The HMM parameters π_i and a_{ij} were initialized randomly. The parameter $b_j(o_k)$ was initiated by determining the probability distribution of MFDWC features using GMM. In this, the mean and the variance of each feature vector for each dimension was determined. The dataset was divided into training and testing using stratified sampling method with 12.5% test patterns. HMM parameters were trained using Baum-Welch algorithm. During testing, given an unknown word, these trained parameters were used to determine the HMM of the unknown word and this way, the spoken word can be identified. Two types of datasets were used in this. The original dataset having digits 1-10 for 8 speakers and the augmented dataset of 160 samples. The results obtained for both datasets are summarized in Table 3.2. The Confusion matrices for the original and the augmented datasets are shown in figures 3.5 and 3.6 respectively.

	Original Dataset		Augmented Dataset	
Number of States in HMM	Training Time (sec)	Train Accuracy (%)	Training Time (sec)	Test Accuracy (%)
2	5.17	100	14.14	60
3	5.38	100	14.93	65
4	5.62	100	16.33	65
5	5.76	100	16.15	70
6	6.19	100	17.14	65
7	6.33	100	17.78	65

Table 3.2: Summary of results of experiments based on HMM

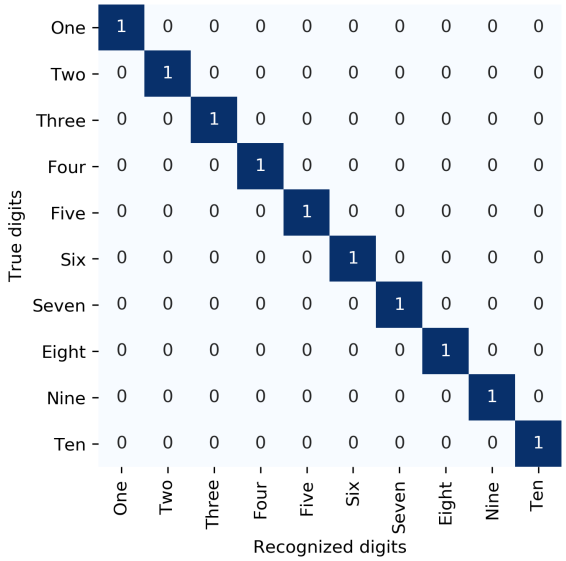

Figure 3.5: Confusion matrix for the original dataset

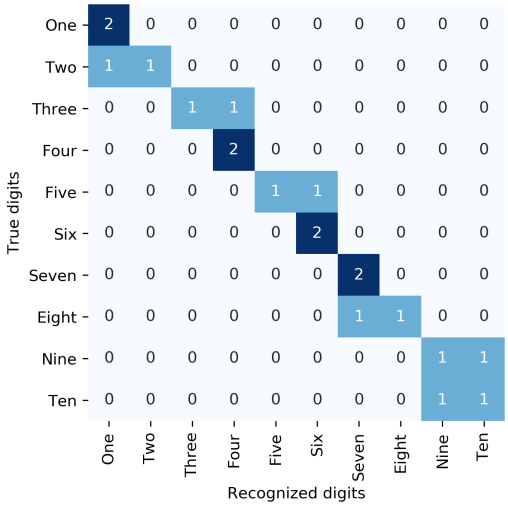

Figure 3.6: Confusion matrix for the Augmented dataset

Chapter 4

Speech recognition of continuous Gujarati sentences

Following the work of SR for Gujarati and isolated words, we focused for ASR for continuous speech. For this purpose, the sentences are to be identified and broken into words which are further processed. To isolate the words from sentences, we proposed a method based on STAC algorithm. The words once isolated are processed to get MFDWC. Further, the MFDWC features of these words were classified using two machine learning techniques: ANN and HMM. The efficiency of ANN model is measured using various metrics such as accuracy, recall and F-score etc. and compared.

4.1 Using Artificial Neural Networks:

In this experiment [48], a speech of 6 speakers was recorded. The speech consists of 5 sentences spoken in Gujarati language, as shown in the Figure 4.1. Recording was done using microphone of usual headphone in normal room noise using Audacity software with sampling rate of 16,000 samples per second and mono channel. Two recordings of each speaker were considered for training. Then, words were extracted from sentences using STAC. Further, MFDWC features were extracted from each word. The lengths of all the feature vectors were made equal using the Cubic Spline Interpolation. These features were trained in the ANN having 676 neurons in the input layer, 1000 neurons in the hidden layer and 1 neuron in the output layer for classification of words. Various wavelets were used in feature extraction step. Wavelet decomposition at level 2 for wavelet Daubechies–6 gave us best results. The networks were trained using back-propagation algorithm with sigmoid activation function and Adam algorithm with ReLU activation function. The results obtained for both networks are summarised in Table 4.1. The trained networks were tested with unknown speech consisting of sentences different from the sentences trained. After being successfully recognized, our system gives sentence output in Gujarati language text.

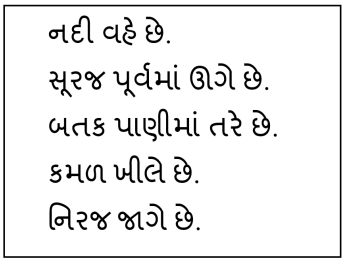

Figure 4.1: Dataset for the continuous SR using ANN

Algorithm	Activation function	Number of Iterations	Training time (sec)	Training loss	Testing accuracy
BP	Sigmoid	956	134	0.00742	84.62 %
Adam	ReLU	1098	29	0.00042	76.92 %

Table 4.1: Summary of experiments of Continuos SR using ANN

ભારત દેશ માં ગુજરાત રાજ્ય આવેલ છે.
ગુજરાત રાજ્ય માં ગુજરાતી ભાષા બોલાય છે.
ગુજરાત રાજ્ય નું પાટનગર ગાંધીનગર છે.
ગુજરાત રાજ્ય નું પાટનગર દિલ્લી નથી.
વડોદરા ગુજરાત રાજ્ય માં આવેલ છે.

Figure 4.2: Dataset for continuous SR using HMM

4.2 Using Hidden Markov Models:

In this experiment, a dataset of 10 recordings having 5 sentences and total 32 words was considered. These sentences are shown in Figure 4.2. There were 16 unique words. These $10 \times 32 = 320$ words were extracted from sentences using the STAC algorithm. The lengths of words were made equal using the Cubic spline interpolation methods. Then MFDWC features were determined for each word. 3-State HMM was built for each unique word. The HMM parameters π_i and a_{ij} were initialized randomly and $b_j(o_k)$ was initiated using GMM. The dataset was divided into training and testing using stratified sampling method. Out of 320 words, 288 were used to train the HMM using the Baum-Welch algorithm. Rest of the 32 words were used for testing. The network was trained in 19.47 seconds. Overall 84.38% test accuracy was obtained. The confusion matrix is as shown in Figure 4.3. Various accuracy measures are as shown in the Table 4.2.

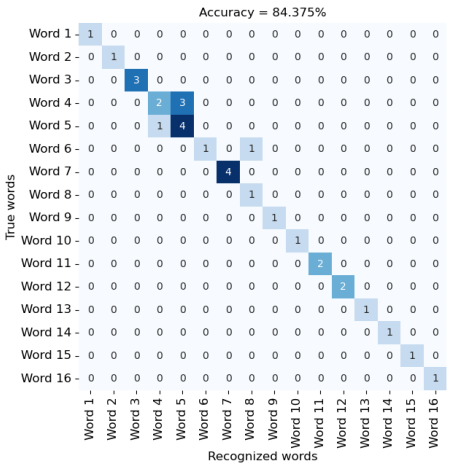

Figure 4.3: Confusion matrix for continuous SR using HMM

We can observe that different accuracy was obtained for both models discussed here. So, to improve accuracy of speech recognition model, we have employed the ensemble techniques like bagging and boosting. For boosting we have used ANN models. In bagging ensemble model we have used combination of HMM and ANN models.

Words	Test Patterns	Precision	Recall	f1-score
Word 1	1	1.00	1.00	1.00
Word 2	1	1.00	1.00	1.00
Word 3	3	1.00	1.00	1.00
Word 4	5	0.67	0.40	0.50
Word 5	5	0.57	0.80	0.67
Word 6	2	1.00	0.50	0.67
Word 7	4	1.00	1.00	1.00
Word 8	1	0.50	1.00	0.67
Word 9	1	1.00	1.00	1.00
Word 10	1	1.00	1.00	1.00
Word 11	2	1.00	1.00	1.00
Word 12	2	1.00	1.00	1.00
Word 13	1	1.00	1.00	1.00
Word 14	1	1.00	1.00	1.00
Word 15	1	1.00	1.00	1.00
Word 16	1	1.00	1.00	1.00
	32			

Table 4.2: Precision, recall and f1-score for Continuous SR using HMM

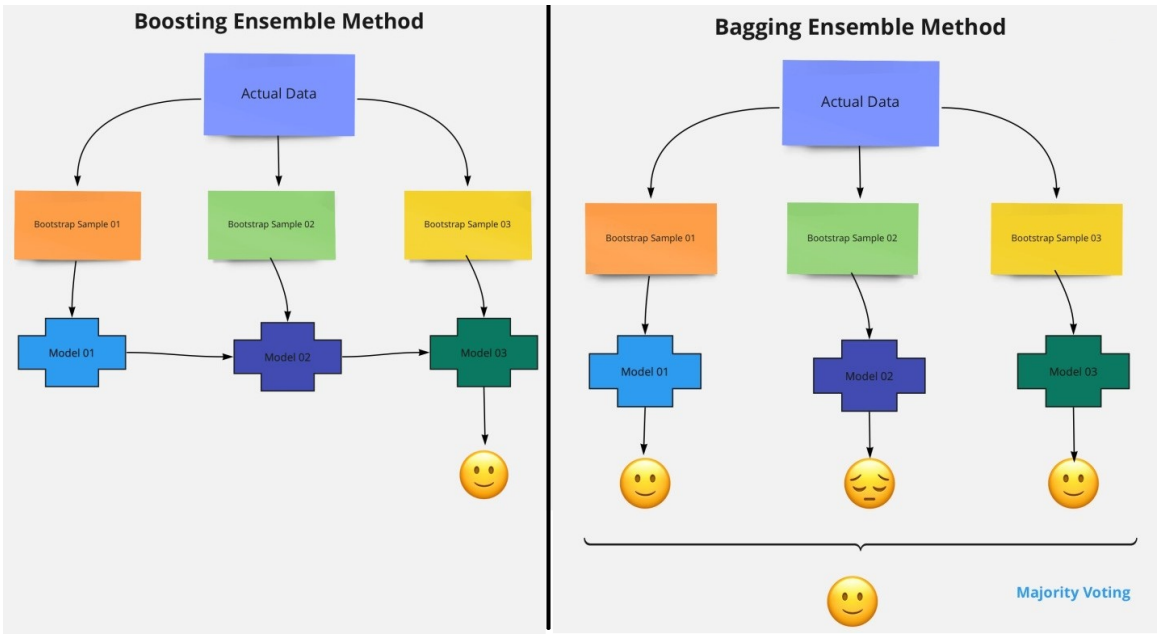

Figure 4.4: Ensemble learning methods

Chapter 5

Gujarati Speech Recognizer Interface

This experiment is mainly based on the development of the Gujarati Speech Recognizer Interface as shown in the Figure 5.1. This interface is able to record a speech and give the text output of the spoken speech. The buttons for performing various tasks are available on interface like reading files, extracting words, determining MFDWC features, Building and training HMM, plotting confusion matrix and recognize the speech. The dataset for this experiment was same as that of previous experiment. In this, a new sentence having 8 words was made using the same 16 words of the previous experiment. The HMM was tested using this sentence. Out of 8 words of this sentence, 7 were matching, giving the accuracy of 87.5% as shown in the Figure 5.1. The updated version of the software contains less number of buttons as shown in the figure 5.2. This kind of interface is useful for the layman, making it possible for diverse applications.

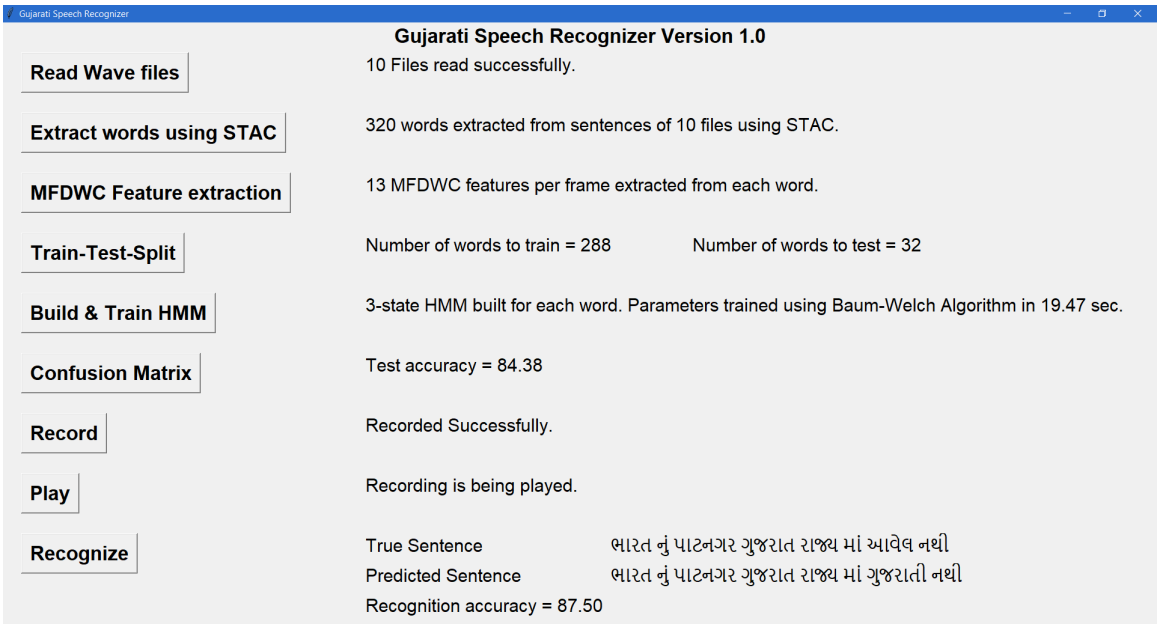

Figure 5.1: Speech recognizer interface

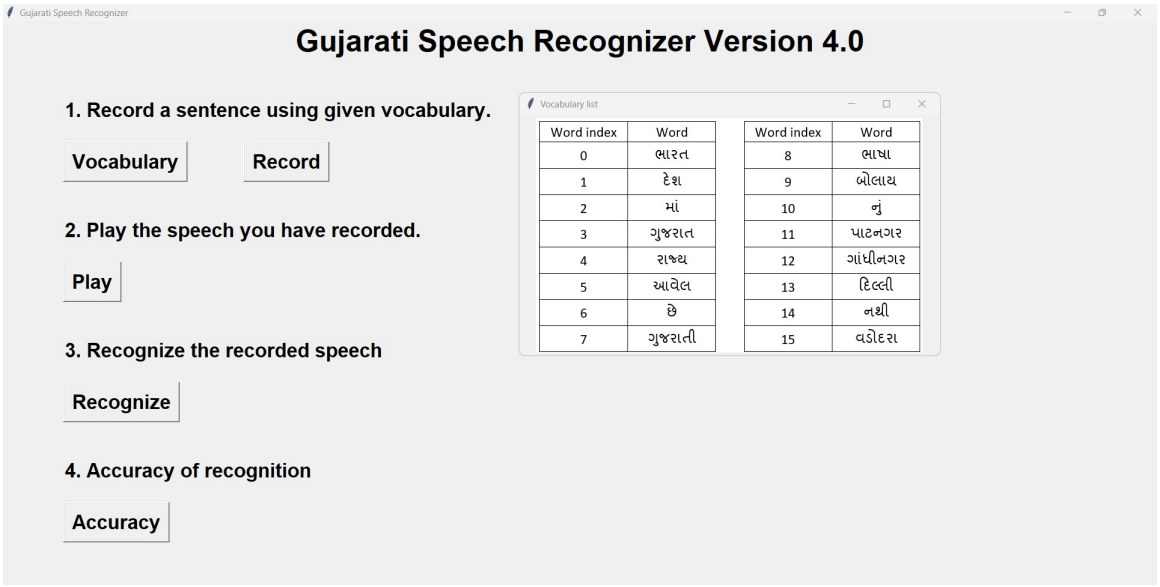

Figure 5.2: Updated Speech recognizer interface

Chapter 6

Conclusions

The research work focuses on various techniques for Automatic Speech Recognition. For this, Gujarati language was considered. For feature extraction, Mel-frequency Cepstral Coefficients and wavelet-based technique Mel-frequency Discrete Wavelet Coefficients were used. For the recognition, various machine learning techniques like Artificial Neural Networks, Radial Basis Function Networks and Hidden Markov Models were used. Two types of experiments were performed: Automatic Speech Recognition for isolated words and Automatic Speech Recognition for continuous sentences. For the experiments of continuous sentences, words were extracted using signal processing technique Short-Term Auto Correlation. The recognition accuracy obtained in these experiments are summarized in the Table 6.1. Moreover, an interface of Gujarati Speech recognizer was developed, having ability to record, play and recognize a speech spoken in the Gujarati language.

Accuracy obtained (in %) for various methods of Speech Recognition in Gujarati language			
Methods	Isolated Word Recognition		Continuous Speech Recognition
	MFCC	MFDWC	MFDWC
DTW	84.44	86.00	-
ANN	74.00	92.00	86.62 (Back Propagation) 76.92 (Adam)
RBFN	92.00	90.00	-
HMM	-	100 (Original data) 70 (Augmented data)	84.38

Table 6.1: Summary of all the experiments

From these experiments, it can be observed that for the feature extraction technique, MFDWC method is better as compared to the MFCC method. Overall we can say that as compared to HMM, ANN or RBF gives more accuracy.

Chapter 7

Bibliography

[1] L. R. Rabiner, B.-H. Juang, and B. Yegnanarayana, *Fundamentals of Speech Recognition*, 2nd ed. Pearson Education, Dorling Kindersley (India) Pvt. Ltd., New Delhi, India., 2010.

[2] B. H. Juang and L. R. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology," pp. 1–24, 2005.

[3] G. Hemantkumar and P. Punitha, "Speech Recognition Technology: A Survey on Indian Languages," *International Journal of Information Science and Intelligent System*, vol. 2, no. 4, pp. 1–38, 2013.

[4] G. Cardona and D. Jain, *The Indo-Aryan Languages*, 1st ed. Routledge, Abingdon, Oxon, United Kingdom, 2007.

[5] C. Kurian, "A Survey on Speech Recognition in Indian Languages," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 5, pp. 6169–6175, 2014.

[6] J. H. Tailor and D. B. Shah, "Review on Speech Recognition System for Indian Languages," *International Journal of Computer Applications*, vol. 119, no. 2, pp. 15–18, 2015.

[7] A. Singh, V. Kadyan, M. Kumar, and N. Bassan, "ASRoIL: a comprehensive survey for automatic speech recognition of Indian languages," *Artificial Intelligence Review*, vol. 53, no. 5, pp. 3673–3704, 2020. [Online]. Available: https://doi.org/10.1007/s10462-019-09775-8

[8] R. B. Parikh and H. Joshi, "Gujarati Speech Recognition – A Review," *Test Engineering and Management*, vol. 83, no. 4, pp. 549–553, 2020.

[9] K. D. Malde, B. B. Vachhani, M. C. Madhavi, N. H. Chhayani, and H. A. Patil, "Development of speech corpora in Gujarati and Marathi for phonetic transcription," in *International Conference Oriental COCOSDA Held Jointly with 2013 Conference on Asian Spoken Language Research and Evaluation, O-COCOSDA/CASLRE 2013*, 2013, pp. 1–6.

[10] M. C. Madhavi, S. Sharma, and H. A. Patil, "Development of language resources for speech application in Gujarati and Marathi," in *International Conference on Asian Language Processing (IALP)*, 2014, pp. 115–118.

[11] V. P. Tank and S. K. Hadia, "Creation of speech corpus for emotion analysis in Gujarati language and its evaluation by various speech parameters," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4752–4758, 2020.

[12] H. B. Chauhan and B. A. Tanawala, "Comparative Study of MFCC And LPC Algorithms for Gujrati Isolated Word Recognition," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 2, pp. 822–826, 2015.

[13] J. Patel and A. Nandurbarkar, "Development and Implementation of Algorithm for Speaker recognition for Gujarati Language," *International Research Journal of Engineering and Technology*, vol. 2, no. 2, pp. 444–448, 2015.

[14] P. Pravin and H. Jethva, "Neural Network Based Gujarati Language Speech Recognition," *International Journal of Computer Science and Management Research*, vol. 2, no. 5, pp. 2623–2627, 2013.

[15] V. A. Desai and V. K. Thakar, "Neural Network Based Gujarati Speech Recognition for Dataset Collected by in-ear Microphone," *Procedia Computer Science*, vol. 93, no. 2016, pp. 668–675, 2016.

[16] A. Chittora and H. A. Patil, "Classification of phonemes using modulation spectrogram based features for Gujarati language," in *International Conference on Asian Language Processing (IALP)*, 2014, pp. 46–49.

[17] J. H. Tailor and D. B. Shah, "Speech Recognition System Architecture for Gujarati Language," *International Journal of Computer Applications*, vol. 138, no. 12, pp. 28–31, 2016.

[18] J. Tailor and D. Shah, "HMM-Based Lightweight Speech Recognition System for Gujarati Language," *Lecture Notes in Networks and Systems*, vol. 10, pp. 451–461, 2018.

[19] S. Valaki and H. Jethva, "A hybrid HMM/ANN approach for automatic Gujarati speech recognition," in *International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017, pp. 1–5.

[20] H. N. Patel and P. V. Virparia, "A Small Vocabulary Speech Recognition for Gujarati," *International Journal of Advanced Research in Computer Science*, vol. 2, no. 1, pp. 208–210, 2011. [Online]. Available: http://www.ijarcs.info/index.php/Ijarcs/article/viewFile/272/262

[21] S. Sharma, M. C. Madhavi, and H. A. Patil, "Development of vocal tract length normalized phonetic engine for Gujarati and Marathi languages," in *17th Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA)*, 2014, pp. 1–6.

[22] V. Desai and V. Thakar, "Word boundary detection for Gujarati speech recognition using in-ear microphone," *1st India International Conference on Information Processing (IICIP)*, pp. 1–6, 2016.

[23] N. Patel, S. Agarwal, N. Rajput, A. Nanavati, P. Dave, and T. S. Parikh, "A comparative study of speech and dialed input voice interfaces in rural India," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2009, pp. 51–54.

[24] J. K. Patel, P. N. Patel, and P. V. Virparia, "Gujarati Language Speech Recognition System for Identifying Smartphone Operation Commands," *National Journal of System and Information Technology*, vol. 8, no. 2, pp. 79–88, 2015.

[25] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Multilingual Speech Recognition with a Single End-to-End Model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4904–4908.

[26] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-scale multilingual speech recognition with a streaming end-to-end model," in *Proc. Interspeech 2019*, 2019, pp. 2130–2134.

[27] A. Diwan and P. Jyothi, "Reduce and Reconstruct: Improving Low-resource End-to-end ASR Via Reconstruction Using Reduced Vocabularies," *ArXiv*, 2020. [Online]. Available: http://arxiv.org/abs/2010.09322

[28] B. M. L. Srivastava, S. Sitaram, R. Kumar Mehta, K. Doss Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. Nayak, "Low Resource Automatic Speech Recognition Challenge for Indian Languages," in *Proc. 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, 2018, pp. 11–14.

[29] J. Billa, "ISI ASR System for the Low Resource Speech Recognition Challenge for Indian Languages," in *Proc. Interspeech 2018*, 2018, pp. 3207–3211.

[30] N. Fathima, T. Patel, C. Mahima, and A. Iyengar, "TDNN-based multilingual speech recognition system for low resource Indian languages," in *Proc. Interspeech 2018*, 2018, pp. 3197–3201.

[31] H. B. Sailor and T. Hain, "Multilingual speech recognition using language-specific phoneme recognition as auxiliary task for indian languages," in *Proc. Interspeech 2020*, 2020, pp. 4756–4760.

[32] P. Prandoni and M. Vetterli, *Signal processing for communications*. EPFL Press, 2008, vol. 1.

[33] M. Vetterli, J. Kovacevic, and V. Goyal, *Foundations of Signal Processing*. Cambridge University Press, 2013. [Online]. Available: http://www.amazon.co.uk/Foundations-Signal-Processing-Martin-Vetterli/dp/110703860X

[34] Z. Tufekci, J. Gowdy, S. Gurbuz, and E. Patterson, "Applied mel-frequency discrete wavelet coefficients and parallel model compensation for noise-robust speech recognition," *Speech Communication*, vol. 48, no. 10, pp. 1294–1307, 2006.

[35] C.-L. Liu, "A Tutorial of the Wavelet Transform," Tech. Rep., 2010.

[36] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[37] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, 2015, pp. 1–15.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, 1st ed. The MIT Press, Cambridge, Massachusetts, USA, 2016. [Online]. Available: http://deeplearning.net/

[39] J. Park and I. W. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.

[40] Y. Wu, H. Wang, B. Zhang, and K.-L. Du, "Using Radial Basis Function Networks for Function Approximation and Classification," *ISRN Applied Mathematics*, pp. 1–34, 2012.

[41] L. R. Rabiner and B. H. Juang, "An Introduction to Hidden Markov Models," *IEEE ASAP Megazine*, pp. 4–16, jan 1986.

[42] D. Jurafsky and J. H. Martin, *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. Prentice Hall, 2008.

[43] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech RecognitionNo Title," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[44] P. Pandit and S. Bhatt, "Automatic Speech Recognition of Gujarati digits using Dynamic Time Warping," *IJEIT*, vol. 3, no. 12, pp. 69–73, 2014.

[45] P. Pandit, S. Bhatt, and P. Makwana, "Automatic Speech Recognition of Gujarati Digits using Artificial Neural Network," in *Proceedings of 19th Annual Cum 4th International Conference of GAMS On Advances in Mathematical Modelling to Real World Problems*. Excellent Publishers, 2014, pp. 141–146.

[46] P. Pandit and S. Bhatt, "Automatic Speech Recognition of Gujarati digits using Radial Basis Function Network," *International Conference on Futuristic Trends in Engineering, Science, Pharmacy and Management*, pp. 216–226, 2016.

[47] P. K. Pandit and S. Bhatt, "Automatic Speech Recognition of Gujarati digits using Wavelet Coefficients," *Journal of The M. S. University of Baroda*, vol. 52, no. 1, pp. 101–110, 2017.

[48] P. Pandit, P. Makwana, and S. Bhatt, "Automatic Speech Recognition of Continuous Speech Signal of Gujarati Language Using Machine Learning," in *AISC*. Springer, Singapore, 2021, pp. 147–159.