Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-10073-7

Automatic speech recognition: a survey

Mishaim Malik[1] • Muhammad Kamran Malik[2] • Khawar Mehmood[3] •
Imran Makhdoom[4]

Received: 31 May 2020 / Revised: 4 September 2020 / Accepted: 13 October 2020
Published online: 10 November 2020

Abstract

Recently great strides have been made in the field of automatic speech recognition (ASR) by using various deep learning techniques. In this study, we present a thorough comparison between cutting-edged techniques currently being used in this area, with a special focus on the various deep learning methods. This study explores different feature extraction methods, state-of-the-art classification models, and vis-a-vis their impact on an ASR. As deep learning techniques are very data-dependent different speech datasets that are available online are also discussed in detail. In the end, the various online toolkits, resources, and language models that can be helpful in the formulation of an ASR are also proffered. In this study, we captured every aspect that can impact the performance of an ASR. Hence, we speculate that this work is a good starting point for academics interested in ASR research.

Keywords Speech recognition · ASR · Automatic speech recognition · Feature extraction · Classification models · Language models

✉ Mishaim Malik
mishaimmalik30@gmail.com

Muhammad Kamran Malik
kamran.malik@pucit.edu.pk

Khawar Mehmood
k.mehmood@unsw.edu.au

Imran Makhdoom
imran.makhdoom@uts.edu.au

[1] Punjab University College of Information Technology (PUCIT), Lahore, Pakistan

[2] Faculty of Punjab University College of Information Technology (PUCIT), Lahore, Pakistan

[3] School of Engineering and Information Technology, University of New South Wales (UNSW) Canberra at ADFA, Canberra, Australia

[4] Faculty of Engineering and IT, University of Technology Sydney, Ultimo, Australia

1 Introduction

Speech is the most natural, efficient and preferred mode of communication between humans. Therefore it can be assumed that people are more comfortable using speech as a mode of input for various machines rather than such other primitive modes of communication as keypads and keyboards. Automatic speech recognition (ASR) system helps us achieve this goal. Such a system allows a computer to take the audio file or direct speech from the microphone as an input and convert it into the text; preferably in the script of the spoken language. An ideal ASR should be able to "perceive" the given input, "recognize" the spoken words and then subsequently use the recognized words as an input to another machine so that some "action" can be performed on it [42, 126, 160]. Retrospectively, we consider ASRs to be the future means of communication between humans and machines.

Human speech and accents have huge variations, and this variation in speech patterns is one of the biggest obstacles in creating an autonomous speech recognition system. Bilingual or multilingual people tend to show more of these variations in their speech patterns than people who speak only one language. The same problem also arises when we add different factors such as gender, social style/dialect, speaking style and speed into the equation [40, 112]. Another obstacle to creating an ASR is finding enough resources to train the ASR model. Currently, such training models are available only for a handful of languages out of a total of approximately 6500 world languages.

Over the past few years, many survey papers have been published to review and examine various aspects of ASR models presented over time. A recently published survey paper [160] discussed the challenges an ASR will have to overcome; and also discussed and analyzed the well-known models of ASR. It analyzes various challenges which include utterance approach and style, different speaker models, vocabulary size, and channel variability. The paper also highlighted three different classification approaches; acoustic-phonetic approach, pattern recognition approach, and artificial intelligence approach. In another work [144], authors reviewed the efficiency of different feature extraction techniques including perceptual linear prediction (PLP), revised perceptual linear prediction (RPLP), and Bark frequency cepstral coefficients (BFCC). The paper compared the results of all these feature extraction techniques on different classification models. [97] also presented some challenges to the real-world implementation of an optimal ASR system. The authors also classified ASR on the basis of speaker mode, speaking mode, and vocabulary size. The paper elaborates the front-end and back-end of an ASR system. The front-end of ASR consists of different feature extraction techniques in detail; whereas the back-end of ASR discussed various classification techniques extensively. Another survey paper [42], also reviewed different feature extraction techniques and classification models. In addition to that, the paper briefly defined different types of speech, speech analysis techniques and their impact on the performance of the system; and word error rate (WER), a metric used to calculate the accuracy of the results produced by an ASR. Similarly, [12] focused solely on ASR for under-resourced languages. This paper discussed the definition of under-resourced languages as well as why their preservation is important. The data collection methods of under-resourced languages and the basic structure of an ASR of under-resourced language were also discussed. Correspondingly, [157] comprehensively explained different hybrid HMM-ANN based ASRs, whereas, [32] gave an overview of different ASRs, as well as different approaches that can be used to recognize speech. This paper also briefly discussed different types of speech recognition techniques. In another endeavor, [86] also discussed different types of ASRs, and neural networks based speech recognition approaches.

Table 1 presents the different highlights of the survey papers, which were discussed in the previous paragraph, in a compact and easily comprehensible form. The columns represent the different points that were covered or were missing in the discussed papers.

Most of the previously conducted studies failed to review the different feature extraction techniques and language models that play a vital part in the construction of an ASR. Similarly, the latest deep learning techniques were also not explained in the above-mentioned survey papers. Whereas, different online toolkits and databases that can help train an ASR were also missing from most of the studies. Hence, this study aims to evaluate the different feature extraction techniques and deep learning classification techniques. In addition to that, different online toolkits, databases, and language models were also assessed.

This study captures all the aspects of an ASR from the feature extraction phase to language models with the following objectives in mind:

- To understand and explain the basic structure of an ASR (shown in Fig. 2) in detail, as well as discuss how using different techniques at different stages can affect the overall performance of the system.
- Discuss in detail the different feature extraction and classification techniques being used for the development of an ASR.
- Evaluate different toolkits and advancements made in language models and how they affect the performance of an ASR.
- Encapsulate all of the information available regarding the different modules of an ASR, including different state-of-the-art deep learning classification techniques.

The rest of the paper is organized as follows: Section 2 discusses different tools, resources and techniques that were used to perform this literature review. Section 3 presents a brief history of ASR, different techniques and datasets that can be employed to calculate the accuracy of the ASR, as well as the basic structure of an ASR. Section 4 explains the state-of-the-art techniques being used to extract features from an audio signal, whereas, Section 5 discusses techniques that can be used for classifying the extracted features. Section 6 explains language models, why they are needed, and their types. Section 7 presents the toolkits that can be used to perform different ASR related tasks, and finally, the survey is concluded in Section 8.

2 Research methodology

Before researching this topic, a literature review is performed to determine the cutting edge technologies in this field. In this regard, IEEE, arxiv.org, Microsoft Academic, and Google Scholar were used to search and obtain the papers relevant to the research domain. Most of the relevant scientific seed words were first identified using the generic words and their synonyms related to the domain. Later on, specific seed words, which were identified from different publications, were used.

This method of searching ensured that all of the keywords were present in the titles of the research articles and publications. The AND operation was used to make sure all of the selected words were present in the titles. Double quotations were also used to ensure

Table 1 Highlights and shortcoming of the discussed surveys

Previous Studies	Highlights	Discussed Feature Extraction Techniques	Discussed Classification Techniques	Discussed Deep Learning Techniques	Discussed Language Models	Discussed Existing ASRs	Discussed Available Toolkits and Databases
[42]	• Examined in detail the different speech analysis techniques.	X	✓	X	X	✓	x
[160]	• Discussed different challenges that an ASR will need to overcome.	✓	✓	X	X	x	x
[144]	• Analyzed in detail the different feature extraction techniques and compared their results using multiple classification methods.	✓	x	X	X	✓	x
[97]	• Different types of classification methods were discussed in detail.	✓	✓	X	X	✓	x
[12]	• Focused solely on ASRs for under-resourced languages. • Analyzed different tools available for speech recognition of under-resourced languages.	✓	✓	X	X	✓	✓
[157]	• Discussed hybrid HMM-ANN ASRs.	X	✓	X	X	✓	x
[32]	• Discussed different ASRs in detail.	X	x	X	X	✓	x
[86]	• Discussed different types neural networks used for speech recognition	X	✓	✓	X	✓	x

that all of the words were present together in titles i.e. present as a phrase rather than in the form of individual words. Out of all of the keywords that were used, "Speech Recognition" yielded the most but noisy results. Hence, to get better results, more queries were added to the seed words. The acquired articles were studied and state-of-the-art classification techniques, datasets, and feature extraction techniques were determined. Fig. 1 presents an overview of the methodology followed to perform the research for this survey.

The three factors, that impacted how the literature was filtered, were relevance to the survey topic, how recently the research was conducted, and how thoroughly the paper covered the chosen topic. Table 2 shows the details of databases used to get the literature.

Fig. 1 Overview of search method

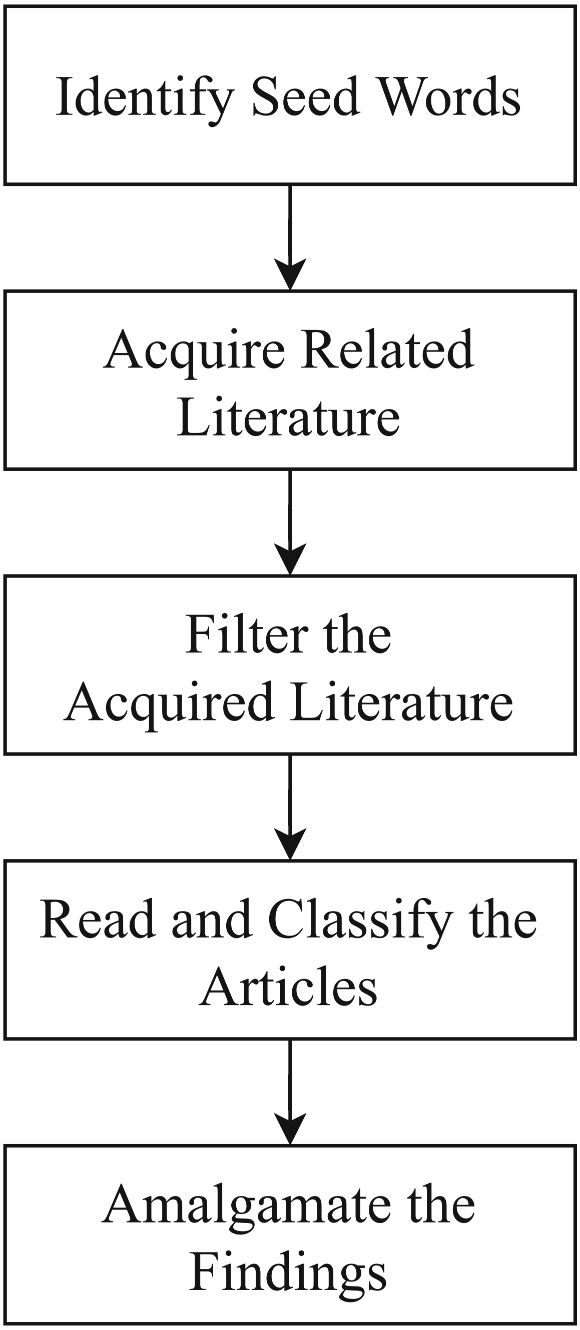

Identify Seed Words
Acquire Related Literature
Filter the Acquired Literature
Read and Classify the Articles
Amalgamate the Findings

Table 2 Databases used for acquiring literature

Database	Research Articles	Conference Papers	Total
ACM	1	3	4
Elsevier	6	0	6
Springer	5	4	9
IEEE	25	50	75
Others	52	34	86

3 Background

Before we get into the technical details of the ASR systems, it is imperative to get familiar with the history of ASR. Hence, this section discusses the first speech recognition system followed by the advancements made to-date. This section also highlights different datasets that can be used for the training and testing purposes of an ASR as well as different evaluation techniques that can be used to measure the performance of an ASR.

Most of the speech recognition models are developed using a generic model. This generic model and its different types are also discussed in this section.

3.1 History and early developments

For quite some time computer scientists have been trying to create a machine that can talk and communicate like a human. Since the early 1950s, researchers have been trying to make a computer understand, interpret and reproduce human languages and speech [53]. The first speech recognition called Audrey was developed in the Bell Laboratories. This system could distinguish between different digits spoken by a single user [33]. Another system was developed in the MIT Lincoln Laboratories in 1959, which could distinguish between 10 phonemes for a single speaker [39]. In the 1970s a lot of important research was made in the area of speech recognition. Russian scientists developed a system that can be used to distinguish words [164]. The ideas of using dynamic programming [138] and pattern recognition algorithms [164] were also presented during these years. In the early 1980s, the hidden Markov model (HMM) was introduced. Even though the HMM was considered to be too simple to identify human languages [62], they still managed to replace the dynamic time warping technique that was being used [69]. In the later years of the 1980s, the n-gram model was introduced. In the early years of the 2000s, the HMM was being used in combination with a feed-forward artificial neural network (ANN) [14]. Nowadays, long-short term memory (LSTM) [14], a type of recurrent neural network (RNN), is being used for speech recognition in combination with different deep learning techniques.

3.2 Evaluation techniques

Evaluation is one of the most important aspects of a conducted research because of its importance this section explains in detail different metrics that can be used to evaluate the performance of an ASR. The performance of a speech recognition system usually depends on two factors, the accuracy of the output produced as well as the processing speed of the ASR.

3.2.1 Speed

The following method can be used to calculate the processing speed of an ASR:

3.2.2 Real-time factor

The real-time factor (RTF) is the most commonly used metric for calculating the speed of a proposed model. The RTF can be computed by using the following formula:

$$RTF = \frac{P}{I}$$

where P is the time taken by the system to process the input and I is the duration of the input audio. If RTF equals 1, then the input audio was processed in "Real-Time". RTF is a highly hardware-dependent value and it is not only limited to calculating the speed of a speech recognition model. It can be used to calculate the speed of any model that can process an audio or video input.

3.2.3 Accuracy

The following methods can be used to measure the accuracy of an ASR:

Word error rate The accuracy of an ASR is hard to calculate as the output produced by the ASR may not have the same length as the ground truth. Word error rate (WER) is the commonly used metric to estimate the performance of an ASR, as it calculates error on word level rather than phoneme level [124]. The WER can be calculated using the following formula:

$$WER = \frac{S + D + I}{N}$$

Where S is the number of substitutions performed in the output text as compared to the ground truth. D is the number of deletions performed, and I is the number of insertions performed. N is the total number of words in the ground truth.

Word recognition rate Word Recognition Rate (WRR) is a variation of WER that can also be used to evaluate the performance of an ASR. It can be calculated using the following formula:

$$\begin{aligned} WRR &= 1 - WER \\ &= \frac{N - S - D - I}{N} \\ &= \frac{H - I}{N} \end{aligned}$$

Where H = N - (S + D) represents the total number of correctly guessed words.

Table 3 List of speech datasets that can be used for training an ASR

Dataset	Open-Source	Hours	Language
LibriSpeech	Yes	1000	English
HUB 5	No	2000	English
TIMIT	No	5.6	English
The CHiME-5	No	50.12	English
TED-LIUM	Yes	452	English
The Spoken Wikipedia [74]	Yes	1005	Multilingual
Common Voice	Yes	1900	Multilingual
CSTR VCTK [162]	Yes	09	English
AISHELL-1 [15]	Yes	170	Mandarin
Persian Consonant Vowel Combination (PCVC) [95]	Yes	–	Persian
Arabic Speech Corpus [49]	Yes	3.7	Arabic

3.3 Datasets

A dataset is essential for the training and testing of an ASR. This section discusses in detail some of the commonly used open-source as well as paid datasets. Table 3 provides a list of the available speech datasets; and their salient features such as total time and spoken languages.

3.3.1 LibriSpeech

LibriSpeech [116] is one of the most frequently used open-source speech-to-text corpus. This dataset consists of 1000 h of audiobooks along with their transcriptions. Because of the large magnitude of the collected data, it was divided into three sets. The first set is comprised of 100 h of training data, the second contains 360 h of training data, and the last set has 500 h of training data. The development set and the testing set have 10.8 and 10.1 h' worth of data, respectively.

3.3.2 2000 HUB5 English evaluation transcripts

2000 HUB5 English evaluation transcripts is the dataset used in deep speech model [50]. It consists of 2000 h of conversational audio and their corresponding transcriptions. This dataset consists of forty source files, all with their corresponding text. Twenty of these files were scripted; a robot operator announces the topic of conversation before the conversation starts. The rest of the twenty files consist of unscripted conversations between Native English Speakers.

3.3.3 TIMIT acoustic-phonetic continuous speech Corpus

Another commonly used dataset for speech recognition is the TIMIT acoustic-phonetic continuous speech corpus [45]. This dataset consists of the recordings of 6300 phonetically rich sentences, read by 630 speakers, where 30% of them are female, and the rest are male speakers. The training set consists of 3.14 h of recording; the rest is divided into the test and development set respectively.

3.3.4 CHiME-5

The CHiME-5 [8] is another dataset that can be used for training an ASR. The main idea behind this dataset was to aid in the creation of a genuinely robust speech recognition system. This dataset contains 50.12 h of recorded conversations in real home environments. The training set of the dataset consists of 40.33 h of data with almost 80,000 utterances. The development set has 4.27 h' worth of data with a little over 7000 utterances. Lastly, the testing set 5.12 h of data with 11,000 utterances.

3.3.5 TED-LIUM Corpus

The TED-LIUM Corpus [131] is an open-source speech dataset containing 452 h of ted talks and their corresponding transcriptions.

3.3.6 Common voice

Common Voice is a great project started by Mozilla, to gather speech data. It is an open-source project, where people can donate their voices, to read out a given sentence, or their time, that will be required to validate whether a particular audio file matches its corresponding transcription. They have gathered 2400 h of data of different languages, out of which 1900 h of data is validated. Currently, they can provide speech datasets of English, German, French, Welsh, Turkish, and 13 other languages.

3.3.7 The spoken Wikipedia

This free dataset contains 1005 h' worth of audio files of three different languages English, German, and Dutch. The English dataset is the largest consisting of 1339 pages of Wikipedia spoken by 465 speakers. This portion of the dataset consists of 395 h of audio files. The German dataset consists of 386 h of audio files covering the content of 1014 pages spoken by 350 speakers. The Dutch dataset is quite small as compared to the other two languages; it consists of only 224 h of data even though it covers the most number of pages of 3171 spoken by 145 speakers.

3.3.8 CSTR VCTK Corpus

This dataset consists of 400 sentences spoken by 109 distinct speakers. All of the speakers are native English speakers with varying ages, gender, and accents. This dataset contains almost 9 h of audio data.

3.3.9 AISHELL-1

This open-source dataset offers 170 h of Mandarin speech data. The dataset consists of 400 unique speakers of all genders and ages. To make the dataset more robust speech on different subjects such as Finance, Science and Technology, Entertainment and Sports were used.

3.4 The architecture of an ASR

The function of an ASR is to take input of a sound wave and convert the spoken speech into text form; the input could be either taken directly using a microphone or as an audio file. This problem can be explained in the following way: for a given sequence input sequence X, where $X = X_1, X_2, \ldots., X_n$, where n is the length of the input sequence, the function of an ASR is to find a corresponding output sequence Y, where $Y = Y_1, Y_2, \ldots., Y_m$, where m is the length of the output sequence. And the output sequence Y has the highest posterior probability P(Y|X), where P(Y|X) can be calculated using the given formula:

$$\begin{aligned} W &= argmax\, P(W/X) \\ &= argmax \frac{P(W)P(X/W)}{P(X)} \end{aligned}$$

where P(W) is the probability of the occurrence of the word, P(X) is the probability that X is present in the signal, and P(X|W) is the probability of the acoustic signal W occurring in correspondence to the word X.

An ASR can generally be divided into 4 modules: a pre-processing module, a feature extraction module, a classification model, and a language model, as shown in Fig. 2. Usually the input given to an ASR is captured using a microphone. This implies that noise may also be carried alongside the audio. The goal of preprocessing the audio is to reduce the signal-to-noise ratio [176]. There are different filters and methods that can be applied to a sound signal to reduce the associated noise. Framing, normalization, end-point detection and pre-emphasis are some of the frequently used methods to reduce noise in a signal [105, 114, 135]. Pre-processing methods also vary based on the algorithm being used for feature extraction. Certain feature extraction algorithms require a specific type of pre-processing method to be applied to its input signal.

After pre-processing, the clean speech signal is then passed through the feature extraction module. The performance and efficiency of the classification module are highly dependent upon the extracted features [3, 78, 178]. There are different methods

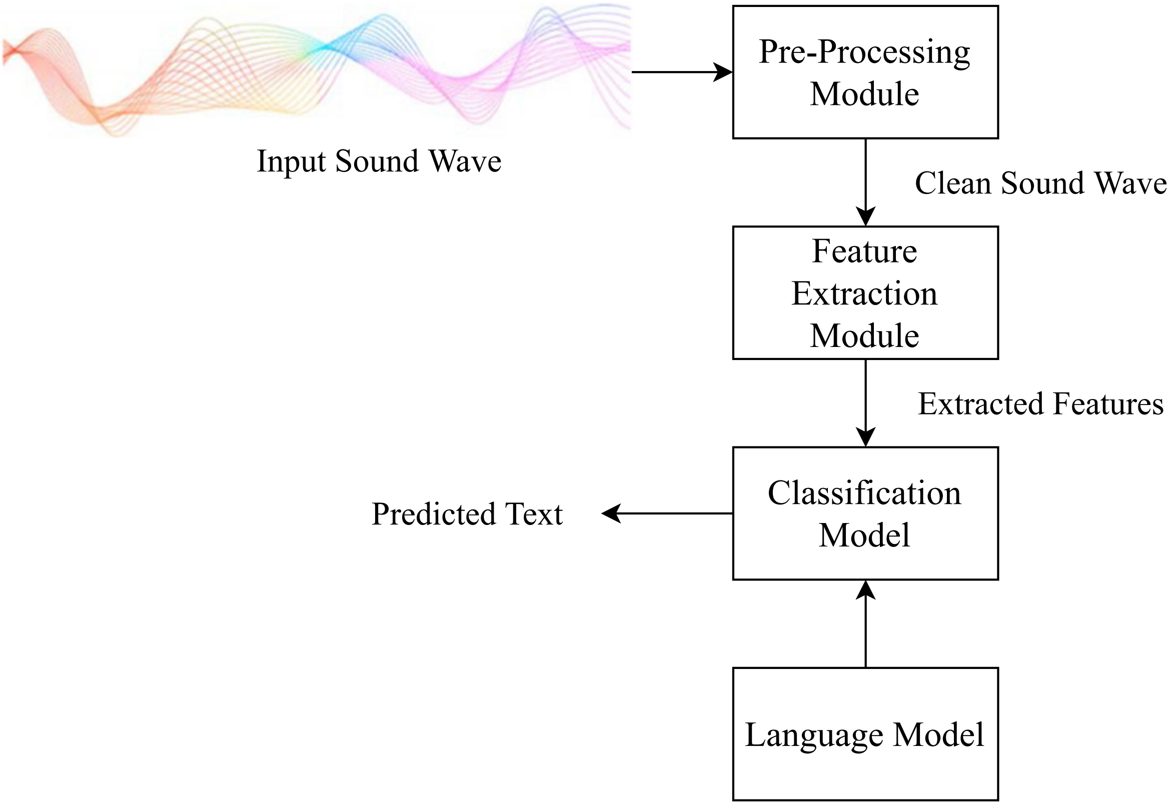


Fig. 2 Basic structure of an ASR

of extracting features from speech signals. Features are usually the predefined number of coefficients or values that are obtained by applying various methods on the input speech signal. The feature extraction module should be robust to different factors, such as noise and echo effect. Most commonly used feature extraction methods are Mel-frequency cepstral coefficients (MFCCs), linear predictive coding (LPC), and discrete wavelet transform (DWT) [40, 78, 112, 127].

The third and final module is the classification model; this model is used to predict the text corresponding to the input speech signal. The classification models take input of the features extracted from the previous stage to predict the text. Like the feature extraction module, there are different types of approaches that can be applied to perform the task of speech recognition. The first type of approach uses joint probability distribution formed using the training dataset, and that joint probability distribution is used to predict the future output. This approach is called a generative approach; HMM and Gaussian mixture models (GMM) are the most commonly used models based on this approach. The second approach calculates a parametric model using a training set of input vectors and their corresponding output vectors. This approach is called the discriminative approach; Support Vector Machines (SVM) and ANN are its most common examples [11, 87]. Hybrid approaches can also be used for classification purposes; one example of such a hybrid model is that of a HMM and ANN [151].

The language model is the last module of the ASR; it consists of various types of rules and semantics of a language. Language models are necessary for recognizing the phoneme predicted by the classifier; and is also used to form trigrams, words or sentences using all of the predicted phonemes of a given input. Most modern ASRs are designed to work without Language Models as well. Such ASRs can predict words and sentences spoken in the given input, but their efficiency can be increased significantly by using a language model [18].

3.4.1 Types of ASR

As shown in Fig. 3, an ASR can be classified on the basis of speaker models, vocabulary being used, channel variability, and speaking style, which can be further classified into two types, utterance speed, and utterance approach.

- Speaker Mode

The purpose of creating an ASR is that it can transliterate any language for any speaker. Languages differ in terms of phonetics, character set, and grammar rules; speakers vary in terms of voice pitch, accent, and personality. Every speaker has a unique voice and speaking style; on this basis, an ASR can be classified into the following three types:

4 Speaker-independent models

Speaker-independent ASRs are developed to recognize multiple speakers. Such systems are not trained for a particular user and are one of the most complex types of systems to design. These systems might offer less accuracy than other methods but are more flexible and can have wide usage in the real world.

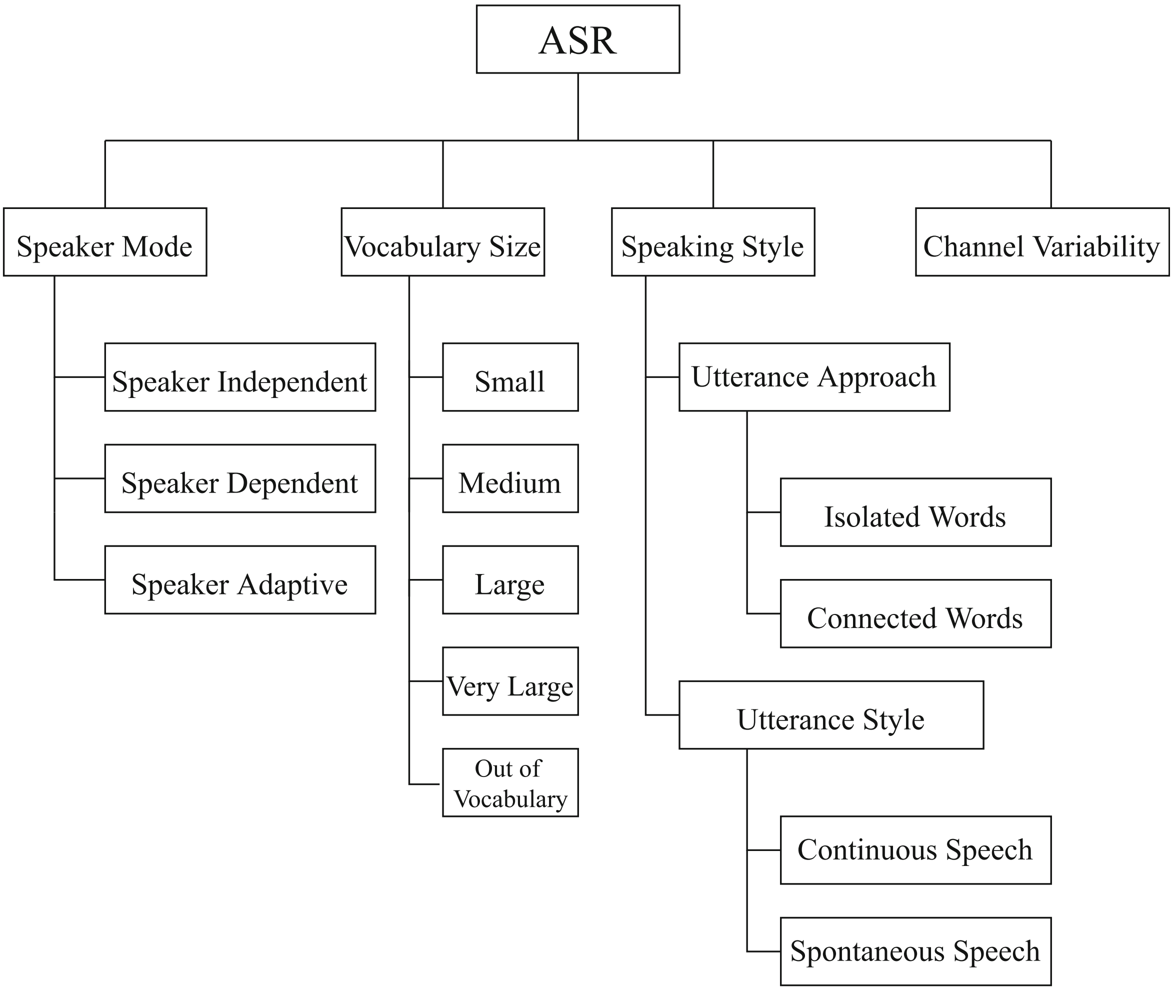


Fig. 3 Types of ASR

5 Speaker-dependent models

Speaker-dependent ASRs are developed to recognize a single user or multiple pre-trained users. Such systems are easily trained and also offer better accuracy than speaker-independent ASRs. But they will not be able to produce the same level of accurate results for voices outside of the user pool that they were trained on.

6 Speaker adaptive models

Speaker adaptive ASRs lie somewhat in between speaker-independent and speaker-dependent ASRs. These systems are trained in such a way that they can learn new speech patterns whenever a new speaker presents itself.

- Vocabulary Size

The vocabulary of an ASR matters a lot as it can affect the complexity, processing time, and the accuracy of the system. The larger the size of the vocabulary, the more complex the system will be; more time will also be required to train the system. The accuracy of the system will also reduce because of the more similar sounding words in the vocabulary. Some ASRs might require a vocabulary of tens of words, for example, a number speech recognition system or a character recognition system. While for others even tens of thousands of words may not be enough; for example, for an ASR that recognizes the English language will require a larger vocabulary than a number recognizing ASR.

7 Small

A small vocabulary can consist of tens of words.

8 Medium

A vocabulary containing hundreds of words is considered to be a medium-sized vocabulary.

9 Large

A large vocabulary can consist of thousands of words.

10 Very large

A very large vocabulary usually has tens of thousands of words.

11 Out of vocabulary

All the words that are not part of vocabulary are mapped as unknown words.

- Speaking Style

In terms of speech recognition, an utterance is a spoken word. A single word, few words, a single sentence, and few sentences can be considered as an utterance as well. Based on utterances type, multiple approaches can be used to develop an ASR.

12 Utterance approach

An utterance is divided into two types: isolated and connected words.

a Isolated Words

A system that is based on the isolated word type of utterance requires its users to take a well-defined pause between each spoken word. This does not necessarily mean that the system will only take one-word input at a time and produce one-word output. Such systems can take multiple words as input but will only process one of them at a time.

b Connected Words

Connected words, on the other hand, consists of a system that works with connected utterances and will take a nominal or no pause between two or more words. Such systems can take an input of multiple words at a time and process them as a whole rather than individually.

13 Utterance style

Since most people have their speaking style, utterances can also be divided into two types on this basis. These two types are continuous and spontaneous speech [82].

a Continuous Speech

In continuous speech utterances, the users of the system are allowed to speak almost naturally. These types of utterances do not require a pause between words. The input given to the system is considered as a whole and is not divided into individual words based on pauses.

b Spontaneous Speech

Spontaneous speech utterances are completely natural. Such utterances may include bogus starts, coughing, laughter, and words like "um" and "ah", etc. These systems are very difficult to develop as the system will require a very large vocabulary. It will also need to be able to differentiate between valid words and other sounds.

- Channel Variability

Another way of classifying ASRs is based on the quality of the input channel. Some ASRs require input signals that are recorded in a clean environment i.e. without any background noise. Noise is unnecessary or unwanted information in the input speech signal. It can be anything from the chirping of birds in the background to distortion from the sound not being recorded correctly. Sometimes the input sound wave also gets distorted when we change its channel by using different software.

Besides noise, the difference in ages, gender, accent, environment, and speaking speed are also considered as variations in the input signal. An ASR should be able to cope with all of the different types of background noises or variations in the input speech signal [40, 71].

14 Feature extraction

The process of feature extraction is applied to remove irrelevant information from the signal. A good feature extraction algorithm should be able to extract the features in real-time and should contain maximum information. Feature extraction algorithms can also be classified based on speech features: temporal and spectral features. The temporal analysis techniques analyze the audio signal in its original form, the time domain. In spectral analysis, as the name implies, the spectral representation of the speech signal is used, the frequency domain. Some of the methods used for feature extraction are the MFCC, PLP, DWT, relative spectral-perceptual linear prediction (RASTA-PLP), and LPC.

14.1 Spectral feature analysis

14.1.1 Mel-frequency Cepstral coefficients

MFCC [37, 114] is one of the most powerful and most commonly used technique for feature extraction [22, 27, 76, 98, 111, 156].

A human ear does not perceive the voice or pitch of a sound linearly. Since many of the applications do not work well with the change in frequency, a scale was introduced in the 1940s, called the Mel-scale. The Mel-scale was developed when researchers were experimenting with how a human ear perceives pitch. It linearized the human auditory system to a linear scale [156]. The experimentations that were performed to develop this scale concluded that only the frequencies between 0 to 1000 Hz could be linearized to the Mel-scale. The values that do not fall in this range were considered to be logarithmic [147]. The following formula can be used to linearize a frequency to Mel-scale:

$$F_{mel} = \frac{1000}{\log(2)} \cdot \left[1 + \frac{F_{Hz}}{1000}\right]$$

Here $\mathrm{F}_{\mathrm{mel}}$ is the resultant linearized frequency, and F_{Hz} is the original frequency of the function.

As we know, a continuous audio function has different values at different points of time. To simplify processing, the audio signal is divided into small frames of either 25 ms [22, 111, 156] [35, 137, 155] or 30 ms [54, 147], where 10 ms of continuous frames overlap. Once the audio signal is divided into frames, each frame is multiplied with the hamming window function, and discrete Fourier transform (DFT) is applied to the result [105]. Sometimes fast Fourier transform (FFT) is also applied to reduce the processing time of the overall process [114]. The results of the Fourier transform is then used to calculate filter bank and then the filter bank is used to calculate log energy outputs using the given below formula:

$$X_i = log_{10}\left(\sum_{k=0}^{N-1} |X(k)| \times H_i(k)\right), \ \ for\ i = 1, \ldots, M$$

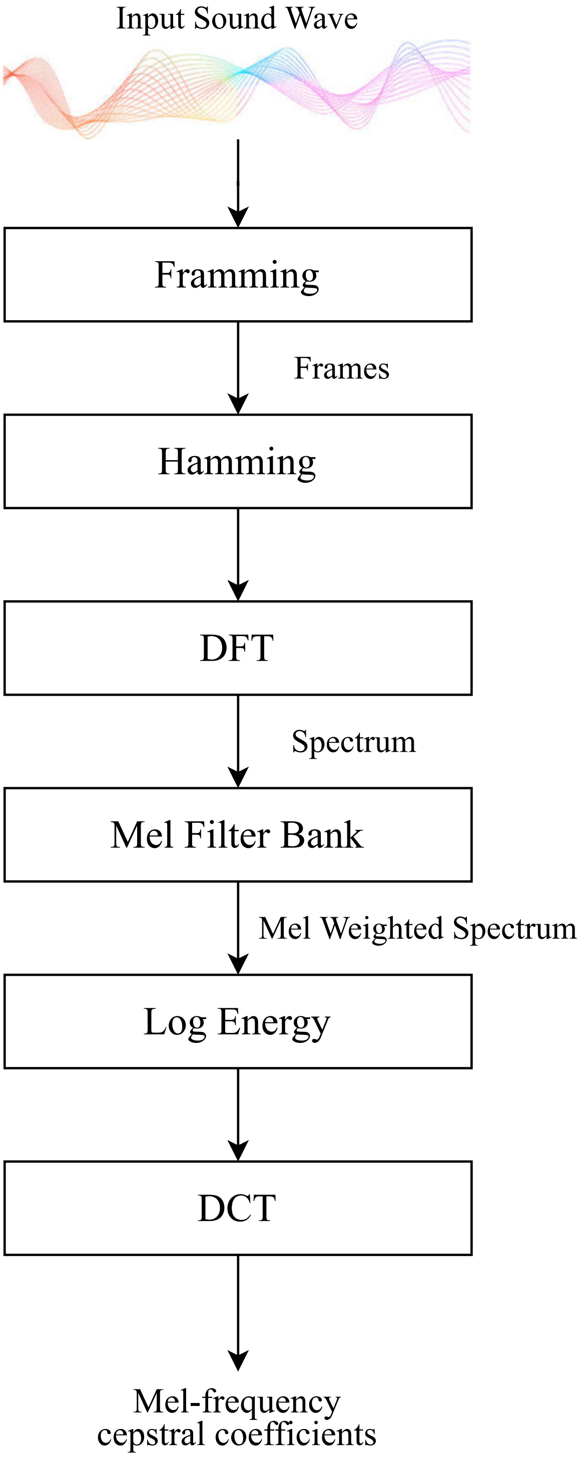
Where $H_i(k)$ is the filter bank, X(k) is the k-th window of source signal X, M is the length of the Fourier Transform, and X_i is the log energy outputs. In the end, Discrete Cosine Transform (DCT) is applied on the log energy outputs using the formula given below:

$$C_j = \sum_{i=1}^{M} X_i \cos\left(j \times \left(i - \frac{1}{2}\right) \times \frac{\pi}{M}\right), \ \ for\ j = 0, \ldots.., J-1$$

Where C_j is the mel-frequency cepstral coefficients, j is the serial index, and J is the total number of MFCC features. DCT allows most of the energy to be preserved while achieving dimensionality reduction by discarding coefficients with high values but low energy [105, 147]. A block diagram that summarizes the process of MFCC is illustrated in Fig. 4.

Though the frames of an input sound are divided into either frame of 25 ms or 30 ms, the influence of one phoneme can extend over more than one frame. Thus, the timing correlation between multiple frames should also be considered for more accurate results. It can be taken

Fig. 4 Block diagram of MFCC process



under consideration by using the delta and delta-delta features of MFCC; the delta MFCC has the addition of the dynamic features; whereas the delta-delta MFCC includes the acceleration features. So, the feature vector obtained from the MFCC algorithm contains three types of features. The first type is the static features, the second is the difference between static features of successive frames or delta features, and the third is the difference between successive dynamic features or delta-delta features. An MFCC feature vector usually consists of thirty-nine dimensions; thirteen for each type of feature; static, dynamic (delta), and acceleration (delta-delta). Another variation of the MFCC feature vector contains the normalized log energy as well; this feature vector also has thirty-nine dimensions, the static feature vector has twelve dimensions in this type instead of the usual thirteen [30, 35, 98, 156].

MFCC may be the most commonly used feature extraction method, but it's not without its limitations. One of the negative features of this algorithm is that it's not adaptive to noise. If even one of the frequency bands in the input signal is distorted the results of MFCC will suffer greatly [47, 63, 91, 106, 111]. Another negative feature is the assumption made during the process of framing; that one phoneme can be mapped to the audio of 25 to 30 ms. As we all know, different speaking styles and accents can sometimes drag one phoneme over the space of two or constrict the information of two phonemes into one frame, so this assumption may not yield the best results. Mean and variance normalization (MVN) [63], cepstral mean normalization [91, 111], and histogram equalization [63] are some of the techniques that can be used to make MFCC more robust.

14.1.2 Linear predictive coding

Linear predictive coding (LPC) [37, 114], released in 1984 [113], is one of the most powerful methods of extracting features from a speech signal, and hence has become one of the most commonly used feature extraction algorithm [107, 108, 151, 174]. Unlike MFCC, which resembles the human auditory system, LPC imitates the basic structure of the vocal tract [30]. It can also be easily compared with the basic model of speech production which is also modelled as a linear but time-varying system for both periodic pulses or voiced sounds and random noises [48, 135, 157].

The basic idea behind this algorithm is that the current sample can be represented as a linear combination of all of the previous samples. The LPC analysis can be calculated by first dividing the input audio into frames and then performing the process of windowing on these frames to make sure there are no discontinuities in the beginning or end of any frame. The last step of the process is to calculate the auto-correlation between the frames. And then the LPC analysis is performed on the obtained auto-correlation values, by using Durbin's Method [48, 108, 113] or by using the formula given below [48, 178]:

$$s[n]\approx \sum_{k=1}^{p} a[k]s[n-k]$$

Where s[n] is the current sample point, p is the total number of previous sample points, which are also called predictors [4], and a[k] which is the predictor coefficient.

The main goal of LPC is to calculate the coefficients of a[k] for each frame where E, the total squared prediction error, is minimum. So, once the LPC analysis is performed, the total squared prediction error can be calculated using the formula given below:

$$E=\sum_{n}\left(s[n]-\sum_{k=1}^{p} a[k]s[n-k]\right)^{2}$$

14.1.3 Linear predictive Cepstral coefficients

After performing an LPC analysis on the given input audio, the following formula is applied to get linear predictive cepstral coefficients (LPCC) [125]:

$$\hat{v}[n] = \ln(p), \qquad for\ n = 0$$
$$\hat{v}[n] = a[n] + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right)\hat{v}[k]a[n-k], \quad for\ 1 \le n \le p$$

Where p is the total number of sample points, $\hat{v}[n]$ are cepestral coefficients, and n is the number of samples present in the anaysis frame.

Recent research [21] examined the performance of LPCC as compared to MFCC. The system that was used to study these feature extraction algorithms could identify twelve Hindi words spoken by five different speakers. This system showed that LPCC and MFCC had similar results. Another research [165] showed that LPCC was 10% more efficient and 5.5% faster than MFCC. Fig. 5 sums up the process of LPCC in the form of a block diagram.

14.1.4 Perceptual linear prediction

PLP uses transformations that are based on a human auditory system. This algorithm has three main characteristics; the spectral resolution of the critical band, application of intensity-loudness power law, and equal loudness curve reduction. By remapping the frequency axis to the Bark scale, PLP incorporates critical band spectral resolution into its spectrum estimate and produces a critical band spectrum approximation. This approximation integrates the energy in critical bands. As we know, human hearing is more sensitive to the middle-frequency range of audible spectrum at conversational speech levels. PLP incorporates this phenomenon in the algorithm by multiplying the loudness curve with the critical spectrum band. By doing this, the high and low-frequency regions are suppressed between the range of 400 kHz and 1200 kHz, which is the mid-range. A nonlinear relationship exists between the perceived loudness and the intensity of sound. Cube root amplitude compression of the loudness equalized critical band spectrum estimate is used to approximate the power law of hearing [88].

To calculate the coefficients of PLP, windowing is performed on the input signal, and then an FFT is applied on the windowed input signal. The resultant signal is then converted into Bark Scale using the formula given below:

$$\theta(B_i) = \sum_{B=-1.3}^{2.5} |X(B-B_i)|^2 \, \psi(B)$$

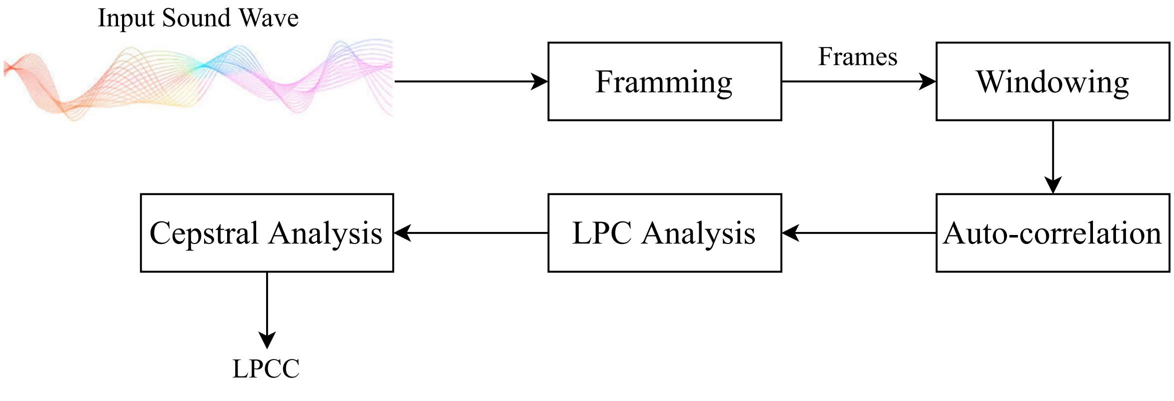


Fig. 5 Block diagram of LPCC

Where is the Bark-scaled frequency, and X is the input signal. The Bark scaled frequency ensures that the critical band frequency selectivity is modelled inside the range of human cochlea [105, 125]. Once the Bark-scaled frequency is calculated, it is weighted according to the equal-loudness curve, and then the intensity-loudness power law is applied to the acquired weighted frequency. Inverse Fourier transform (IFT), linear predictive analysis, and cepstral analysis are performed in order to get the PLP coefficients [105, 125]. Fig. 6 summarizes the steps performed in PLP in the form of a block diagram.

The research performed in [55] showed an HMM-ANN system that recognized English language phonemes and used PLP as its feature extraction algorithm. The system used TIMIT corpus for training and testing purposes. The accuracy achieved was 64.9%, but when the system was tested on HTIMIT, which consists of speech data collected over different telephone channels, the accuracy dropped to 34.4%. The research performed in [44] discussed the performance of PLP in comparison with MFCC in noisy environments. The research used two different types of noise signals: white and street noise. The system used was a multi-lingual system that could recognize words of six languages: Hungarian, English, French, Italian, Spanish, and German. The results obtained from the research showed that PLP achieved 0.2% more accuracy than MFCC.

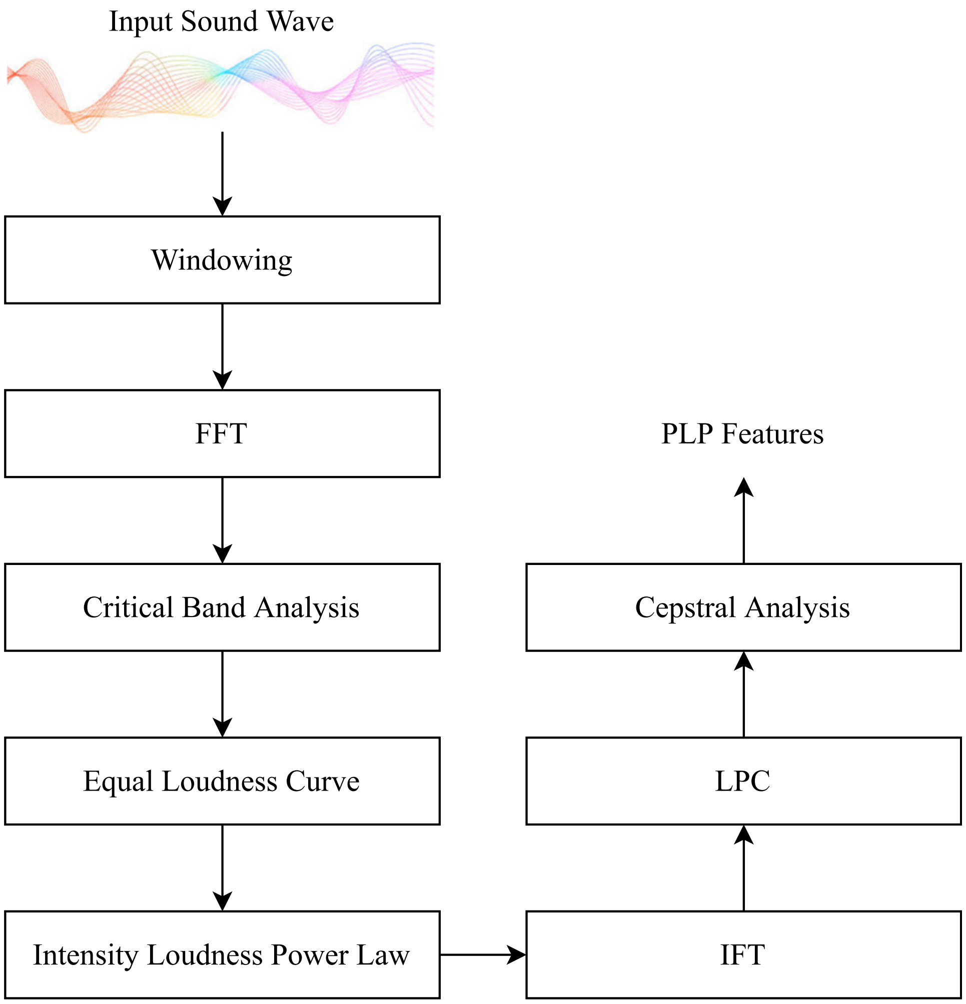


Fig. 6 Block diagram of PLP analysis

14.2 Temporal feature analysis

14.2.1 Relative spectra–perceptual linear prediction

RASTA-PLP analysis specializes in noisy environments by merging RASTA and PLP analysis. It can easily be observed that often training and testing data's conditions differ, testing data usually contains more real-life factors such as noise, inter-speaker variations, intra-speaker variations, and a difference in the transmission channel. The basis of RASTA [139] analysis is that the temporal properties of the environment, in which the input signal was recorded, varies from the temporal properties of the speech. So, by using a band-pass filter on all frequencies in each sub-band, the short-term noise is smoothed, and the difference in training and testing environments is reduced significantly. The block diagram shown in Fig. 7 explains the steps performed to calculate RAST-PLP features.

Another research [56], compared LPC, MFCC, and RASTA-PLP as feature extraction techniques for a system that recognized digits of Kannada language. The input signals were pre-processed using wavelet transforms; DWT was used for clean signals, whereas, the wavelet packet transform (WPT), was used for noisy signals. For clean speech signals, MFCC had the highest accuracy of 94%, followed by LPC, which had an accuracy of 82%, and RASTA-PLP had an accuracy of 54%. For the noisy signals without pre-processing, RASTA-PLP had the highest accuracy of 73%, followed by MFCC, with an accuracy of 60%, and LPC had the lowest accuracy of 53%. After applying the WPT, the accuracies of all three feature extraction methods were increased, with RASTA-PLP having the highest accuracy of 83%.

Hence we can easily say that, for noisy datasets, RASTA-PLP performs much better than any other feature extraction method, whereas it may not perform as well for clean speech signals. It was also observed in [56] that RASTA-PLP can have an even better performance when combined with WPT.

14.2.2 Discrete wavelet transform

We know that speech signals are not stationary and contain both temporal and frequency information. Even though most algorithms focus only on frequency information, temporal information is equally important [5, 121, 137]. DWT takes into consideration the temporal

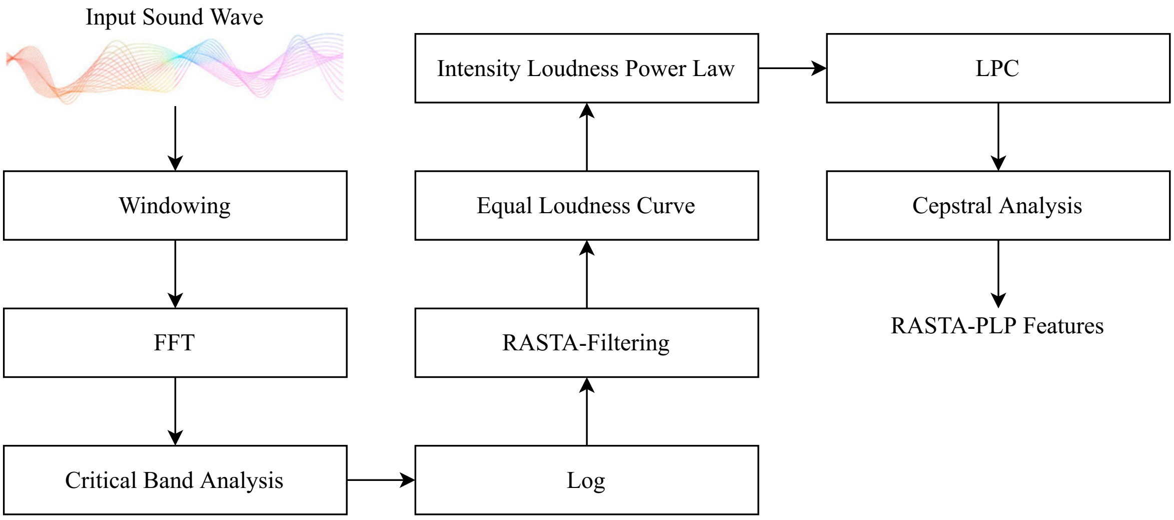


Fig. 7 Block diagram of the process of RASTA-PLP

information present in the input audio signal by re-scaling, shifting, and then analyzing the mother wavelet to obtain the temporal information present in the input signal. Because of this, the input signal is not only analyzed on different frequency levels but with different resolutions as well [5, 121].

So, DWT is based on multi-resolution analysis, according to which lower frequency components appear for a much longer duration than the higher frequency components in a speech signal. Because of this reason, instead of using the same size window, different sizes of windows are used for lower and higher frequency components. For a higher frequency component, a narrow window is used, and a wider one is used for lower frequency components [121]. DWT was created to replicate the working of a human auditory system, where decreasing frequency resolution is used to analyze the increasing frequencies present in a signal [135].

The DWT analysis divides the input speech signal into two types of coefficients: detail and approximation coefficients. The detail coefficients represent the low-scale high-frequency components of the input signal, and approximation coefficients represent the highscale low-frequency components [78, 127]. DWT can be performed using a formula proposed by Stephane G. Mallat [96] this is a fast pyramidal algorithm that uses multi-rate filter-banks, called Mallattree decomposition. The algorithm decomposes the signal into detail and approximation coefficients, as shown in Fig. 8.

The input speech signal is passed through a high-pass and a low-pass filter to get the detail and approximation coefficients, and then the results obtained from filters are down-sampled by two, the results obtained after down-sampling are the required coefficients. The process of applying the filters and down-sampling can be mathematically expressed in the form of the formulas given below [5]:

$$y_{low}[k] = \sum_{n} x[n] \times h[2k-n]$$
$$y_{high}[k] = \sum_{n} x[n] \times g[2k-n]$$

Where x[n] is the input signal, h[n] is the low-pass filter, and g[n] is the high-pass filter. The approximation coefficients can be further divided by using the same steps repeatedly.

Speech signal often lies in the lower frequency components of a signal, even if the higher frequency components are removed from a signal, the speech present in the signal will still be understandable even though the overall sound of the signal will be different. The research done in [43] shows that instead of using the detail coefficients, using approximation coefficients to generate octave achieves better accuracy.

DWT coefficients are obtained by concatenating the approximation and detail coefficients starting from the last decomposition level. The total number of decomposition levels is chosen based on the frame size. The frame sizes between 3 and 6 octaves are commonly used. The filters used for computing DWT should be a quadrature mirror filter (QMF), which can be calculated using the formula given below:

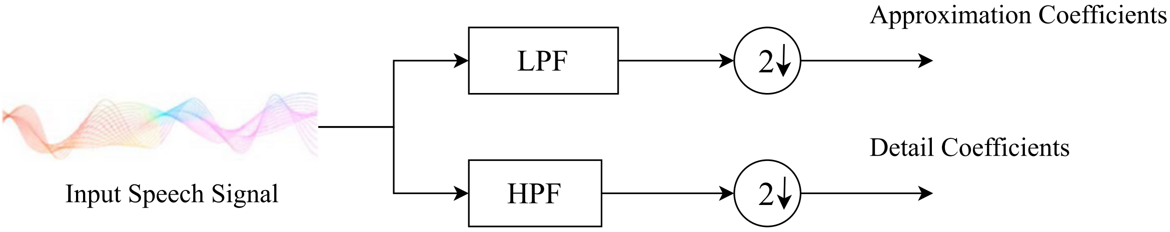


Fig. 8 Decomposition of speech signal into high frequency and low-frequency components

$$g[L-1-n] = (-1)^n \times h[n]$$

where L is the length of the filter. The QMF relationship will ensure that the original input can be perfectly reconstructed from the decomposed signal.

DWT is very robust to noise as it works with localized time and frequency information. Hence, if one of the frequency bands of the input signal is altered by the noise, it will not affect all of the coefficients produced by this algorithm. Due to this reason, many of the researches related to ASRs used DWT as their feature extraction method [43, 64, 100, 151, 167].

14.2.3 Wavelet packet transform

WPT is very similar to DWT. The only difference is that the detail and approximation coefficients are more decomposed in WPT as compared to DWT [78]. The research done in [105] compared the performance of DFT based algorithms with algorithms based on DWPT for the task of speech recognition. One of the DFT based algorithm under consideration was MFCC. This research showed that DWPT based methods performed better as compared to DFT based algorithms. When compared against MFCC, a reduction of 20% in word error rate was achieved with a DWPT based method. Another research [105], compared the performances of WPT against DWT. The system being used was an ASR that could identify the Malayalam language. Here DWT outperformed WPT, as DWT achieved an accuracy of 89%, as compared to, WPT which could only achieve 61%.

14.3 Summary

From the above discussion, it was easy to conclude that in the past, feature extraction techniques that focused on spectral analysis preferred over techniques that used temporal analysis. However, over the past few years, it became obvious that spectral analysis alone was not enough to gather maximum information from the input speech signal. Hence, the wavelet techniques, which used temporal analysis, were used in some researches instead of MFCC and LPC. DWT achieved better results for the task of phoneme recognition than the more commonly used MFCC.

Storage space is a factor that should be taken into consideration when discussing feature extraction techniques. DWT is preferred if there is limited space available, as its feature vector is much smaller in size. There are other feature extraction techniques, such as Principal Component Analysis (PCA), Vector Quantization (VQ) and Linear Descriptive Analysis (LDA), that can also be used in combination with other methods, such as MFCC, to reduce the dimensions of their feature vectors. Different researches used VQ with MFCC [153] and DWT [127] to utilize its clustering property to improve the performance of their ASR. Whereas the PCA and LDA were used to reduce the dimensionality of the feature vector, all the while making the system more robust [38, 60, 163]. Another point to be considered when selecting a feature extraction technique is the type of environment the ASR will be deployed in. In clean environments, MFCC, PLP, and LPC achieved good accuracies; whereas, for noisy environments, DWT, LPCC, and WPT showed better results. One way to make an ASR more robust is to combine the MFCC, PLP, and LPC with either DWT or WPT. Another way is to use RASTA-PLP, which performs best in a noisy environment but not in clean environments.

Table 4 summarizes the advantages and disadvantages of all of the above-mentioned techniques.

Table 4 Advantages and disadvantages of the discussed feature extraction methods

Feature Extraction Method	Advantages	Disadvantages
Mel-Frequency Cepstral Coefficients [84]	• MFCC provides good discrimination between phonemes [48]. • It closely resembles the human auditory perception system because it is not linear [48]. • It can capture important information present in the signal [48].	• It is not robust to noise [76] [111]. • It may not be able to map continuous phonemes correctly.
Linear Predictive Coding [91]	• It represents the vocal tract and is an accurate and reliable method of getting features [103]. • It is very robust and can extract features even from speech signals that have a low bit rate.	• It can not successfully distinguish between words containing similar-sounding phonemes [142]. • It might not be able to represent speech, as LPC assumes that the given signal is stationary and hence, cannot analyze local events accurately. • The feature coefficients have a high correlation among them.
Linear Predictive Cepstral Coefficients [109]	• The high correlation of LPC is removed by applying the cepstral analysis [119]. • It is more robust than a simple LPC analysis.	• It might not be able to represent speech properly, as it assumes that the given signal is stationary and cannot analyze local events accurately. • It cannot retain prior information in the testing phase.
Perceptual Linear Prediction [88]	• The difference between voiced and unvoiced inputs is reduced. • It is independent of the length of the vocal tract. • The feature vector produced has relatively fewer dimensions.	• The feature vector being produced is highly dependent on the spectral balance of the formant amplitudes. • Channel, noise, and the equipment used to get the input signal can easily change the spectral balance.
Relative Spectra–Perceptual Linear Prediction [72]	• It is very robust. • It removes slow and fast variations present in the speech signal [57]. • RASTA-PLP captures low modulation frequencies, which correspond to speech in a signal [20].	• It doesn't perform well for speech signals without noise.
Discrete Wavelet Transform [168]	• DWT considers temporal information present in the signal alongside frequency information. • It can perform de-noising tasks successfully [99]. • The input signal can be recreated perfectly from the decomposed parts.	• The same base wavelet is used for all input signals, which makes this algorithm inflexible.
Wavelet Packet Transform [120]	• Same as DWT, the only difference is that this algorithm gives details present in the high-frequency bands as well.	• The same basic wavelets need to be used for all speech signals, which makes this algorithm inflexible.

15 Classification

After features are extracted, they are passed as input to a classifier. This is one of the most important and time-consuming modules, as a classifier predicts the phoneme or word that is spoken in the input signal. The job of a classifier is to learn the relationship between the given input audio features, and their corresponding text or phonemes. They are first trained using the training data, which should be big enough for a classifier to recognize the specific patterns present in the speech signal and their correspondence to the output phonemes. Many types of research have been conducted to find which classifier is best suited for speech recognition. The most commonly used classifying techniques for speech recognition are HMM, ANN, and SVM.

15.1 Hidden Markov model

HMM has been one of the most successful classifiers in terms of speech recognition. Due to this reason, it is also one of the most commonly used technique [26, 83, 114, 118, 137]. It is very flexible and can easily adapt according to the required structure. Hence, making it very easy to train and implement, with efficiency [13, 68, 114, 137].

HMM is a stochastic model, and the number of states established during the process of training is fixed and pre-defined. These states may vary from the number of hidden states in the input speech signal. HMM assume that the given speech signal can be characterized as a parametric random process, and thus its parameters can be determined in a well-defined and precise manner. This algorithm is an extension of the Markov chain, which can produce output symbols regardless of the state they are in [13, 110]. Resultantly, the output of HMM is a probabilistic function of the state, and for the input sequence, the state sequence is not observable, hence, the use of the word hidden in the name of the algorithm. An example of HMM is shown in Fig. 9. This example was taken under consideration since most ASRs use left-to-right HMMs to properly model the temporal features present in the input speech signal.

Mathematically, HMM can be defined as $\lambda(S, M, A, B, \pi)$, where $S = S_1, S_2, \ldots.., S_n$, and is the set containing all possible states. M is the total number of unique output symbols per state. A: a_{ij} is the probability of state transition, where a_{ij} is the probability of transitioning from state S_i to S_j, it can be calculated using the following formula:

$$a_{ij} = P\left(T_{t+1} = S_j \mid T_t = S_i\right)$$

B: $b_j(k)$ is the probability of an output symbol and can be calculated using the formula given below:

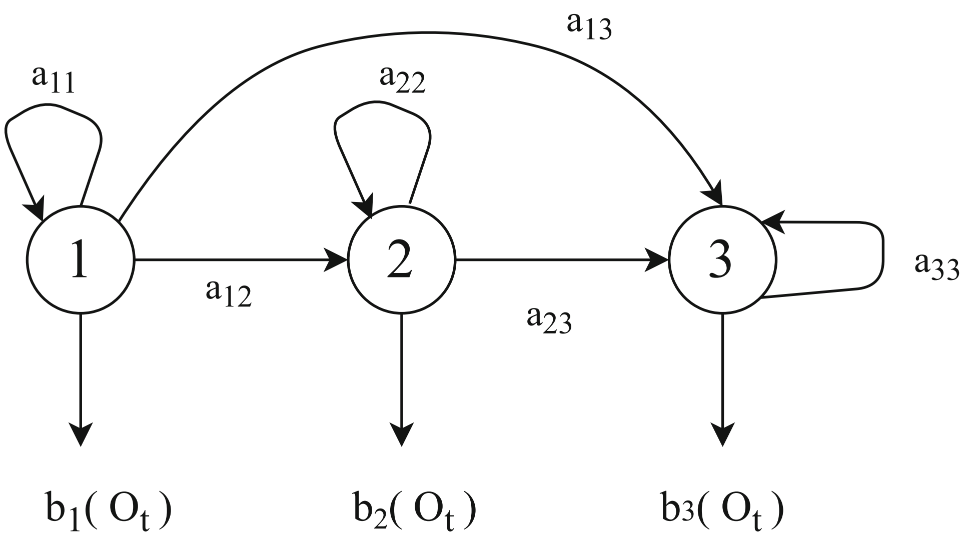


Fig. 9 An example of left-to-right HMM with three states

$$b_j(k) = P(v_k \; at \; t \mid T_t = S_t\Big)$$

π is the set of initial state probabilities, and it contains the probabilities of every state S_i as a start state, and V = {v_1, v_2, …., $v_{m\}}$ is the set of all possible output symbols. For an input set of observations O = o_1, o_2, ….., o_T and an HMM model λ = (A, B, π), we can use the following formula to calculate the probability of a single observation [13, 123]:

$$P_r(O \mid \pi, A, B\Big) = \sum_q \pi_{q_t} \prod_{t=1}^{T} a_{q_{t-1}} b_{q_t} (O_t)$$

A combination of wavelet transform and HMM was introduced in [68]. HMM, and wavelet transforms were used together to boost the performance of wavelet-based algorithms. This hybrid model was called the Hidden Markov Tree (HMT) model. Even though the wavelet transformation algorithms produced great results for speech recognition, their performance could improve if dependencies between their coefficients could also be calculated, as each wavelet was treated independently. With the HMT model, Markov structures were created between the wavelet coefficients to model the dependencies. These structures were not applied directly to the wavelets but were applied in between the wavelet coefficient states. The resultant binary tree had wavelets connected vertically across the scale. The performance comparison between HMT and some wavelet transformations was done by applying them both to a simple classification problem. As predicted, the HMT showed better results than wavelet-based algorithms. As mentioned in Table 4, wavelet based algorithms are very robust. De-noising of different noisy speech signals was also performed to compare the performance of HMT. Again, HMT showed better results than wavelet-based algorithms. [1] presented an enhanced version of HMT that can be used for feature extraction.

CDHMM [29, 98, 99] is the most recently developed approach using HMM. This technique uses a maximum likelihood (ML) algorithm for training and recognition of HMM. Using this technique, variations occurring within and between phonemes can be calculated [98]. CDHMM can be further improved by using the large margin classifiers in the training process. When compared with conventional Machine Language techniques, this technique had reduced error rates [19, 29, 70].

15.2 Artificial neural networks

ANN are great classifiers, and they produce the best results for pattern recognition problems. They are used for their capability to learn and organize according to the dataset provided at the training stage. They work exceptionally well with unknown data and can classify unknown data effectively. The drawback of using an ANN is that they tend to over train and face the local minima problem. They also ignore the time variability present in the speech signal; this problem can be solved by using Hybrid HMM-ANN models. The hybrid model is used to get the advantages of both the models [137].

Some of the widely used ANN are discussed below.

15.2.1 Multilayer Perceptrons

Multilayer perceptrons (MLP) have proven to be the most efficient, successful, and commonly used type of ANN [137, 156]. An MLP is a simple feed-forward neural network containing at least three layers: input, output, and hidden. Fig. 10 shows the basic structure of an MLP.

This algorithm is applied during the training phase; it is based on the backpropagation approach and the concepts of lateral inhibition. The generated output is based on the output neuron with the highest activation. One of the major drawbacks of this model is that they can only take input of fixed length, which makes them unable to handle the dynamicity of the input speech signal. Another problem is that this algorithm can only deal with small vocabularies efficiently, which makes them a good phoneme recognizer but not an efficient word recognizer [67].

The work proposed in [141] used MLP to recognize digits of the Urdu language. The dataset used for the training purposes composed of speech signals of a single user, recorded in a clean environment. FFT and MFCC were used to extract the features from the speech signal. An accuracy of 94% was achieved in the testing phase. Another research [145] used MLP to recognize Persian digits. An accuracy of 98% was achieved by first using MFCC to perform denoising on the dataset, and DWT was used to extract the features. The dataset used for training purposes consisted of the data of a single male speaker. [102] used a deep MLP network to perform speech emotion recognition. The research used the speech data present in the IEMOCAP database [146]. The network was composed of an input layer, five hidden layers, and three output layers, one layer for each metric. The model achieved mean scores of 0.453 and 0.469 when testing with speaker-independent and speaker-dependent data, respectively.

Sparse multilayer perceptrons (SMLP) [6, 67] is a technique that is based on the concept of MLP. SMLP is almost identical to MLP in the structure; the only difference is that one of the hidden layers of SMLP must produce a sparse matrix as output.

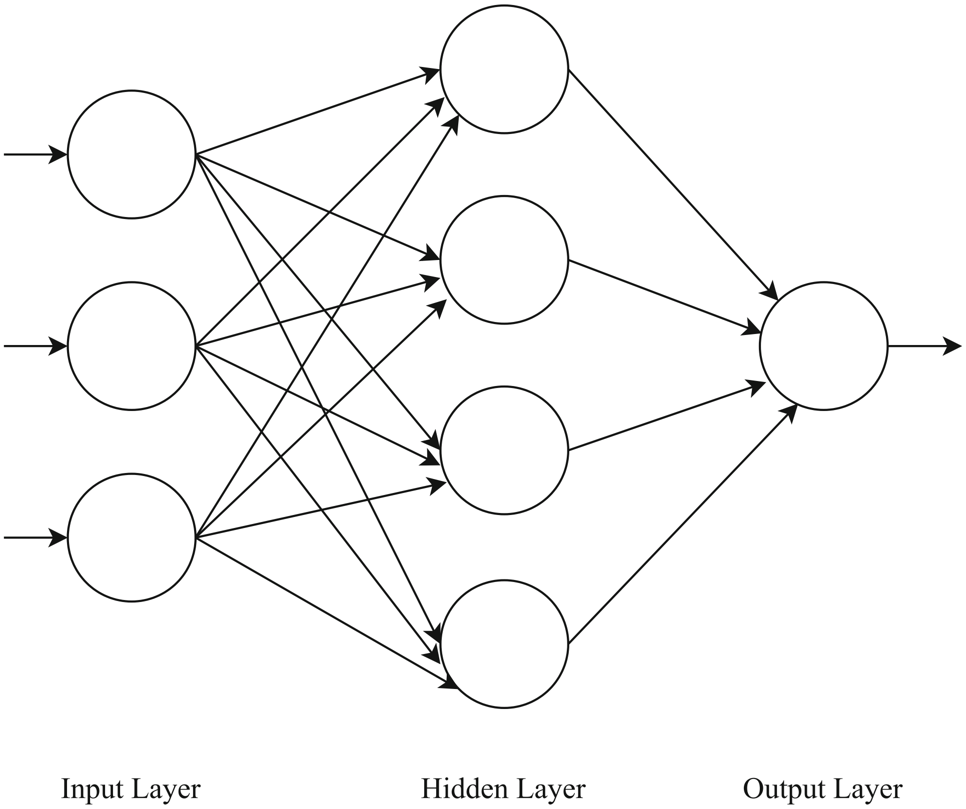


Fig. 10 An example of a simple MLP

15.2.2 Self-Organising maps

Self-organizing maps (SOM) were introduced in 1982 by Teuvo Kalevi Kohonen [6]. The main idea behind SOM is that input signals are placed in such a way that they can produce a contour map from a higher dimension input space to a lower-dimensional feature space. So, the input signal is first placed randomly in the input feature space, which is then organized into different clusters. Each of the formed clusters represents a unique feature of the input signals. Because of this, SOM can easily differentiate between different features present in the input signal [16, 21, 147].

The SOM can differentiate between the signals without supervision and therefore have no example of potential output. Hence, for the SOM network to be trained satisfactorily, we need a significant number of training samples. This algorithm can be performed by applying the following three steps. The first step is to calculate the level of similarity between the pattern present in the input signal and the neurons present in the output layer. The required similarity can be calculated with the help of the predefined formula of Euclidean distance. After that, the synaptic weights are determined, using the formula given below:

$$w_j(n+1) = w_j(n) + \alpha(n)h_{j,i(x)}(n)\left(x(n) - w_j(n)\right)$$

Where x is the input function, $\mathrm{w_j(n)}$ is the weights of neuron j at the time n, α(n) is the learning rate and hj,i(x) is the neighbourhood function.

The research performed in [75] used SOM in combination with DWT to perform the task of vowel recognition. This system was named the wavelet self-organizing maps (WSOM), which used SOM to model the input speech signals, and the resultant SOM mapping was used to adapt the wavelets. The WSOM obtained an accuracy of 55%. Another research [31] used SOM to convert variable length feature vectors into fixed-length feature vectors. This technique ensured that the MLP classification model used in the system will always have fixed-length feature vectors even though the length of the input signal can be variable.

The research performed in [21] used SOM to identify twelve different Hindi words spoken by five speakers. The SOM used in the research consisted of an input layer, a competitive layer, and an output layer; it is a modified version of a basic SOM called supervised SOM. In this research, four different types of features were extracted from the input signal, and these features included the intensity, and the pitch of the signal, MFCC, and LPC. The accuracy of every speaker was analyzed independent of the other speakers. The highest accuracy was achieved by the intensity features, whose mean-SOM and median-SOM, accuracy was 98.17% and 98.54%, respectively. The other feature extraction techniques achieved approximately 89% accuracy.

15.2.3 Radial basis functions

Radial basis functions (RBF) have the basic ANN structure, i.e., an input layer, an output layer, and a hidden layer. The main difference between RBF and other ANN structures is that Gaussian function is used in the hidden layer. The main task of the RBF model is to generate clusters on the basis of patterns present in the input speech signal. The Gaussian function is then used to form a relationship between all of the created clusters. This relationship is formed by applying the Gaussian function in the centers of these clusters. Hence, the output of this model can be calculated using the formula given below:

$$y = \sum_{h=1}^{H-1} w_h \Phi_h(x)$$

where H is the total number of hidden layers, w_h are the linear weights, x is the input signal, and Φ_h is the Gaussian function, which can be calculated using the following formula:

$$\varPhi_h = e^{\left(\|x-c_h\|/2\sigma_h^2\right)}$$

Where c_h is the centre of the Gaussian function and σ_h is the width of the Gaussian function.

The research done in [17] compared the performance of MLP and RBF. The features were extracted using LPCC, and the system that was used to compare the two classifiers could identify six words of the English language, which are spoken by six speakers. MLP achieved 96% accuracy while RBF achieved an accuracy of 98.69%. The training and testing speed of RBF was also faster than MLP. The research performed in [117] combined RBF and HMM. The main task of this system was to recognize words spoken in a continuous speech environment. Cepstrum analysis was performed on the input speech signal to extract features from it. The RBF-HMM hybrid approach created a new HMM for every word in the training data and associated a target value to each of these HMM. The target value was then used to calculate the best possible number of neurons for the hidden layer of the network. This system achieved an accuracy of 80% for recognizing ten words and with a total number of eight neurons in the hidden layer.

[166] researched the combination of wavelet transformation and RBF to create a robust ASR. The wavelet transform and RBF were combined in such a way that the activation function of RBF was replaced with a wavelet transformation. The accuracy of the system was tested over sixteen speakers speaking different numbers of words, in different environments. The wavelet-RBF hybrid model achieved better results as compared to a simple RBF network. But it was observed that as the number of words in the vocabulary increased, the accuracy of the hybrid model decreased to the point where it was equivalent to the simple RBF network. Hence, it was assumed that for large vocabularies simple RBF network is better than a wavelet-RBF model.

[159] proposed a model using temporal RBF features to recognize Arabic letters. The model is divided into three modules: preprocessing, feature extraction, and classification. The preprocessing module removes salience from the input signal and then performs normalization, pre-emphasis, framing, and windowing on the signal. Once the signal is pre-processed, its different statistical features were calculated. The calculated features are then used as input for the RBF-based classification model. The research achieved a recognition rate of 98.175%.

15.2.4 Recurrent neural network

RNN [58] model doesn't require any phonetic dictionaries or extra human effort to transcribe the input audio if it's trained properly. For a given input sequence $x = (x_1,\ldots..,x_T)$, a RNN will calculate two things, the output vector $y = (y_1,\ldots,y_T)$ and the vector used to store the values of its hidden states $h = (h_1,\ldots..,h_T)$. An RNN uses the formula given below to find the values of the output and hidden vector:

$$h_t = \mathcal{H}(W_{ih}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{ho}h_t + b_o$$

Where $\mathscr{H}$ is the activation function, W_{ih} denotes the weight matrix used between the input and the hidden states units, W_{ho} represents the weight matrix used between the hidden and output units, and h_{t-1} represents the previous state's values.

Since most RNNs use LSTM cells, the following formulas can be used to mathematically describe it.

$$\begin{gathered}
i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \\
c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t = o_t \tanh(c_t)
\end{gathered}$$

where i_t represents the value of the input gate for the current iteration, f represents the forget gate, o represents the output gate, c represents the cell activation function, and σ represents logistic sigmoid.

One major shortcoming of using a simple RNN is that it would only consider the previous context. However, in speech recognition, the future context is equally important as the previous context. So, instead of using a simple RNN, bidirectional RNN can be used to address this shortcoming. As the name describes, a bidirectional RNN processes the input vector in both directions and keep separate hidden state vector for each direction, and the following formulas can be used to describe the processing of a bidirectional RNN:

$$\begin{gathered}
\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right) \\
\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \\
y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_o
\end{gathered}$$

Neural networks, both feed-forward and recurrent, can be only used for frame-wise classification of the input audio. This problem can be addressed by using HMMs to get the alignment between the input audio and its transcribed output. Another method would be to use CTC [58], as the objective function, as it trains the model without knowing the initial alignment between the given input and the transcribed output. To decode the output of a CTC network, there are two methods. One method is to pick the output with the highest probability at the end of every time step. Another way is to use the beam search. If the beam search is used, then a dictionary and a language model can also be integrated with the model to increase its efficiency. Fig. 11 shows an example of a simple RNN.

[134] presents an Attention-based Transducer, where the encoder is composed of a pyramid LSTM layer, a simple LSTM layer, and a multi-head self-attention layer. The input signal is fed to the pyramid LSTM layer; the output of this layer is then concatenated with the two previous outputs. The concatenated output is then passed to the LSTM layer and then finally to the multi-head attention layer. The decoder used in the model is also composed of two simple LSTM layers. The data used consisted of 10 K hours of in-house English speech data gathered by the authors using the LAIX learning application. The proposed model achieved a WER of 10.3% and a RTF of 0.19.

The model proposed in [46] used RNN to recognize Bengali speech. The network consisted of three fully-connected layers, followed by a bidirectional RNN layer and then another fully connected

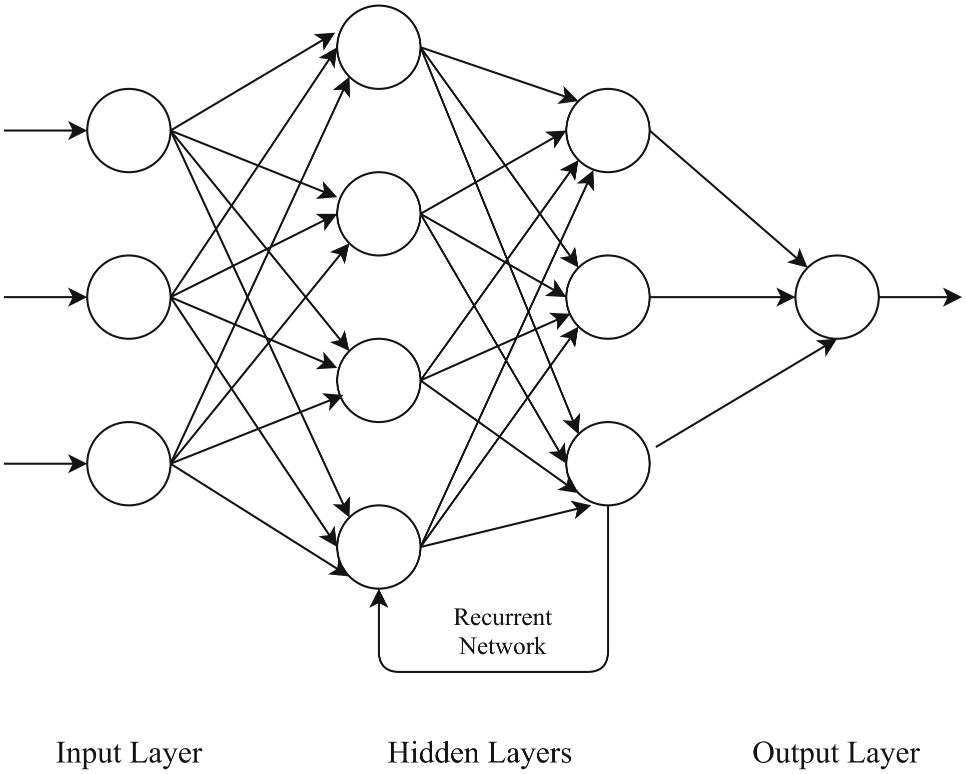


Fig. 11 A simple RNN with two hidden layers

layer with softmax as the activation function. The authors used 33 h of data from 508 speakers. The model achieved a WER of 34 with a dropout rate of 0.5, in combination with CTC and a language model. The research proposed in [171] presented an audio-visual speech recognition system. The used an RNN-transducer, the encoder was composed of five bi-directional LSTM layers, the decoder consisted of two layers of uni-directional LSTM, and the joint space was 640 dimensional. A five-layer CNN model, called V2P, was used to extract features from the input video. The dataset used for testing and training purposes consisted of transcribed YouTube videos of 31,000 h. They received a WER% of 21.5 on the audio-only system and 20.5 on the audio-visual system.

[66] compared the performance of commonly used types of RNN, such as GRU and LSTM, with a simple RNN. They used the TED-LIUM Corpus [131] for testing and training. The model used consisted of one input and output layer with five hidden layers, where the fourth layer is bidirectional. LSTM performed the best with both 500-node architecture and 1000-node architecture, having WER% of 77.55, and 65.04 respectively. In terms of time, RNN was the fastest to train and LSTM took the longest.

15.2.5 Convolutional neural network

A convolutional neural network (CNN) is another commonly used type of ANN. Such networks are generally used for computer vision (CV) tasks, but due to their good feature generation, and discrimination capability, they are also widely applied in the field of natural language processing (NLP).

A common CNN architecture is formed of alternative pooling and convolutional layers, with fully connected layers in the end. A convolutional layer is composed of set neurons, where each neuron acts as a kernel. A convolutional kernel divides the input signal into smaller signals, called receptive fields. A kernel than convolves with the input signal by multiplying itself with the corresponding elements of the receptive field [94]. The following mathematical representation can be used to express the convolutional function:

$$g(x, y) = i(x, y) * h(x, y)$$

Where i(x, y) represents the input signal, h(x, y) represents the applied filter, and g(x, y) is the resultant convolved filter. The same filter, with the same set of weights, is used on all of the receptive fields. This particular feature allows CNN to capture most of the features present in a signal without using a large number of weights.

The focus of a convolution layer is to extract as many features as possible from a signal. But once those features are identified, the exact locations of a particular feature don't matter as long as its approximate location relative to the other features is maintained [94]. The pooling layer performs the job of down-sampling, by retaining only the dominant value in each of the receptive fields, hence, further reducing the size of the input signal. By reducing the signal size, not only the network becomes less complex, but it also reduces the chances of over-fitting and increases generalization.

The research presented in [143] used the combination of CNN and Bidirectional LSTM (BLSTM) to recognize Mandarin speech. The proposed model consisted of four CNN blocks, each with four layers, followed by a layer of BLSTM, and then a fully connected layer. The input signal was batch normalized before being processed by the network. Each of the four CNN blocks consisted of a convolutional layer, followed by a batch normalization layer, then a Rectified Linear Unit (ReLU) activation layer and in the end a max pooling layer. The AISHELL-1 [15] dataset was used for training and testing. The proposed model achieved a WER% of 19.2. [156] used three different types of input, which included MFCC, power spectrum, and raw wave format, were tried with their model. The model proposed in the paper than 12 convolutional layers. The stride of the convolutional model varied with the type of input. Increasing the stride did not affect MFCC whereas it was observed that with the power spectrum and raw waveform the overall stride of the network played a vital role. LibriSpeech [116] was the dataset that was used for training, testing and validating purposes. When used with MFCC the model produced 7.2% WER, with power spectrum as its input the model had 9.4% WER, and lastly with raw waveform it had 10.1% WER.

15.2.6 Fuzzy neural network

Fuzzy neural networks (FNN) is a hybrid technique that incorporates concepts of a fuzzy system in neural networks. Because of the usage of fuzzy systems, a membership function is used to make sure every element is mapped to a proper degree of membership. This membership function proves to be very useful to map speech signals, as they have no clear boundaries [99]. Another advantage of using FNN is that an ANN requires a large amount of data to be effectively trained. But FNN shows better results with even small datasets as they converge during the learning phase [73].

The work proposed in [99] used wavelet transforms, CDHMM, and FNN to recognize fifty words. When compared with a simple CDHMM, the hybrid model was proven to be more successful in a noisy environment; by achieving 15.2% more accuracy. Though, in a clean environment, CDHMM performed better, having a difference of 7.6% in their accuracies.

Adaptive Neuro-Fuzzy Inference System (ANFIS) [73, 170] is a widely used FNN-based speech recognition system, which employs different fuzzy inference techniques to perform classification of data. The work proposed in [73] recognized isolated words of the Persian language. The input dataset was first divided into clusters using SOM and Linear VQ. Once the input was clustered, ANFIS was used to classify the data. The results obtained showed that the ANFIS performed better than a conventional FNN. Another research [170] achieved an accuracy of 85.24% while recognizing Malay digits, using ANFIS.

15.3 Support vector machines

Recently, SVM has been adopted to perform the task of speech recognition. SVM can be implemented independently [52] or as a hybrid model with HMM [133, 152]. SVM constructs a hyperplane as the decision plane, in such a way that the distance between the classes is maximized. The formula given below can be used to calculate the decision surface:

$$f(x_i) = w^T \times \Phi(x_i) + b$$

where w and b are the weight vectors and bias value, $\Phi(x_i)$ is the kernel function. The input feature space is mapped to different higher dimensional feature space using different kernel functions. Using the higher dimensional feature space, the assumption is made that the different classes are linearly separable. One of the commonly used kernel functions is the polynomial function [79], which can be calculated using the following formula:

$$k\left(x_i, x_j\right) = \left(x_i \cdot x_j + 1\right)^d$$

where d is the degree of the polynomial. Another commonly used function is the Gaussian radial basis function [16], which can be mathematically expressed as:

$$k\left(x_i, x_j\right) = \exp\left(-\gamma \left\|x_i - x_j\right\|^2\right)$$

Here the value of ϒ can be either >0 or = $1/2\sigma^2$.

Even though SVMs are considered to be good classifiers, they are not commonly used for ASR. One of the biggest issues with SVM in the context of ASR is that they cannot take variable input, which is usually the case with ASR. Researches performed in [79, 152] discussed the solutions to this problem. SVM also tend to have a high computational cost when classifying more than two classes. Different methods were proposed to tackle the multi-class problem of SVM in [149] [148, 172, 173], but the most commonly used techniques reduces the multi-class problem into a set of binary class SVM. The following two techniques can be used to resolve the multiclass problem:

a- The one-against-all technique
b- The one-against-one technique

15.3.1 The one-against-all technique

In this method, a multi-class SVM is divided into multiple binary class SVM based on their number of classes. The number of binary class SVM is equal to the number of classes in multi-class SVM.

All of the binary class SVM create decision planes between the class corresponding to them and all of the other classes. Different voting techniques are then used for choosing the output for a given input [149, 172]. The one-against-all technique creates a relatively small amount of binary SVM, whereas, it requires a large dataset for the training of each binary SVM. A common problem faced by dividing the multi-class SVM into binary class SVM is the unclassifiable region problem. For this technique, the unclassifiable region problem can be solved by either using continuous decision functions or by implementing fuzzy SVM. A fuzzy SVM uses a membership function in such a way that different input points will contribute differently to the learning process of SVM [36, 59, 90]. Both techniques can be used to evaluate the performance of a system and are comparable to each other. However, implementing continuous decision functions are easier and simpler to implement than fuzzy SVM.

The work presented in [169] shows a comparison between SVM and ANN. The SVM method employed followed the one-against-all technique in combination with the RBF kernel. Whereas the ANN method employed was the MLP network. Both networks were trained to recognize 12 isolated vowels of the Thai language. The SVM network not only took less processing time in the training phase but also achieved a higher accuracy. The accuracy of the SVM network was 87.08%, while the accuracy of the MLP network was 82.72%.

15.3.2 The one-against-one technique

Unlike the one-against-all technique, which creates binary SVMs based on the number of classes, the one-against-one technique creates binary SVMs for all possible pairs. By creating all possible pairs of classes, this technique distinguishes each class from all the other classes. Like the one-against-all technique, there exist many voting techniques for the one-against-one technique as well, which can be employed to choose the output class for a given input [149] [172].

The one-against-one method may create a relatively higher number of binary SVMs, but it requires less training data. This technique also has a lower computational cost, as a classifier, can be ignored if its two corresponding classes are rarely required to be distinguished [79, 133]. The system presented in [9] was used for phoneme classification. MFCC was used to extract fixed-length feature vectors from the input audio. The classifier used in this system was a one-against-one classifier in combination with a majority voting technique. This system was tested on the TIMIT dataset and was compared against HMM. The designed SVM system got an accuracy of 77.6%, which was 4% better than the HMM. The accuracy of HMM was 73.7%.

15.4 Summary

One of the most commonly used methods of classification for speech is HMM. The reason behind its ubiquitous usage is its ability to successfully model the temporal information present in the speech signals. Even though improvements were made in the field of speech recognition using HMM, but the results obtained were not optimal. So, modifications were done to the HMM in the form of ANN and SVM. The techniques of ANN and SVM can be employed independently or as a hybrid model with HMM.

In the past, MLP was the most commonly used type of ANN. Though nowadays, RNN and RBF are applied more frequently. As discussed above, SVM has shown better or at least comparable results to HMM. Though SVM are inherently binary classifiers, different modifications techniques such as the one-against-all and one-against-one can allow them to classify multiple classes successfully.

Table 5 presents the advantages and disadvantages of using different classification techniques.

16 Language model

Advancements in the field of speech recognition have increased the need for language models as speech doesn't necessarily follow rigid grammatical rules. Speaking style of person and their regional and social dialects also affects the input instance. So, a good language model is required to deal with these problems in real-time [40, 41]. A language model consists of a vocabulary set, the search space, and the searching technique. Language models use structural constraints of a language to predict the probabilities of the occurrence of a word, for a specific word sequence. These structural constraints can vary from language to language.

The difference between a classifier and a language model is that a classifier maps speech signals to its closest possible word sequence, whereas, a language model checks the occurrence probability of the word sequence produced by the classifier. A very common example of this is, in American English, the phrases 'recognize speech' and 'wreck a nice beach' sound almost the same, but their meanings are entirely different. These ambiguities are easier to eliminate if a language model is used in combination with a classification model.

16.1 Types of language models

The language models can be divided into two types: static and dynamic.

16.1.1 Static language models

One of the most commonly used techniques of the static language models is the n-gram model. Generally, bigram or trigram language models are used where a trigram model holds more information [41]. The research presented in [98] shows by using a bigram language model in combination with their system. By adding a language model, they achieved an accuracy that was approximately 3% higher than the original accuracy. Another research [2] shows a reduction in the occurrence of out-of-vocabulary (OOV) words by using the n-Gram model.

A major drawback of using such language models is that they cannot adapt if a different speech of a different domain is given as input.

16.1.2 Dynamic language models

Dynamic language models calculate probabilities based on previously analyzed data. Thus, they can easily adapt to different speech domains. This technique is highly useful when transfer learning is being applied; i.e., a pre-trained model of a specific language or speech domain is being used as a basis to train a model for a different language. Some commonly used language models techniques are long-distance n-grams [2], triggers [34, 61], cache models [130], and tree-based models [129].

Table 5 Advantages and disadvantages of the discussed classification models

Classification Models	Advantages	Disadvantages
Hidden Markov Model [154]	• HMM can model the time distribution of a speech signal [79]. • It is very easy to implement. • HMM can process variable-length inputs. • It can model both discrete and continuous sequences.	• The long term dependencies are ignored as it is assumed that the current state is only dependent on the preceding state [25].
Artificial Neural Network [10]	• ANN is very robust. • This algorithm is self-learning and organizing. • It can easily adapt to new environments.	• It can often over train and get stuck in local minima problems.
Multilayer Perceptrons [158]	• They are easy to implement. • MLP doesn't get stuck in local minima as it uses a random probability distribution.	• They can only take fixed input lengths. • It works only for small vocabularies. • The temporal information present in the speech signal is ignored.
Self-Organizing Maps [6]	• No prior information is needed for training a SOM [128]. • Parallel computing can be used to compute it [128]. • It can easily adapt to new sample points.	• The trial and error method is used to find its parameters [128]. • The mapping obtained from the training phase might be lost when applied in the real-world [128].
Radial Basis Functions [104]	• It is very simple to implement [140]. • RBF is very robust [140]. • It can easily discriminate between different words of a vocabulary [140].	• It is invariant to shift in time [17].
Recurrent Neural Network	• RNNs can process variable length inputs easily. • They retain temporal information. • Different time steps can share weights	• They are computationally expensive and require a lot of training time. • They are more suspectable to vanishing and exploding gradients.
Convolutional Neural Network	• It can easily detect important features present in a signal. • It requires less training time.	• It cannot capture temporal features properly. • It cannot deal with variable size input.
Fuzzy Neural Network [51]	• It doesn't get stuck at local minima [99]. • FNN requires a relatively small amount of data for training [73, 99].	• It cannot model the time distribution of speech signals [99].
Support Vector Machines [177]	• SVM is very robust [79]. • It does not face the problems of over-training and being stuck in local minima [79]. • It can take high dimensional vectors as input [79].	• It can only take fixed length inputs [79]. • The computational costs increase with the number of output classes [79].
SVM (One-Against--All) [89]	• It produces a low number of binary SVM.	• This technique requires relatively more training data.
SVM (One-Against--One) [28]	• This technique requires less training data as compared to the one-against-all technique [152].	• It creates a considerable number of binary SVM. • This approach cannot deal with the problem of unclassified regions [161].

Once its chosen which technique of language model is being employed, we also need to choose which decoding search technique needs to be used to find the best result from the specified number of responses.

16.2 Decoding techniques

A decoding technique uses an acoustic model, a language model, and the spoken utterance to find the most likely word sequence. One of the most obvious methods would be to enumerate over all possible outputs to find the most likely. As the number of outputs will grow exponentially with the length of the word, hence this technique can only be employed in tasks with a very small dataset. Various pruning algorithms [41, 80] can be used to remove the low scoring hypothesis, to make searching more efficient.

Viterbi search and n-best search are two of the most commonly used decoding techniques.

16.2.1 Viterbi search

In this approach, all of the hypotheses that are associated with a particular speech utterance are considered and are directly compared with each other. Viterbi search is impractical for even medium-sized projects due to its huge computational cost; hence, Viterbi beam search [7, 65] is mostly used as it considerably reduces the size of the search space. In the Viterbi beam search, only those hypotheses whose likelihood falls under a particular radius are considered.

The research performed in [92] presents an HMM-based speaker-independent ASR. The proposed system used the TIMIT dictionary to generate its word transcription dictionary. The Viterbi algorithm was used to perform sentence decoding on the generated dictionary. Besides the Viterbi algorithm, the word pair technique was also utilized to get a smooth transition between two words. The performance of the system was increased to 92.2% from 60.1% with the usage of both dictionary and word pair techniques.

16.2.2 N-best search

The N-best search [23] is very similar to the Viterbi Search; the main difference between the two algorithms is that where Viterbi Search provides the best hypothesis, the n-best search provides the n-best hypothesis. One major drawback of this algorithm is that the short hypotheses have more chances of being chosen as the long hypotheses are more prone to errors. To overcome this problem, two of the most commonly used methods are the search method [41] and the pruning method [80].

16.3 Summary

From the above discussion, it can be concluded that a language model is required for systems that capture large vocabulary. Nowadays, a lot of research is being performed to optimize language models. And currently, n-gram and n-best search models are the most commonly used methods.

17 Toolkits and online resources

A lot of work has been done on perfecting the task of speech recognition. Some of the work performed over the years is available to us in the form of personal assistant tools such as Cortana and Siri. Even though a lot of research has been performed in this field, but most of the work has not been made available publicly. Table 6 discusses some of the toolkits and online resources that have been made publicly available, as well as a few other commonly used tools.

17.1 Kaldi

Kaldi [179] is an open-source speech recognition tool. This tool was developed in C++ and can easily be deployed on multiple operating systems. Currently, this toolkit only supports the English language. Kaldi can be used to extract features; it can perform classification tasks as well. The features can be extracted using multiple methods, which include the most commonly used MFCC, and cepstral mean and variance normalization (CMVN) and i-vectors. Deep neural networks (DNN) are used to perform the task of classification.

17.2 CMU Sphinx

CMU Sphinx [24] is another open-source speech recognition tool. This tool was developed in Java and can provide pre-trained models for several languages, including English, French, Mandarin, Russian, and German. CMU Sphinx uses MFCC to extract features and uses an HMM-based model to perform the classification task. It also provides an online tool that can be used to create language models for Sphinx.

17.3 Julius

Julius [122] is an open-source speech recognition tool that was originally designed to recognize the Japanese language. Over the years, with the help of different researches, a usable model for the English language was also developed. Julius itself is a language independent decoding program which can be used to create a recognizer for any language as long as an acoustic model and language model is available for that language.

17.4 Hidden Markov model toolkit

Hidden Markov Model Toolkit (HTK) [81] is a portable toolkit that can be used to manipulate and build hidden Markov models. The original purpose behind creating this toolkit was to perform the task of speech recognition, but it can be used for other pattern recognition problems as well.

Table 6 Toolkits and online resources available for speech recognition

Tool Name	Programming Language	Open-Source	Languages Supported
Kaldi	C++	Yes	English
CMU Sphinx	Java	Yes	English, French, Mandarin
Julius	C	Yes	English, Japanese
HTK	C	No	English
RWTH ASR	C++	No	English

17.5 RWTH ASR

RWTH ASR [85] is another toolkit that was developed for speech recognition. This toolkit can be used for both speech recognition and speaker adaptation. It utilizes MFCC and PLP to extract features. The acoustic modeling is performed using GMM. One significant limitation of this toolkit is that it is only available on Linux and macOS.

17.6 Summary

The above discussion determines that the currently available toolkits and online resources rely upon more traditional technologies such as HMM and GMM. A significant shortcoming of these tools is that they can only be employed for particular languages. Therefore, it can be reasoned that more publicly accessible speech recognition tools need to be developed; that can cater to different languages and not be confined to high resource languages only.

18 Concluding summary

Table 7 shows a list of researches done in the field of speech recognition over the past few years.

It can be viewed that researches are more focused on creating optimal large-vocabulary speaker-independent continuous speech ASR. It can also be observed that despite not being the best feature extraction algorithm, as mentioned before, MFCC is still the more favored choice. [37, 175] compared it with other techniques. [37] compared it with PLP and LPCC, where MFCC performed significantly better than the other algorithms. In [175], however, WPT consistently performed better than MFCC. The key difference between the two pieces of research is the size of their datasets; [175] used the dataset comprising of 1000 different speakers, and [37] used the dataset containing the voices of 11 distinct speakers.

For classification, HMM seems to be the most popular choice, particularly HMM hybrid models. [133] performed for the task of recognizing the English language speech, by achieving an accuracy of 94.10%. Another hybrid model, introduced in [156], performed well by obtaining an accuracy of 77.83%. Whereas, [37, 92, 98] showed average results by using a simple HMM. ANN-based researches [101, 132] also showed promising results. Hence, it can be concluded that HMM alone is not enough to achieve the goal of speech recognition. Hybrid models and ANN can help achieve much better results.

Deep learning models, such as the ones presented in [77, 136], also shows promising results. [77] used a CNN-based model in combination with CTC loss and managed to get a WER% of 8.07 using a 6-g language model. They also used the concept of transfer learning to show how their model could be used as a base model for low-resource speech recognition systems. [136] used a RNN-based encoder-decoder model and performed multiple experiments on it.

In the end, the research that used a relatively larger dataset produced better results by using a language model [98]. Therefore, we can presume that using a language model can prove to be beneficial when dealing with large vocabularies.

Table 7 Comparison between different ASRs

Ref No.	Research Name	Language	Feature Extraction Method	Classification Method	Language Model	Accuracy Obtained	Used Dataset
[37]	Speaker independent continuous speech recognition	Romanian	PLP, MFCC, LPC	HMM	–	75.78, 90.41, 63.55	Internal Dataset
[98]	Speaker and context independent phoneme recognition	English	MFCC	CDHMM	Bigram	63.07	TIMIT
[156]	Speaker independent continuous phoneme recognition	English	MFCC	HMM-MLP Hybrid Model	Bigram	77.83	TIMIT
[133]	Speaker independent word recognition	English	MFCC	HMM-SVM Hybrid Model	RM Word-Pair Grammar	94.10	DARPA Resource Management (RM1) corpus [115]
[9]	Speaker independent continuous phoneme recognition	English	MFCC	SVM	–	77.60	TIMIT
[175]	Speaker independent isolated word recognition	English	MFCC, WPT	HMM	–	37.9, 31.22 (Avg. WER%)	SpeechDat2
[150]	Speaker dependent isolated word recognition	Malayalam	DWT	MLP	–	80	Internal Dataset
[93]	Speaker independent word recognition	English	LPC	RBF and pattern matching algorithm hybrid model	–	91	Internal Dataset
[77]	Speaker independent word recognition	English	MFCC	CNN based model with CTC loss	6-g	8.07 (WER%)	LibriSpeech
[136]	Speaker independent word recognition	English	MFCC	RNN-based Encoder-Decoder model	–	6.1 (WER% for Short Utterances), 4.8 (WER% for Long Utterances)	Internal Dataset

19 Conclusion

This survey paper discussed and reviewed different techniques and approaches that are used to perform the task of speech recognition. Based on the discussion on the basic architecture of an ASR, it is concluded that an ASR is dependent on three modules: feature extraction module, classification module, and the language model. Hence, different feature extraction methods, their advantages, and disadvantages, as well as their basic structure, were also highlighted. Similarly, from the analysis of classification models, it is inferred that HMM performed the best. Many recently employed techniques and their results are also reviewed. In the end, the last module of the speech recognition system, the language model, was examined. It is concluded that the addition of a language model; can greatly affect the accuracy of an ASR. Even though only sub-optimal methods are currently being used to create language models, further research in this field will prove to be beneficial for the task of speech recognition.

Compliance with ethical standards

Conflict of interest None.

Declarations Not applicable.

References

1. Abdulla W H, Kasabov N (1999) The concepts of hidden Markov model in speech recognition.
2. Abe S (2003) Analysis of multiclass support vector machines. Thyroid 21(3):3772
3. Alkhaldi W, Fakhr W, Hamdy N (2002) Automatic speech/speaker recognition in noisy environments using wavelet transform, The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002., Tulsa, OK, USA, pp. I-463, doi: https://doi.org/10.1109/MWSCAS.2002.1187258.
4. Anusuya MA, Katti SK (2011) Front end analysis of speech recognition: a review. Int J Speech Technol 14(2):99–145
5. Anusuya MA, Katti SK (2011) Comparison of different speech feature extraction techniques with and without wavelet transform to Kannada speech recognition. Int J Comput Appl 26(4):19–24
6. Atmaja BT, Akagi M (2020) Deep multilayer Perceptrons for dimensional speech emotion recognition. arXiv preprint arXiv:2004.02355.
7. Bahl LR, Brown PF, de Souza PV, Mercer RL (1989) A tree-based statistical language model for natural language speech recognition. IEEE Trans Acoust Speech Signal Process 37(7):1001–1008
8. Barker J, Watanabe S, Vincent E, Trmal J (2018) The fifth'CHiME'speech separation and recognition challenge: dataset, task and baselines. arXiv preprint arXiv:1803.10609.
9. Batuwita R, Palade V (2010) FSVM-CIL: fuzzy support vector machines for class imbalance learning. IEEE Trans Fuzzy Syst 18(3):558–571
10. Baum LE, Eagon JA (1967) An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bull Am Math Soc 73(3):360–363
11. Bernardo JM, Bayarri MJ, Berger JO, Dawid AP, Heckerman D, Smith AFM, West M (2007) Generative or discriminative? Getting the best of both worlds. Bayesian stat 8(3):3–24
12. Besacier L, Barnard E, Karpov A, Schultz T (2014) Automatic speech recognition for under-resourced languages: a survey. Speech Comm 56:85–100
13. Birkenes O, Matsui T, Tanabe K, Siniscalchi SM, Myrvoll TA, Johnsen MH (2009) Penalized logistic regression with HMM log-likelihood regressors for speech recognition. IEEE Trans Audio Speech Lang Process 18(6):1440–1454
14. Bourlard H A, Morgan N (2012). Connectionist speech recognition: a hybrid approach (Vol. 247). Springer Science & Business Media.

15. Bu H, Du J, Na X, Wu B, Zheng H (2017). Aishell-1: an open-source mandarin speech corpus and a speech recognition baseline. In 2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA) (pp. 1-5). IEEE.
16. Busso C, Bulut M, Lee CC, Kazemzadeh A, Mower E, Kim S, Chang JN, Lee S, Narayanan SS (2008) IEMOCAP: interactive emotional dyadic motion capture database. Lang Resour Eval 42(4):335–359
17. Campos MM, Carpenter GA (1998) WSOM: building adaptive wavelets with self-organizing maps. In 1998 IEEE international joint conference on neural networks proceedings. IEEE world congress on computational intelligence (cat. No. 98CH36227) (Vol. 1, pp. 763-767). IEEE
18. Chan W, Jaitly N, Le Q, Vinyals O (2016) Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4960-4964). IEEE.
19. Chang T H, Luo Z Q, Deng L, Chi C Y (2008) A convex optimization method for joint mean and variance parameter estimation of large-margin CDHMM. In 2008 IEEE international conference on acoustics, speech and signal processing (pp. 4053-4056). IEEE.
20. Chen C P, Bilmes J, Ellis D P (2005) Speech feature smoothing for robust ASR. In proceedings.(ICASSP'05). IEEE international conference on acoustics, speech, and signal processing, 2005. (Vol. 1, pp. I-525). IEEE.
21. Cheng O, Abdulla W, Salcic Z (2005) Performance evaluation of front-end processing for speech recognition systems. The University of Auckland.
22. Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ..., Jaitly, N. (2018) State-of-the-art speech recognition with sequence-to-sequence models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4774–4778). IEEE.
23. Chow Y, Dunham M, Kimball O, Krasner M, Kubala G, Makhoul J, ..., Schwartz R (1987) BYBLOS: The BBN continuous speech recognition system. In ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 12, pp. 89–92). IEEE
24. Chow YL, Schwartz R (1989) The n-best algorithm: an efficient procedure for finding top n sentence hypotheses. In proceedings of the workshop on speech and natural language (pp. 199-202). Association for Computational Linguistics
25. Clarkson P, Moreno PJ (1999) On the use of support vector machines for phonetic classification. In 1999 IEEE international conference on acoustics, speech, and signal processing. Proceedings. ICASSP99 (cat. No. 99CH36258) (Vol. 2, pp. 585-588). IEEE
26. Coifman R R, Meyer Y, Wickerhauser V (1992) Wavelet analysis and signal processing. In In Wavelets and their applications.
27. Collobert R, Puhrsch C, Synnaeve G (2016) Wav2letter: an end-to-end convnet-based speech recognition system. arXiv preprint arXiv:1609.03193.
28. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
29. Crouse MS, Nowak RD, Baraniuk RG (1998) Wavelet-based statistical signal processing using hidden Markov models. IEEE Trans Signal Process 46(4):886–902
30. Cutajar M, Gatt E, Grech I, Casha O, Micallef J (2013) Comparative study of automatic speech recognition techniques. IET Signal Proc 7(1):25–46
31. Cutajar M, Gatt E, Micallef J, Grech I, Casha O (2010) Digital hardware implementation of self-organising maps. In Melecon 2010-2010 15th IEEE Mediterranean Electrotechnical conference (pp. 1123-1128). IEEE
32. Dansena D K, Rathore Y A Survey Paper on Automatic Speech Recognition by Machine
33. Davis KH, Biddulph R, Balashek S (1952) Automatic recognition of spoken digits. J Acoust Soc Am 24(6):637–642
34. Deshmukh N, Picone J (1995) Methodologies for language modeling and search in continuous speech recognition. In proceedings IEEE Southeastcon'95. Visualize the future (pp. 192-198). IEEE
35. Du X P, He P L (2006) The clustering solution of speech recognition models with SOM. In international symposium on neural networks (pp. 150-157). Springer, Berlin, Heidelberg.
36. Duan KB, Keerthi SS (2005) Which is the best multiclass SVM method? An empirical study. In international workshop on multiple classifier systems (pp. 278-285). Springer, Berlin, Heidelberg
37. Dumitru C O, Gavat I (2006) A comparative study of feature extraction methods applied to continuous speech recognition in romanian language. In proceedings ELMAR 2006 (pp. 115-118). IEEE.
38. Fontaine V, Ris C, Leich H (1996) Nonlinear discriminant analysis with neural networks for speech recognition. In 1996 8th European signal processing conference (EUSIPCO 1996) (pp. 1-4). IEEE.
39. Forgie JW, Forgie CD (1959) Results obtained from a vowel recognition computer program. J Acoust Soc Am 31(11):1480–1489
40. Forsberg M (2003) Why is speech recognition difficult. Chalmers University of Technology.

41. Friedman JH (1996) Another approach to polychotomous classification. Statistics Department, Stanford University, Technical Report
42. Gaikwad SK, Gawali BW, Yannawar P (2010) A review on speech recognition technique. Int J Comput Appl 10(3):16–24
43. Gamulkiewicz B, Weeks M (2003) Wavelet based speech recognition. In 2003 46th Midwest symposium on circuits and systems (Vol. 2, pp. 678-681). IEEE.
44. Ganapathy S, Thomas S, Hermansky H (2009) Modulation frequency features for phoneme recognition in noisy speech. J Acoust Soc Am 125(1):EL8–EL12
45. Garofolo JS (1993) TIMIT acoustic phonetic continuous speech corpus. Linguist Data Consortium 1993
46. Graves A, Fernández S, Gomez F, Schmidhuber J (2006) Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In proceedings of the 23rd international conference on machine learning (pp. 369-376)
47. Gupta M, Gilbert A (2001) Robust speech recognition using wavelet coefficient features. In IEEE workshop on automatic speech recognition and understanding, 2001. ASRU'01. (pp. 445-448). IEEE.
48. Hai J, Joo E M (2003) Improved linear predictive coding method for speech recognition. In fourth international conference on information, communications and signal processing, 2003 and the fourth Pacific rim conference on multimedia. Proceedings of the 2003 joint (Vol. 3, pp. 1614-1618). IEEE.
49. Halabi N (2016) Modern standard arabic phonetics for speech synthesis (Doctoral dissertation, University of Southampton).
50. Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, ..., Ng A Y (2014) Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.
51. Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. J Geophys Res 76(8): 1905–1915
52. Helmi N, Helmi BH (2008) Speech recognition with fuzzy neural network for discrete words. In 2008 fourth international conference on natural computation (Vol. 7, pp. 265-269). IEEE
53. Hemakumar G, Punitha P (2013) Speech recognition technology: a survey on Indian languages. Int J Inf Sci Intell Syst 2(4):1–38
54. Hennebert J, Hasler M, Dedieu H (1994) Neural networks in speech recognition. Department of Electrical Engineering, Swiss Federal Institute of Technology, 1015.
55. Hermansky H (1990) Perceptual linear predictive (PLP) analysis of speech. The. J Acoust Soc Am 87(4): 1738–1752
56. Hermansky H, Morgan N (1994) RASTA processing of speech. IEEE Trans Speech Audio Process 2(4): 578–589
57. Hermansky H, Morgan N, Bayya A, Kohn P (1991) RASTA-PLP speech analysis. In Proc. IEEE Int'l Conf. Acoustics, speech and signal processing (Vol. 1, pp. 121-124).
58. Hou X (2009) Noise robust speech recognition based on wavelet-RBF neural network. In PIAGENG 2009: intelligent information, control, and communication Technology for Agricultural Engineering (Vol. 7490, p. 74902O). International Society for Optics and Photonics
59. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425
60. Hu X, Zhan L, Xue Y, Zhou W, Zhang L (2011) Spoken arabic digits recognition based on wavelet neural networks. In 2011 IEEE international conference on systems, man, and cybernetics (pp. 1481-1485). IEEE.
61. Huang X, Alleva F, Hon HW, Hwang MY, Lee KF, Rosenfeld R (1993) The SPHINX-II speech recognition system: an overview. Comput Speech Lang 7(2):137–148
62. Huang X, Baker J, Reddy R (2014) A historical perspective of speech recognition. Commun ACM 57(1): 94–103
63. Hung JW, Fan HT (2009) Subband feature statistics normalization techniques based on a discrete wavelet transform for robust speech recognition. IEEE Signal Process Lett 16(9):806–809
64. Hunt A, Favero R (1994) Using principal component analysis with wavelets in speech recognition. In SST Conf., ASSTA Inc., Perth (pp. 296-301).
65. Illina I, Gong Y (1996) Improvement in N-best search for continuous speech recognition. In proceeding of fourth international conference on spoken language processing. ICSLP'96 (Vol. 4, pp. 2147-2150). IEEE
66. Islam J, Mubassira M, Islam MR, Das AK (2019) A speech recognition system for Bengali language using recurrent neural network. In 2019 IEEE 4th international conference on computer and communication systems (ICCCS) (pp. 73-76). IEEE
67. Jiang H, Li X, Liu C (2006) Large margin hidden Markov models for speech recognition. IEEE Trans Audio Speech Lang Process 14(5):1584–1595
68. Juang BH, Rabiner LR (1991) Hidden Markov models for speech recognition. Technometrics 33(3):251–272

69. Juang B H, Rabiner L R (2005) Automatic speech recognition–a brief history of the technology development. Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara, 1, 67.
70. Jung S, Son J, Bae K (2004) Feature extraction based on wavelet domain hidden Markov tree model for robust speech recognition. In Australasian joint conference on artificial intelligence (pp. 1154-1159). Springer, Berlin, Heidelberg.
71. Kaur P, Singh P, Garg V (2012) Speech recognition system; challenges and techniques. Int J Comput Sci Inf Technol 3(3):3989–3992
72. Kesarkar M P (2003) Feature extraction for speech recognition. Electronic systems, EE. Dept., IIT Bombay.
73. Khan A, Sohail A, Zahoora U, Qureshi AS (2020) A survey of the recent architectures of deep convolutional neural networks. Artif Intell Rev, 1–62
74. Köhn A, Stegen F, Baumann T (2016) Mining the spoken wikipedia for speech data and beyond. In proceedings of the tenth international conference on language resources and evaluation (LREC'16) (pp. 4644-4647).
75. Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43(1):59–69
76. Korba M C A, Messadeg D, Djemili R, Bourouba H (2008) Robust speech recognition using perceptual wavelet denoising and mel-frequency product spectrum cepstral coefficient features. Informatica, 32(3).
77. Kriman S, Beliaev S, Ginsburg B, Huang J, Kuchaiev O, Lavrukhin V, ..., Zhang Y (2020) Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6124–6128). IEEE
78. Krishnan VV, Anto PB (2009) Features of wavelet packet decomposition and discrete wavelet transform for malayalam speech recognition. Int J Recent Trends Eng 1(2):93
79. Krüger SE, Schafföner M, Katz M, Andelic E, Wendemuth A (2005) Speech recognition with support vector machines in a hybrid system. In Ninth European Conference on Speech Communication and Technology
80. Kupiec J (1989) Probabilistic models of short and long distance word dependencies in running text. In Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989
81. Lamere P, Kwok P, Gouvea E, Raj B, Singh R, Walker W, ..., Wolf P (2003) The CMU SPHINX-4 speech recognition system. In IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong (Vol. 1, pp. 2–5)
82. Lawrence R (2008) Fundamentals of speech recognition. Pearson Education India.
83. Lazli L, Sellami M (2003) Connectionist probability estimators in HMM arabic speech recognition using fuzzy logic. In international workshop on machine learning and data Mining in Pattern Recognition (pp. 379-388). Springer, Berlin, Heidelberg.
84. Lee J Y, Hung J W (2011) Exploiting principal component analysis in modulation spectrum enhancement for robust speech recognition. In 2011 eighth international conference on fuzzy systems and knowledge discovery (FSKD) (Vol. 3, pp. 1947-1951). IEEE.
85. Lee A, Kawahara T, Shikano K (2001) Julius—an open source real-time large vocabulary recognition engine
86. Lekshmi KR, Elizabeth S (2016) Automatic speech recognition using different neural network architectures – a survey. Int J Comput Sci Inf Technol 7(6):2422–2427
87. Leung K F, Leung F H, Lam H K, Tam P K S (2003) Recognition of speech commands using a modified neural fuzzy network and an improved GA. In the 12th IEEE international conference on fuzzy systems, 2003. FUZZ'03. (Vol. 1, pp. 190-195). IEEE.
88. Li T F, Chang S C (2007) Speech recognition of mandarin syllables using both linear predict coding cepstra and Mel frequency cepstra. In ROCLING 2007 poster papers (pp. 379-390).
89. Lin CT (1996) Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. Prentice hall PTR
90. Lin CF, Wang SD (2002) Fuzzy support vector machines. IEEE Trans Neural Netw 13(2):464–471
91. Liu X (2009) A new wavelet threshold denoising algorithm in speech recognition. In 2009 Asia-Pacific conference on information processing (Vol. 2, pp. 310-313). IEEE.
92. Lowerre BT (1976) The HARPY speech recognition system. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE
93. Maheswari NU, Kabilan AP, Venkatesh R (2010) A hybrid model of neural network approach for speaker independent word recognition. Int J Comput Theory Eng 2(6):912

94. Makino T, Liao H, Assael Y, Shillingford B, Garcia B, Braga O, Siohan O (2019) Recurrent neural network transducer for audio-visual speech recognition. In 2019 IEEE automatic speech recognition and understanding workshop (ASRU) (pp. 905-912). IEEE
95. Malekzadeh S, Gholizadeh M H, Razavi S N (2018). Persian vowel recognition with MFCC and ANN on PCVC speech dataset. arXiv preprint arXiv:1812.06953.
96. Mallat SG (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans Pattern Anal Mach Intell 11(7):674–693
97. Mehla R, Aggarwal R (2014) Automatic speech recognition: a survey. Int J Adv Res Comput Sci Electron Eng (IJARCSEE) 3(1):45–53
98. Messaoud Z B, Hamida A B (2010) CDHMM parameters selection for speaker-independent phone recognition in continuous speech system. In MELECON 2010-2010 15th IEEE Mediterranean Electrotechnical conference (pp. 253-258). IEEE.
99. Meyer Y (1993) Wavelets: Algorithms and Applications, SIAM, Philadelphia, 1993. MR 95f, 94005.
100. Milone DH, Di Persia LE (2008) Learning hidden Markov models with hidden Markov trees as observation distributions. Inteligencia artificial. Revista Iberoamericana de Inteligencia Artificial 12(37): 7–13
101. Modic R, Lindberg B, Petek B (2003) Comparative wavelet and mfcc speech recognition experiments on the slovenian and english speechdat2. In ISCA tutorial and research workshop on non-linear speech processing
102. Mohamadpour M, Farokhi F (2009) A new approach for Persian speech recognition. In 2009 IEEE international advance computing conference (pp. 153-158). IEEE
103. Molau S, Pitz M, Schluter R, Ney H (2001) Computing mel-frequency cepstral coefficients on the power spectrum. In 2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (cat. No. 01CH37221) (Vol. 1, pp. 73-76). IEEE.
104. Morgan N, Bourlard H (1990). Continuous speech recognition using multilayer perceptrons with hidden Markov models. In international conference on acoustics, speech, and signal processing (pp. 413-416). IEEE
105. Mporas I, Ganchev T, Siafarikas M, Fakotakis N (2007) Comparison of speech features on the speech recognition task. J Comput Sci 3(8):608–616
106. Muller D N, De Siqueira M L, Navaux P O A (2006) A connectionist approach to speech understanding. In the 2006 IEEE international joint conference on neural network proceedings (pp. 3790-3797). IEEE.
107. Nataraj K S, Pandey P C, Shah M S (2011) Improving the consistency of vocal tract shape estimation. In 2011 National Conference on communications (NCC) (pp. 1-5). IEEE.
108. Nehe NS, Holambe RS (2009) New feature extraction techniques for Marathi digit recognition. Int J Recent Trends Eng 2(2):22
109. Nehe NS, Holambe RS (2012) DWT and LPC based feature extraction methods for isolated word recognition. EURASIP J Audio Speech Music Process 2012(1):7
110. Nguyen P, Heigold G, Zweig G (2010) Speech recognition with flat direct models. IEEE J Sel Top Sign Proces 4(6):994–1006
111. Nouza J, Zdansky J, Cerva P (2010) System for automatic collection, annotation and indexing of Czech broadcast speech with full-text search. In MELECON 2010–2010 15th IEEE Mediterranean Electrotechnical Conference (pp. 202–205). IEEE
112. O’Shaughnessy D (2008) Automatic speech recognition: history, methods and challenges. Pattern Recogn 41(10):2965–2979
113. O'Shaughnessy D (1988) Linear predictive coding. IEEE potentials 7(1):29–32
114. O'Shaughnessy D (2003) Interacting with computers by voice: automatic speech recognition and synthesis. Proc IEEE 91(9):1272–1305
115. Pallett DS, Fiscus JG, Garofolo JS (1990) DARPA resource management. In speech and natural language: proceedings of a workshop held at Hidden Valley, Pennsylvania, June 24-27, 1990 (p. 298). Morgan Kaufmann pub
116. Panayotov V, Chen G, Povey D, Khudanpur S (2015) Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5206-5210). IEEE.
117. Paul AK, Das D, Kamal MM (2009) Bangla speech recognition system using LPC and ANN. In 2009 seventh international conference on advances in pattern recognition (pp. 171-174). IEEE
118. Paulson LD (2006) Speech recognition moves from software to hardware. Computer 39(11):15–18
119. Picone JW (1993) Signal modeling techniques in speech recognition. Proc IEEE 81(9):1215–1247
120. Ping Z, Li-Zhen T, Dong-Feng X (2009) Speech recognition algorithm of parallel subband HMM based on wavelet analysis and neural network. Inf Technol J 8(5):796–800
121. Polikar R (1996) The wavelet tutorial.

122. Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, ..., Silovsky J (2011) The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Process Soc
123. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
124. Rabiner L, Juang B H (1993) Fundamental of speech recognition prentice-hall international.
125. Rabiner L, Levinson S (1981) Isolated and connected word recognition-theory and selected applications. IEEE Trans Commun 29(5):621–659
126. Radha V, Vimala C (2012) A review on speech recognition challenges and approaches. Doaj Org 2(1):1–7
127. Ranjan S (2010) A discrete wavelet transform based approach to Hindi speech recognition. In 2010 international conference on signal acquisition and processing (pp. 345-348). IEEE.
128. Rosenblatt F (1961). Principles of neurodynamics. Perceptrons and the theory of brain mechanisms (no. VG-1196-G-8). Cornell aeronautical lab Inc Buffalo NY
129. Rosenfeld R (1994) A hybrid approach to adaptive statistical language modeling. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE
130. Rosenfeld R, Huang X (1992) Improvements in stochastic language modeling. In Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992
131. Rousseau A, Deléglise P, Esteve Y (2012) TED-LIUM: an automatic speech recognition dedicated corpus. In LREC (pp. 125-129).
132. Rybach D, Gollan C, Heigold G, Hoffmeister B, Lööf J, Schlüter R, Ney H (2009) The RWTH Aachen University open source speech recognition system. In Tenth Annual Conference of the International Speech Communication Association
133. Sabah R, Ainon RN (2009) Isolated digit speech recognition in Malay language using neuro-fuzzy approach. In 2009 third Asia international conference on Modelling & Simulation (pp. 336-340). IEEE
134. Saeed TR, Salman J, Ali AH (2019) Classification improvement of spoken arabic language based on radial basis function. Int J Electr Comput Eng 9(1):2088–8708
135. Saha G, Chakroborty S, Senapati S (2005) A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. In proceedings of the NCC (pp. 56-61).
136. Sainath TN, Pang R, Rybach D, He Y, Prabhavalkar R, Li W, ..., McGraw I (2019) Two-pass end-to-end speech recognition. arXiv preprint arXiv:1908.10992
137. Sak H, Senior A, Rao K, Beaufays F (2015) Fast and accurate recurrent neural network acoustic models for speech recognition. arXiv preprint arXiv:1507.06947.
138. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26(1):43–49
139. Sárosi G, Mozsáry M, Mihajlik P, Fegyó T (2011) Comparison of feature extraction methods for speech recognition in noise-free and in traffic noise environment. In 2011 6th conference on speech technology and human-computer dialogue (SpeD) (pp. 1-8). IEEE.
140. Sayers C (1991). Self organizing feature maps and their applications to robotics
141. Sha F, Saul LK (2007) Large margin hidden Markov models for automatic speech recognition. In advances in neural information processing systems (pp. 1249-1256)
142. Shanthi TS, Lingam C (2013) Review of feature extraction techniques in automatic speech recognition. Int J Sci Eng Technol 2(6):479–484
143. Shewalkar A, Nyavanandi D, Ludwig SA (2019) Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. J Artif Intel Soft Comput Res 9(4):235–245
144. Singh MT, Fayjie AR, Kachari B (2015) A survey report on speech recognition system. Int J Comput Appl 121(11)
145. Sivaram GS, Hermansky H (2011) Multilayer perceptron with sparse hidden outputs for phoneme recognition. In 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5336-5339). IEEE
146. Sivaram GS, Hermansky H (2011) Sparse multilayer perceptron for phoneme recognition. IEEE Trans Audio Speech Lang Process 20(1):23–29
147. Smaragdis P, Radhakrishnan R, Wilson K W (2009) Context extraction through audio signal analysis. In multimedia content analysis (pp. 1–34). Springer, Boston, MA
148. Solera-Ureña R, Padrell-Sendra J, Martín-Iglesias D, Gallardo-Antolín A, Peláez-Moreno C, Díaz-de-María F (2007) Svms for automatic speech recognition: a survey. In Progress in nonlinear speech processing (pp. 190–216). Springer, Berlin, Heidelberg
149. Sonkamble BA, Doye DD, Sonkamble S, PICT P, MMCOE P (2009) An efficient use of support vector machines for speech signal classification. In Proc eighth WSEAS Int Conf computational intelligence., man-machine systems and cybernetics (pp. 117-120)

150. Sukumar AR, Shah AF, Anto PB (2010) Isolated question words recognition from speech queries by using artificial neural networks. In 2010 second international conference on computing, communication and networking technologies (pp. 1-4). IEEE.
151. Tang X (2009) Hybrid hidden Markov model and artificial neural network for automatic speech recognition. In 2009 Pacific-Asia conference on circuits, communications and systems (pp. 682-685). IEEE.
152. Tang H, Meng CH, Lee LS (2010) An initial attempt for phoneme recognition using structured support vector machine (SVM). In 2010 IEEE international conference on acoustics, speech and signal processing (pp. 4926-4929). IEEE
153. Tavanaei A, Manzuri M T, Sameti H (2011) Mel-scaled discrete wavelet transform and dynamic features for the Persian phoneme recognition. In 2011 international symposium on artificial intelligence and signal processing (AISP) (pp. 138-140). IEEE.
154. Thubthong N, Kijsirikul B (2001) Support vector machines for Thai phoneme recognition. Int J Uncertainty Fuzziness Knowledge Based Syst 9(06):803–813
155. Toshniwal S, Sainath T N, Weiss R J, Li B, Moreno P, Weinstein E, Rao K (2018) Multilingual speech recognition with a single end-to-end model. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4904-4908). IEEE.
156. Tóth L (2011) A hierarchical, context-dependent neural network architecture for improved phone recognition. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5040–5043). IEEE
157. Trentin E, Gori M (2001) A survey of hybrid ANN/HMM models for automatic speech recognition. Neurocomputing 37(1–4):91–126
158. Trentin E, Gori M (2003) Robust combination of neural networks and hidden Markov models for speech recognition. IEEE Trans Neural Netw 14(6):1519–1531
159. Umarani SD, Raviram P, Wahidabanu RSD (2009) Implementation of HMM and radial basis function for speech recognition. In 2009 international conference on Intelligent Agent & Multi-Agent Systems (pp. 1-4). IEEE
160. Vadwala AY, Suthar KA, Karmakar YA, Pandya N (2017) Survey paper on different speech recognition algorithm: challenges and techniques. Int J Comput Appl 175(1):31–36
161. Vapnik V (2013) The nature of statistical learning theory. Springer science & business media
162. Veaux C, Yamagishi J, MacDonald K (2016) Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.
163. Veisi H, Sameti H (2011) The integration of principal component analysis and cepstral mean subtraction in parallel model combination for robust speech recognition. Digital Signal Process 21(1):36–53
164. Velichko VM, Zagoruyko NG (1970) Automatic recognition of 200 words. Int J Man Mach Stud 2(3): 223–234
165. Venkateswarlu R L K, Kumari R V (2011) Novel approach for speech recognition by using self—organized maps. In 2011 international conference on emerging trends in networks and computer communications (ETNCC) (pp. 215-222). IEEE.
166. Venkateswarlu RLK, Kumari RV, Jayasri GV (2011) Speech recognition using radial basis function neural network. In 2011 3rd international conference on electronics computer technology (Vol. 3, pp. 441-445). IEEE
167. Walker SL, Foo SY (2003) Optimal wavelets for speech signal representations. J Syst Cybern Inform 1(4): 44–46
168. Wang Y, Han K, Wang D (2012) Exploring monaural features for classification-based speech segregation. IEEE Trans Audio Speech Lang Process 21(2):270–279
169. Wang Y, Wang S, Lai KK (2005) A new fuzzy support vector machine to evaluate credit risk. IEEE Trans Fuzzy Syst 13(6):820–831
170. Wang D, Wang X, Lv S (2019) End-to-end mandarin speech recognition combining CNN and BLSTM. Symmetry 11(5):644
171. Wang B, Yin Y, Lin H (2020) Attention-based transducer for online speech recognition. arXiv preprint arXiv:2005.08497
172. Weston J, Watkins C (1998) Multi-class support vector machines (pp. 98-04). Technical report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, may
173. Weston J, Watkins C (1999) Support vector machines for multi-class pattern recognition. In Esann (Vol. 99, pp. 219-224)
174. Wijoyo S, Wijoyo S (2011) Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot. In proceedings of 2011 international conference on information and electronics engineering (ICIEE 2011) (pp. 28-29).

175. Woodland PC, Leggetter CJ, Odell JJ, Valtchev V, Young SJ (1995) The 1994 HTK large vocabulary speech recognition system. In 1995 international conference on acoustics, speech, and signal processing (Vol. 1, pp. 73-76). IEEE
176. Yegnanarayana B, Veldhuis RN (1998) Extraction of vocal-tract system characteristics from speech signals. IEEE Trans Speech Audio Process 6(4):313–327
177. Yu H, Xie T, Paszczynski S, Wilamowski BM (2011) Advantages of radial basis function networks for dynamic system design. IEEE Trans Ind Electron 58(12):5438–5450
178. Zamani B, Akbari A, Nasersharif B, Jalalvand A (2011) Optimized discriminative transformations for speech features based on minimum classification error. Pattern Recogn Lett 32(7):948–955
179. Zhao Y, Wakita H, Zhuang X (1991) An HMM based speaker-independent continuous speech recognition system with experiments on the TIMIT DATABASE. In acoustics, speech, and signal processing, IEEE international conference on (pp. 333-336). IEEE computer society

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mishaim Malik completed her M.Phil (Computer Science) degree from Punjab University College of Information Technology (PUCIT), Lahore, Pakistan & is currently working with a private software firm. (email: mishaimmalik30@gmail.com)

Muhammad Kamran Malik , a Ph.D. (Computer Science), is currently with the Faculty of Punjab University College of Information Technology (PUCIT), Lahore, Pakistan. (email: kamran.malik@pucit.edu.pk)

Khawar Mehmood is currently pursuing the Ph.D. degree in computer science with the School of Engineering and Information Technology, University of New South Wales (UNSW) Canberra at ADFA, Australia (e-mail: k.mehmood@unsw.edu.au)

Imran Makhdoom is with the Faculty of Engineering and IT, University of Technology Sydney, Australia (e-mail: imran.makhdoom@uts.edu.au)