Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Bridging communication between hearing and visually impaired individuals via speech recognition and synthesis

Dawit Teklu Weldeslasie

dawit.tekulu@gmail.com

Aksum University

Mehamed Ahmed Abdurrahman

Aksum University

Yonas Desta Sheshegu

Aksum University

Research Article

Keywords: Assistive technology, multimodal communication, signal processing, HCI, deaf, blind

Posted Date: August 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10099226/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Bridging communication between hearing and visually impaired individuals via speech recognition and synthesis

Dawit Teklu Weldeslasie[1*] https://orcid.org/0009-0000-5388-5197, Mehamed Ahmed Abdurrahman[2] https://orcid.org/0000-0001-8643-3744, Yonas Desta Sheshegu [3]

[1]Aksum University, Aksum, Ethiopia, Department of Information Technology
[2]Aksum University, Aksum, Ethiopia, Department of Computer Science
[3]Aksum University, Aksum, Ethiopia, Department of Electronic Communication Engineering

*Corresponding author(s). E-mail(s): dawit.tekulu@gmail.com
Contributing authors: mohammeda128@gmail.com, destayonas2@gmail.com

Abstract

Effective communication remains a profound challenge for individuals with sensory disabilities, often leading to social isolation and limited opportunities for societal integration. To address these barriers, this paper presents a Speech Recognition and Synthesis Tool (SRST) designed to facilitate bidirectional communication specifically between hearing and visually impaired individuals. Built as a software-only solution, the system requires no specialized external hardware beyond a standard computer, ensuring high accessibility, cost-effectiveness and operational efficiency. This study was focused on acoustic signal processing via pattern recognition techniques for isolated words and short, trained conversational sentences, bypassing higher-level multimodal linguistic structures such as grammar, semantics, and pragmatics. Similarly, the text-to-speech component operates on rule-based algorithmic and dictionary-driven approaches to synthesize text into audible speech without contextual semantic analysis. By engineering a dual-window Graphical User Interface (GUI) within the MATLAB environment, a functional communication bridge was established. The system robustly transcribes real-time acoustic input into high-contrast digital text for hearing-impaired users while concurrently converting textual string inputs into synthesized, intelligible verbal waveforms via an optimized diphone concatenation engine for visually impaired individuals. Experimental validation indicates that the prototype meets its design objectives, achieving an average operational efficiency of approximately 90%. The architectural design of the system offers straightforward scalability, allowing the core vocabulary and phonetic database to be expanded using the established parameterization and modeling protocols. The prototype developed in this research demonstrates a baseline framework for language-independent speech processing, serving as a foundational architecture for more advanced speech-driven interfaces. To build upon the results of this study, we recommend integrating a machine translation layer between the ASR and TTS pipelines would allow the system to function as an automated language translator, expanding its utility to cross-lingual assistive communication.

Key words: Assistive technology, multimodal communication, signal processing, HCI, deaf, blind

1. Introduction

Communication is the fundamental process of transferring information, thoughts, opinions, or ideas between entities via speech, writing, or somatic signs. In the modern era, communication technologies serve as the primary catalyst for globalization, effectively interconnecting individuals across the planet. So, the evolution of communication paradigms can be categorized into three revolutionary stages [1].

The first was communication revolution marked by the dawn of written communication through pictographs. While these early records were carved into stone and lacked mobility, they established the foundation for permanent information storage. The second was communication revolution defined by the

transition of writing onto lightweight media such as papyrus, clay, wax, and paper. This era introduced standardized alphabets, allowing uniform language dispersion across vast geographic distances. The third is communication revolution characterized by the contemporary digital era, where information is transmitted instantaneously across the globe via controlled electromagnetic waves and electronic signals. Human spoken and written languages within these eras are structured systems composed of symbols (lexemes) governed by syntactic rules (grammars). For the vast majority of the population, leveraging these systems through modern channels, such as cellular networks, electronic mail, instant messaging, and video conferencing is seamless [2].

However, exchanging information remains a profound challenge for individuals with sensory disabilities. For instance, an individual with severe hearing loss or profound deafness faces immense barriers using traditional voice-based telecommunications due to an inability to hear auditory responses. Conversely, a visually impaired or completely blind individual can easily navigate auditory channels but faces significant obstacles interacting with standard digital interfaces due to the inability to read visual displays.

Consequently, direct, unassisted communication between a deaf individual and a blind individual is exceptionally difficult without an external mediator. While this mediator can be human or technological, existing assistive solutions in the current market are often scarce, prohibitively expensive, and highly inaccessible to the average user.

Long before the advent of digital signal processing, researchers attempted to construct mechanical systems capable of replicating human speech. Early historical accounts from the 11th and 13th centuries describe rudimentary "speaking heads" attributed to figures such as Gerbert of Aurillac, Albertus Magnus, and Roger Bacon [3]. The scientific foundation of speech synthesis began to mature in 1779 when Danish scientist Christian Kratzenstein, working at the Russian Academy of Sciences, constructed acoustic models of the human vocal tract capable of producing sustained vowel sounds. This was followed by Wolfgang von Kempelen’s bellows-operated "acoustic-mechanical speech machine" in 1791, which integrated primitive tongue and lip models to articulate consonants. In 1837, Charles Wheatstone replicated and improved upon von Kempelen’s architecture, a design later resurrected by Paget in 1923, while M. Faber developed the "Euphonia" in 1857 [4].

The transition to electronic synthesis occurred in the 1930s when Bell Laboratories introduced the VOCODER (Voice Coder), a keyboard-driven electronic speech analyzer and synthesizer. This architecture was subsequently refined by Homer Dudley into the VODER, which was showcased at the 1939 New York World's Fair [5]. By the late 1940s, Dr. Franklin S. Cooper and his colleagues at Haskins Laboratories developed the *Pattern Playback* machine. This system converted visual representations of speech phonemes, spectrograms, back into acoustic signals, enabling researchers like Alvin Liberman to isolate critical acoustic cues required for phonetic perception [6]. While early electronic speech synthesizers were marked by a distinct robotic quality and limited intelligibility, modern synthesis has advanced significantly, though a perceptible gap remains between synthesized output and natural human speech.

Recent research has highlighted the importance of integrating paralinguistic vocalizations into automatic speech recognition (ASR) and text-to-speech (TTS) systems. Despite the advancements, many languages, particularly low-resource languages, face significant challenges in developing effective speech recognition and synthesis systems. Innovative solutions are emerging to enhance the capabilities of speech recognition systems. The future of speech recognition and synthesis technologies is promising, with ongoing research focusing on enhancing user experience and accessibility [7]. The advancements in speech recognition and synthesis technologies are paving the way for more inclusive and expressive communication systems. As we continue to address the challenges faced by low-resource

languages and explore innovative solutions, the potential applications of these technologies will expand, ultimately enhancing communication across diverse linguistic and cultural contexts. To address this socio-technical gap, this paper proposes an accessible, software-driven framework designed to bridge the communication divide between these two communities.

2. Problem statement

Despite rapid advancements in technology, medical engineering, and modern communication networks, significant demographic disabled community remains marginalized in low income countries. Incidences of sensory impairments resulting from congenital conditions, progressive oral and systemic diseases, and traumatic accidents continue to rise globally [8]. Individuals afflicted with profound visual, auditory, or vocal impairments are frequently rendered unable to utilize communication technology, isolating them from the digital economy and their broader communities [9][10].

In Africa, the core challenge lies in the fact that contemporary speech recognition and synthesis systems are developed primarily for typically developing users. Commercially available assistive communication mediators remain scarce, rely heavily on proprietary hardware, and are cost-prohibitive for average users. While technology continues to evolve exponentially, there is a distinct lack of cohesive, software-only frameworks that bridge the specific bidirectional communication gap between deaf-mute and visually impaired individuals without requiring human intervention or expensive specialized equipment.

To mitigate these limitations, this study presents a zero-cost, software-driven speech recognition and synthesis tool implemented via a MATLAB Graphical User Interface (GUI). The overarching proposed solution was to establish an accessible, hardware-independent speech-text-speech framework that facilitates communication without a human mediator. The operational pipeline of the proposed framework is divided into the following two primary subsystems.

2.1 The speech synthesis

The speech synthesis was designed for users with hearing or vocal impairments. As shown in Fig 1, the user inputs text into the system interface which utilizes rule-based and dictionary-driven algorithms to convert the written string into synthesized acoustic signals. This allows a visually impaired interlocutor to receive the message audibly.

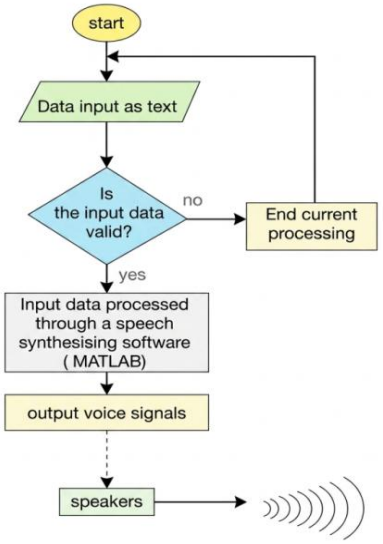


Fig.1 the speech synthesis flowchart

2.2 The speech recognition subsystem

The speech recognition subsystem shown in Fig 2 was designed for visually impaired users. The user responds verbally to the system. The framework performs acoustic signal processing and pattern recognition on the input waveform to transcribe the isolated words or conversational sentences into text, displaying it on-screen for the hearing-impaired user.

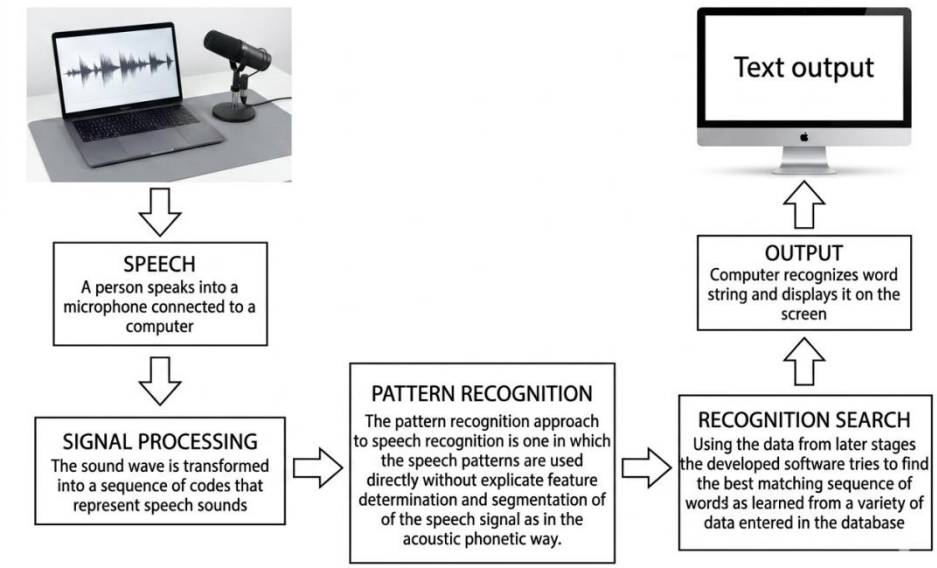


Fig.2 speech recognition subsystem

By eliminating the need for auxiliary hardware beyond a standard computer microphone and speaker, the Speech Recognition and Synthesis Tool (SRST) provide a highly stable, accessible, and scalable pedagogical tool. Beyond peer-to-peer assistance, it functions as an educational aid, allowing lectures captured via a standard headset to be logged as real-time text files, thereby bridging the socio-technical divide in classrooms, public service organizations, and commercial workspaces.

3. Literature review

Human-computer interaction (HCI) focuses on the modalities through which users interface with computational systems. Traditional manual inputs (e.g., keyboards) and visual outputs pose severe barriers for individuals with physical or sensory restrictions. Automatic speech recognition (ASR) frameworks mitigate these challenges by translating spoken acoustic signals into machine-readable digital text [11][12].

A comprehensive speech-driven interface comprises automatic speech recognition (ASR), which the computational emulation of human auditory perception to transcribe spoken words, and speech understanding (SU), which the higher-level semantic, syntactic, and pragmatic analysis of the transcribed text to infer intent. Achieving robust ASR is a knowledge-intensive process that must account for highly variable environmental and biological factors, spanning acoustic waveform variability, phonetic co-articulation, and ambient noise.

3.1 Technical approaches to speech recognition

3.1.1 The acoustic-phonetic approach

The acoustic-phonetic approach [13] postulates that speech sounds possess distinct, identifiable acoustic properties that can be modeled linearly. The standard workflow operates sequentially across four phases as shown in Fig 3.

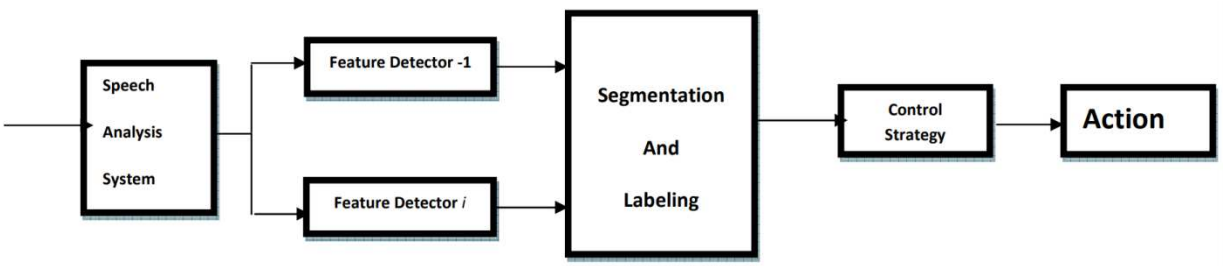


Fig.3 pipeline of an acoustic-phonetic system

First, the speech analysis stage generates a spectral representation of the time-varying signal using linear predictive coding (LPC) or filter bank methods. Next, feature detection maps these parameters to a matrix of broad phonetic properties. The segmentation and labeling phase then isolates stable spectral regions and matches them to individual phonetic units, which are finally decoded into word sequences.

3.1.2 The pattern recognition approach

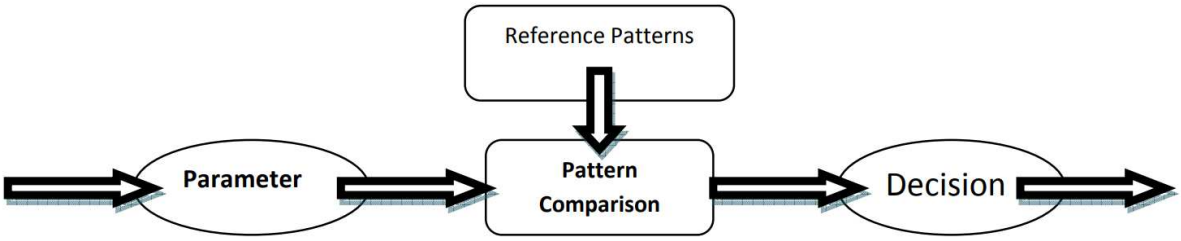


Fig.4 block diagram of the pattern recognition approach

The pattern recognition approach [14] eliminates explicit phonetic feature determination and segmentation. Instead, it operates directly on raw parameterized acoustic patterns shown in Fig 4. So, supervised learning strategy acts on a training set to establish reference templates or statistical models of target words/phrases. Moreover, an unknown test utterance is compared directly against the stored reference templates, and a decision rule evaluates the goodness-of-match to classify the input.

3.1.3 The artificial intelligence approach

The artificial intelligence methodology [15] functions as a hybrid paradigm, combining the formal rules of the acoustic-phonetic approach with the statistical classification strengths of pattern recognition. It integrates expert systems, neural networks, or symbolic logic to analyze acoustic features and automate the decoding decision process.

3.2 Selection of the pattern recognition framework

The pattern recognition approach was selected as the core engine for the proposed speech recognition and synthesis tool (SRST) due to the following three advantages. One, the underlying algorithms are structurally transparent and highly compatible with real-time digital signal processing. Two, the architecture maintains high classification accuracy across diverse vocabularies, speaker profiles, and acoustic feature sets. Three, because no explicit, language-specific syntactic rules are hardcoded, the framework is insensitive to vocabulary adjustments, grammar syntax, or semantic variations.

3.3 Challenges in the system design

Designing reliable ASR systems remains a complex task due to the high dimensionality and non-linear variability of human speech. Raw audio signals contain significant redundant data (e.g., background noise, speaker-specific anatomy markers) that does not contribute to classification. Furthermore, acoustically distinct sounds can generate similar waveforms, whereas minor prosodic inflections can completely distort a phoneme's time-domain wave. Modeling the complete acoustic signal mathematically generates a computational load too heavy for real-time processing on standard edge-computing hardware, requiring efficient feature extraction techniques to isolate invariant characteristics. So, speech synthesis is the artificial production of the human voice via a text-to-speech (TTS) architecture split into a linguistic front-end and an acoustic back-end synthesizer as shown in Fig 5.

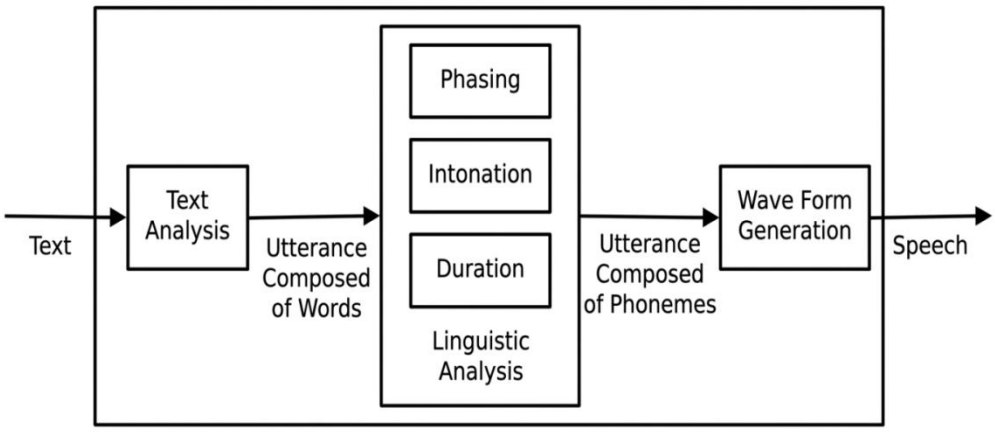


Fig.5 architecture of a text-to-speech processing pipeline

First, raw strings containing numbers, abbreviations, and symbols are expanded into fully written-out orthographic words. Next, the normalized text is converted into phonetic transcriptions and tagged with prosodic rules (phrase boundaries and intonation) to form a symbolic linguistic representation. Finally, the back-end synthesizer maps the symbolic representation into a continuous acoustic waveform evaluated by its naturalness (human-like cadence) and intelligibility (ease of understanding).

3.4 speech synthesis system paradigms

Research efforts have focused on improving the expressiveness and emotional accuracy of speech synthesis. Wang, Tianrui, et al. [16] introduce Emo-VITS, a system based on the VITS model, which enables emotion-controlled text-to-speech (TTS) synthesis. Their approach involves an emotion network that extracts and fuses global and local emotional features via an attention mechanism, resulting expressive synthetic speech.

Pérez-Córdoba et al. [17] present advances in Iberian languages, covering topics such as speech perception, analysis, synthesis, and speaker verification. Their work reflects ongoing efforts to adapt speech technologies to specific linguistic contexts, addressing both technical challenges and application-specific needs.

EUH Divyanshu et al. [18] reports that ensemble learning models have achieved superior accuracy in automatic age and gender recognition, which are crucial for personalized applications and user profiling. Similarly, A Chaurasia et al. [19] review accessibility techniques, noting that speech-to-text and voice recognition systems have attained low word error rates.

Karibayeva et al. [20] focus on Turkic languages, developing and evaluating a multilingual speech corpus to support recognition and synthesis tasks. Their work underscores the importance of creating extensive datasets to improve speech technology performance in underrepresented languages, which is essential for inclusive and global speech applications.

Therefore, the literature indicates that speech recognition and synthesis technologies have advanced considerably, driven by innovations, with ongoing efforts to enhance accuracy, emotional expressiveness, multilingual support, and accessibility in diverse application contexts including Ethiopia.

4. Methodology

At the architectural level, our automatic speech recognition (ASR) engines comprise two distinct operational phases, training and testing inference. The training pipeline shown in Fig 6 encompasses speech acquisition, signal pre-processing, feature extraction, and statistical acoustic modeling. The testing pipeline shown in Fig 7 mirror-matches these feature extraction steps to perform pattern classification against the trained models.

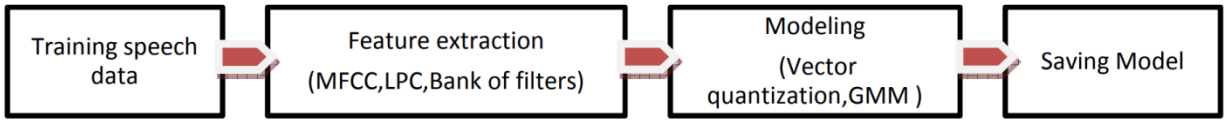


Fig.6 block diagram of the training phase

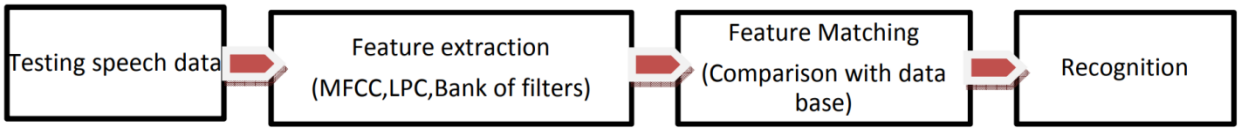


Fig.7 block diagram of the testing phase

4.1 Speech acquisition and data preparation

The reliability of an acoustic training vector space was governed by four primary environmental and physiological variables. First, the system utilizes an isolated word automatic speech recognition (IWASR) database compiled from 20 distinct speakers (15 male, 5 female) from Aksum city. Second, field recordings were conducted within the electrical communication and computer laboratories of Aksum University, Ethiopia. The environment was intentionally subjected to ambient operational noise to ensure the trained models remain robust against real-world acoustic disturbances. Third, acoustic waveforms were captured using a standard, noise-sensitive omnidirectional microphone, matching consumer-grade computing setups. Finally, the vocabulary was strictly constrained to specific isolated words and short conversational sentences.

4.2 Speech signal pre-processing and feature extraction

4.2.1 Analysis of quasi-stationary signals

As shown in Fig 8, speech profiles are non-stationary signals when analyzed over long durations (>200). However, when observed over short windows (between 5 ms and 100 ms), human speech exhibits quasi-stationary characteristics where the spectral properties remain relatively constant. Consequently, short-time spectral analysis forms the baseline for feature parameterization.

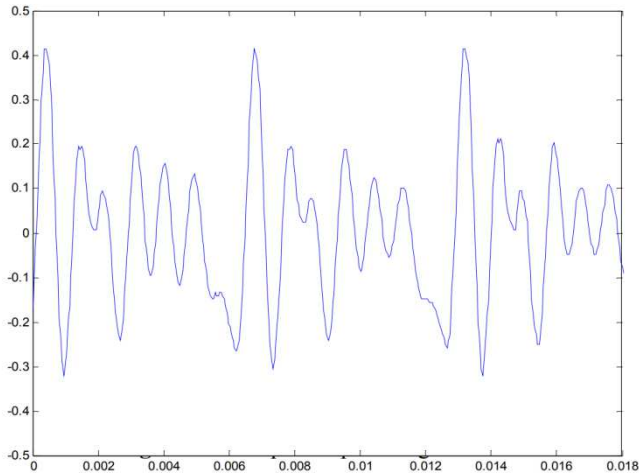


Fig.8 speech signal taken from laboratory

While linear predictive coding (LPC) provides an efficient approach to estimate vocal tract formants by modeling the throat and mouth as a single acoustic tube, it exhibits significant degradation when modeling side branches (e.g., nasal sound zeros) or mixed acoustic sources (e.g., fricative hisses). Because inaccuracies in formant estimation bleed critical acoustic data into the signal residue, our methodology relies exclusively on mel-frequency cepstral coefficients (MFCC) to mimic the non-linear auditory perception of the human ear.

4.2.2 The MFCC processing pipeline

The continuous acoustic signal was digitized at a sampling rate exceeding 10 kHz to prevent aliasing and capture frequencies up to 5 kHz, encapsulating the primary energy band of human speech. The MFCC front-end executes across five sequential stages as shown in Fig 9.

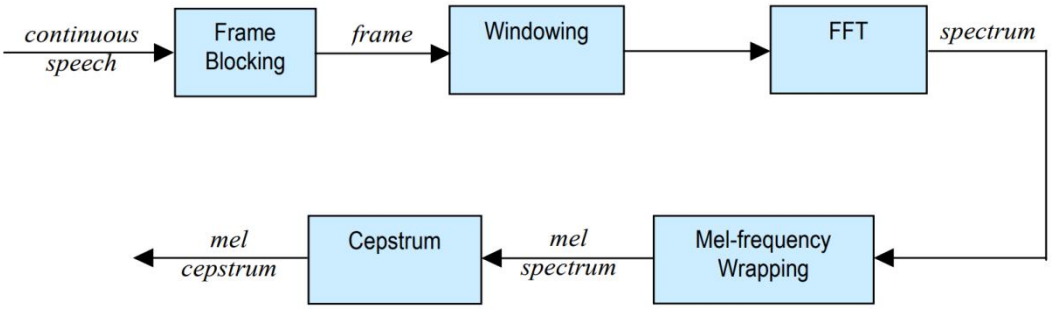


Fig.9 block diagram of the MFCC processor

In Fig 9, the continuous signal is segmented into overlapping discrete frames of N samples, with adjacent frames separated by a shift of M samples (M < N). To facilitate a computationally efficient radix-2 fast fourier transform (FFT), the frame size is set to N = 256 (corresponding to an approximate 30 ms window), with an overlap parameter of M = 100. To minimize spectral leakage and signal discontinuities at the boundaries of each frame, a hamming window w(n) tapers the signal edges to zero. The windowed signal yl(n) for a frame xl(n) is defined mathematically as:

$$y_l(n) = x_l(n)w(n), \quad 0 \le n \le N-1$$

Where the coefficients of the Hamming window follow the structure:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

To transition each frame from the time domain to the frequency domain, a discrete fourier transform (DFT) is executed via fast fourier transform (FFT) algorithm:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-j2\pi kn/N}, \qquad k = 0,1,2,\ldots,N-1$$

The absolute magnitude spectrum Xk is retained for subsequent spectral processing, where positive frequencies map between 0 and – Fs/2.

Human psychological pitch perception behaves non-linearly, exhibiting linear spacing below 1000 Hz and logarithmic spacing above 1000 Hz. To replicate this behavior, the magnitude spectrum was passed through an overlapping triangular bandpass filter bank shown in Fig 10. The spacing and bandwidth of the filters are constant along the subjective Mel-scale. For this implementation, a bank of K = 20 overlapping triangular filters was applied directly in the frequency domain.

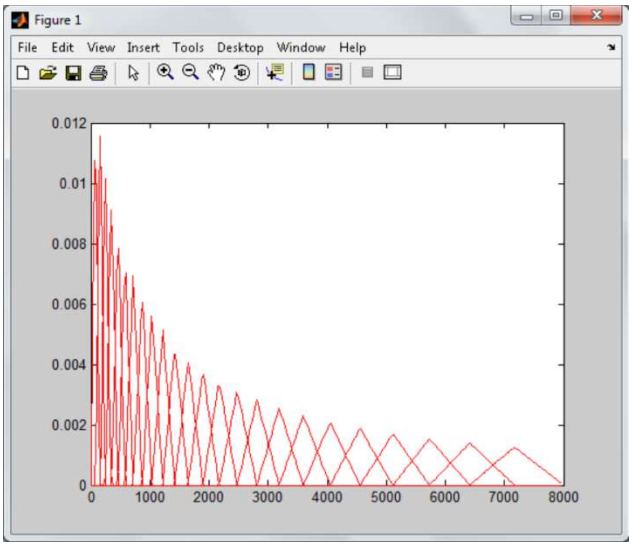


Fig.10 sample of mel-spaced filter bank

To compress the information rate and isolate local spectral properties, the logarithm of the mel-wrapped power spectrum is converted back to the quasi-time domain. Because the log-mel spectral coefficients are purely real numbers, a discrete cosine transform (DCT) is applied. If the log-energy outputs of the filter bank are denoted as Sk, the final MFCCs (cn) are formulated as:

$$\tilde{c}_n = \sum_{k=1}^{K} (\mathrm{Log}\tilde{S}_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \qquad n = 0,1,\ldots,K\text{-}1$$

4.3 Statistical acoustic modeling

4.3.1 Vector quantization (VQ) via the LBG algorithm

To minimize data storage requirements and accelerate downstream similarity matching, Vector Quantization [21] maps continuous high-dimensional MFCC vectors into a discrete codebook of representative spectral shapes. This fixed-to-fixed length block-coding compression method transforms continuous acoustic coordinates into localized encoding regions partitioned by code vectors (see Fig.11).



Fig.11 one-dimensional VQ

The system implements the linde-buzo-gray (LBG) iterative splitting algorithm to construct the codebook, bypassing multi-dimensional integration. So, the global centroid of the entire training sequence is calculated to initialize a codebook of size 1. This initial vector is split into two via a small perturbation vector. The K-means iterative optimization loop runs until convergence to stabilize the two new code vectors. The resulting centroids are split sequentially (2 to 4 to 8 …) until the target codebook size is achieved. By representing speech frames via codebook indices, similarity computations during decoding are optimized from expensive euclidean distance calculations down to simple table lookups (see Fig.12).

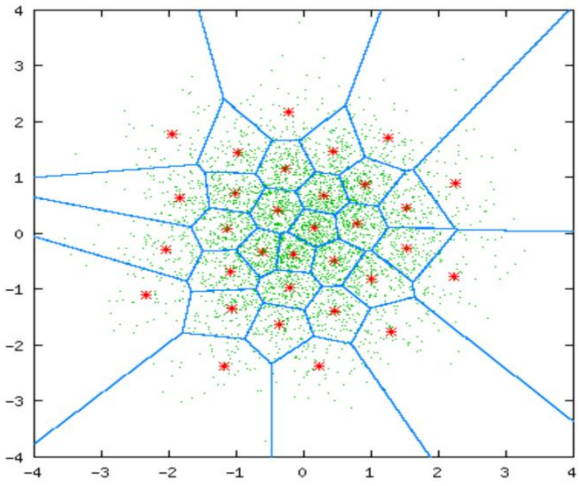


Fig.12 two-dimensional VQ

4.3.2 Gaussian mixture models (GMM)

While hidden Markov models (HMMs) temporal state transitions over time-varying frames, Gaussian mixture models (GMMs) treat the acoustic features as a combination of statistically independent underlying components. These components parametrically model the multi-modal probability density function (PDF) of distinct acoustic classes belonging to specific words.

Representing a convex combination of multi-dimensional Gaussian densities, a GMM serves as a universal density approximator [22]. For an input feature vector X and a predefined number of mixture components M, the conditional probability density are defined as:

$$P(x/\,i) = 1/(2\pi)D/2 * \left|\sum\right|^{1/2}) * \exp^{-1/2*(X-\mu)T*\sum^{-1}*(X-\mu)}$$

4.4 Operational pipeline and silence removal

To prevent ambient background noise from contaminating the acoustic models and to optimize processing efficiency, non-speech intervals are discarded at both boundaries of the utterance. The system incorporates two distinct silence removal strategies. First, during the offline training and modeling phase, Audacity software is used to manually clean and truncate the training .wav corpora. Second, during real-time inference, manual editing is impossible. The framework deploys an automated VAD algorithm [23] that calculates short-time energy thresholds and zero-crossing rates (ZCR) to differentiate between voiced speech, unvoiced speech, and background silence.

During the online testing phase, an unknown input utterance sequence Y = Y1, Y2, …, YT is processed by the VAD engine, parameterized into MFCC features, and passed to a maximum likelihood classifier. The classifier evaluates the likelihood function across the stored word-specific GMM structures, returning the text string that exhibits the highest probability match for display to the hearing-impaired user.

5. Implementation of the software

The proposed software-driven framework comprises two primary technical subsystems implemented within a cohesive MATLAB graphical user interface (GUI); an automatic speech recognition (ASR) subsystem that converts acoustic input into digital text, and a text-to-speech (TTS) synthesis subsystem that maps input text strings back into synthesized vocal waveforms. This dual-architecture design establishes a direct, bidirectional, and unassisted communication channel between visually impaired and hearing- or speech-impaired individuals.

5.1 Text-to-speech (TTS) synthesis framework

The implementation of the speech synthesis subsystem transforms raw text strings into an intelligible acoustic signal. The internal architecture of the implemented TTS engine is structured across four fundamental computational layers as shown in Fig 13.

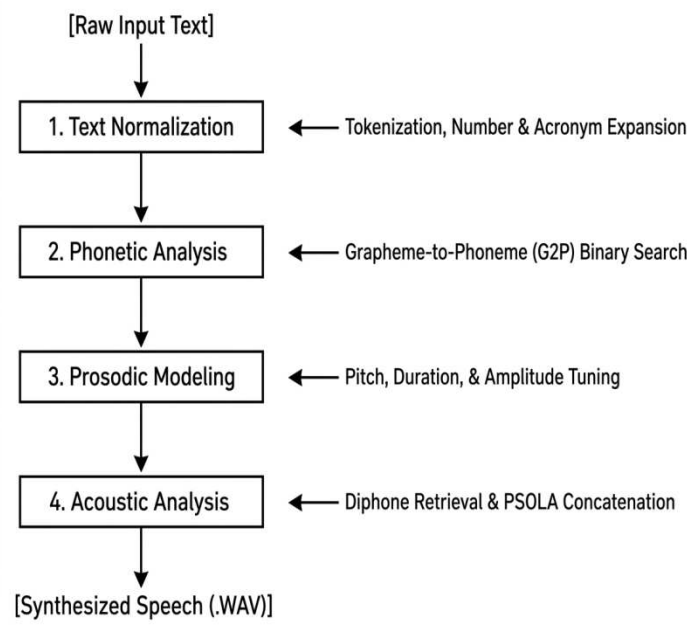


Fig.13 layered architectural pipeline of the implemented TTS synthesis engine

5.1.1 Text normalization layer

The linguistic front-end processes raw character sequences through a three-stage pipelined parser to generate clean, standard orthographic word arrays. In stage one, numbers and numerical strings are contextually expanded into their textual equivalents. The module isolates context to alternate between mathematical cardinal values, sequential strings, and fractional expressions. For example:

Temporal/Identifier Context: "1776" to "seventeen seventy six"
Serial/Telephonic Context: "1776" to "one seven seven six"
Quantifier/Cardinal Context: "1776" to "one thousand seven hundred and seventy six"
Floating-Point Extraction: "0.303" to "point three knot three"

In stage two, character strings representing corporate or technical acronyms are converted into individual letter strings or unified pronunciation blocks based on structural classification (e.g., "NATO" to "nayto"; "HIV" to "aitcheyevee"). Standard textual abbreviations are systematically replaced via lookup filters with their expanded semantic forms (e.g., "Mrs." to "Misses"; "St. Joseph St." to "Saint Joseph Street").

In stage three, using a dedicated delimiter array, sentences are isolated into structural units consisting of single tokens, numbers, or normalized acronyms. The tokenizer handles multi-character phonetic combinations (e.g., digraphs like sh, ch, and th) to preserve structural integrity prior to phonetic mapping, implementing deterministic boundaries (e.g., "China" to "Ch-ai-na").

5.1.2 Phonetic analysis

The normalized text tokens (graphemes) must be mapped to their corresponding phonological codes. The system utilizes a standardized 44-phoneme English target matrix, comprising 19 vowel structures and 25 distinct consonant footprints.

To determine word pronunciation efficiently, systems generally evaluate dictionary-based lookups, rule-based phonetic algorithms, or a hybrid of both. This design implements a high-performance, Dictionary-Based Approach [24], utilizing a dictionary derived from the Carnegie Mellon university (CMU) pronouncing database, containing over 175,000 words mapped to their target diphone structures.

To achieve real-time execution speeds suitable for continuous dialogue, word lookup is optimized via a native binary search algorithm. First, the complete dictionary matrix is initialized with defined start, middle, and end indexes. Next, during each iteration target input word is evaluated alphabetically against the word at the middle index. Then, if the target word is alphabetically smaller, the upper search boundary is constricted otherwise, the lower boundary is shifted. Finally, this bounds reduction splits the data space exponentially, ensuring lookup latencies under 1.0 second on standard consumer compute cores without requiring heavy memory caching overhead.

5.1.3 Multi-level data structure

Once lookups resolve successfully, phonological strings are packaged into a multi-level data structure (MLDS). The MLDS [25] acts as an intermediate matrix containing structural data required by the downstream synthesizer, including isolated word strings, mapped diphone representations, and target prosodic parameters (fundamental pitch trajectories, duration constraints, and localized amplitude scaling parameters). So, structuring data at this layer, the system decouples linguistic front-end processing from digital signal manipulation, facilitating modular code optimization.

5.1.4 Acoustic synthesis and waveform concatenation

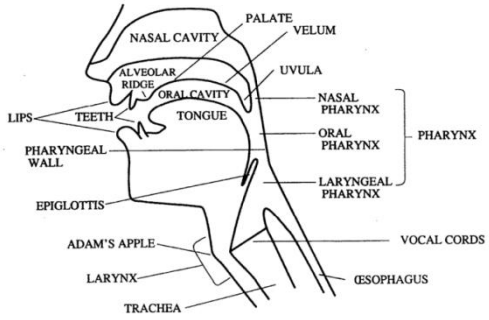


Fig.14 the model of the human vocal tract

The implementation techniques for synthesizing speech were based on models of the Human Vocal Tract [32] as shown in Fig 14. The final back-end layer constructs continuous speech by retrieving pre-recorded natural human diphone fragments from an offline audio catalog. Diphones represent the acoustic transitions captured directly at the boundary from the midpoint of one phoneme to the midpoint of the next (e.g., consonant-consonant, consonant-vowel, vowel-consonant, vowel-vowel transitions), preserving organic co-articulation signatures.

The system performs digital signal manipulation, within MATLAB [26], using automated file parsing engines to load, read, and manipulate uncompressed audio waveforms (.WAV). To ensure that the concatenated diphone strings sound natural and intelligible rather than robotic, the system dynamically alters pitch, duration, and amplitude metrics.

So, during pitch adjustment, the fundamental frequency (F0) is modified exclusively across vowel windows. To elevate the vocal pitch, the interval between explicit pitch marks is compressed, shrinking the pitch period. Conversely, expanding the spacing between pitch marks lowers the perceived fundamental pitch.

At the duration alteration the system deploys a pitch-synchronous overlap and add (PSOLA) algorithm. Increasing the number of overlapping signal additions extends the perceived duration of the phoneme without introducing pitch distortion, whereas decreasing the overlap rate shortens the segment.

During amplitude scaling, localized volume and emphasis variations are induced by scalar multiplication of the discrete audio vector x(n) by an adjustment constant alpha, $y(n) = \alpha.x(n)$. An emphasis coefficient where $\alpha > 1$ amplifies the output voltage envelope, while a fractional coefficient where $\alpha < 1$ attenuates the acoustic signal.

During the final concatenation phase, the PSOLA framework [27][28] acts directly upon the overlapping boundaries of consecutive diphones to smooth phase transitions and suppress audible clicks or digital glitch artifacts. When joining distinct words to form sentences, the engine switches to a straight boundary concatenation routine, inserting programmatic silent pauses to replicate natural conversational pacing.

5.2 Automatic speech recognition implementation

The inverse execution pipeline operates within the ASR module to process spoken answers from visually impaired users. When an audible waveform is captured by the audio hardware interface, an online voice activity detection (VAD) filter monitors short-time energy envelopes and zero-crossing distributions to strip silent frames from the signal stream.

The remaining acoustic signal data is normalized, windowed using a discrete Hamming profile, and parameterized via a fast fourier transform (FFT) into 20-dimensional mel-frequency cepstral coefficient (MFCC) feature vectors. These vectors are passed through an optimized classification engine trained via the linde-buzo-gray vector quantization (LBG-VQ) clustering algorithm and a multi-modal Gaussian mixture model (GMM).

By applying a maximum likelihood estimation decision rule against the word models stored within the directory architecture, the system accurately matches the acoustic input to its corresponding textual string, printing it directly to the GUI screen for reading by the hearing-impaired interlocutor.

6. Results and discussion

The experimental evaluation of the bidirectional communication framework was conducted by assessing both the ASR engine and the TTS synthesis engine. The complete system was prototyped and integrated within a dual-window MATLAB graphical user interface.

6.1 Speech recognition evaluation

The performance of the ASR subsystem was validated under two testing conditions; offline validation using an independent test corpus (excluding files from the modeling phase) and online live inference. The signal processing front-end parameterizes speech patterns using a 20-dimensional mel-frequency cepstral coefficient (MFCC) feature extraction routine.

6.1.1 Acoustic modeling and GMM-VQ convergence

To ensure environmental robustness, training data were collected across two distinct acoustic settings and three distinct diurnal intervals (morning, noon, and evening). Individual frame-level MFCC vectors were concatenated into a global matrix and quantized via the linde-buzo-gray vector quantization (LBG-VQ) clustering algorithm.

To refine classification boundaries, the vector quantization centroids (means, covariances, and mixture weights) were optimized using a multi-modal Gaussian mixture model (GMM). Comparative testing revealed that incorporating the GMM density estimation improved the classification accuracy by approximately 1% compared to utilizing a standalone LBG-VQ minimum-distance approach. This marginal difference suggests that the localized codebook centroids captured the invariant acoustic envelopes effectively, while the GMM provided smoother probabilistic boundaries for marginal phoneme distributions.

6.1.2 Inference optimization

During online inference, a voice activity detection (VAD) module executes short-time energy and zero-crossing rate (ZCR) assessments to trim silent frames dynamically. The primary computational bottleneck observed during the testing phase is the execution latency of the maximum likelihood estimation classifier. The engine performs an exhaustive sequential comparison between the test vector sequence Y and every word model configuration stored in the database:

$$\hat{w} = \arg\max_{k} P(Y \mid \lambda_k)$$

Because the search space scales linearly with vocabulary size, decoding latencies increase proportionately. Although establishing likelihood thresholds for each model configuration can theoretically truncate the search path, selecting fixed thresholds proved impractical due to the high inter-speaker variance of acoustic features under changing ambient conditions.

Table 1, speech recognition classification accuracy

No	Operational Mode	Verification Accuracy
1	Offline Corpus (Isolated Words)	91.2%
2	Online Inference (Quiet Room)	89.5%
3	Online Inference (Ambient Noise)	84.1%

As summarized in Table 1, the system achieves an overall average verification accuracy of approximately 90%. The primary source of error stemmed from phonetic misclassifications among words exhibiting overlapping or highly similar phone characters. To mitigate these local errors, expanding the variability and sample size of the training corpora for these specific acoustic profiles is necessary.

6.1.3 Architectural mitigations

To enable hands-free interaction for visually impaired users, the system incorporates an open-ended audio polling loop. The input signal power is monitored continuously; if the short-time energy drops below a calibrated noise floor threshold, the signal is discarded as silence, and the system loops back to the recording step. However, live deployment revealed two core vulnerabilities in this automated approach.

First, non-speech impulsive noises (e.g., friction, material impacts, ambient collisions) surpass the short-time energy threshold. The algorithm attempts to decode these artifacts, inducing processing overhead and rendering erroneous textual outputs.

Second, because the MATLAB script processes loops sequentially, the execution of the feature-matching pipeline introduces an input blackout window. If a user initiates an utterance while the system is decoding a prior artifact, the speech waveform is clipped or omitted, resulting in failed or incomplete recognition.

To resolve these technical limitations, the system architecture was adapted to support an asymmetric control loop. The window controls are managed manually by the hearing-impaired interlocutor via GUI triggers. Furthermore, the recognition and synthesis loops are structurally isolated into independent threads, ensuring that the acoustic output generated by the TTS engine does not feed back into and activate the ASR microphone loop.

6.1.4 Cross-lingual adaptability

The pattern-recognition architecture operates independently of language-specific grammar models, making it natively flexible and adaptable across diverse vocabularies. To demonstrate this capability, a trilingual conversation script was successfully verified within the system, integrating proper nouns and phrases from Tigrigna (*"Hagos"*), Amharic (*"Addis"*), and English (*"Thank you"*).

6.2 Speech synthesis implementation and refinement

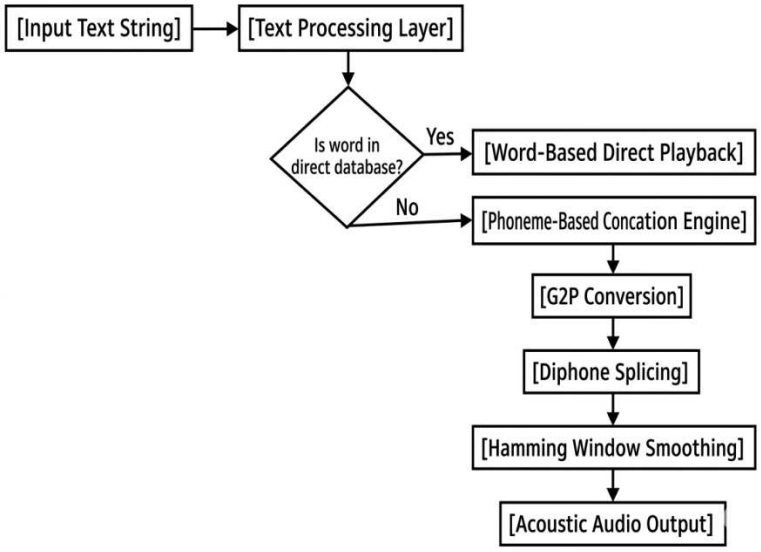


Fig.15 algorithmic routing matrix for the text-to-speech synthesis framework

As shown in Fig 15, the TTS subsystem implements both word-based direct lookup synthesis and phoneme-based concatenation models.

6.2.1 Processing and database compilation

Fig 16 shows a sound recorded after and before edition performed. Acoustic units were compiled using two distinct methodologies. At first, Manual Audio Corpora Engineering [29][30] in which targeted vowels and consonants were recorded using an omnidirectional microphone and clean-edited to trim trailing

channel noise and border silences. To minimize environmental noise contamination from consumer hardware, recordings were optimized by utilizing isolated spaces during low-activity nocturnal intervals.

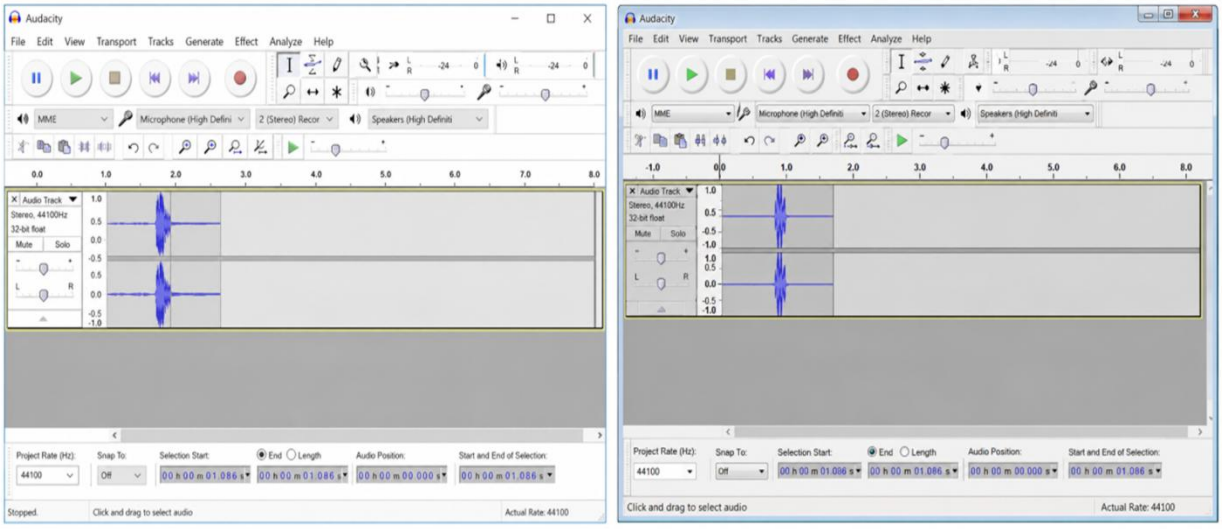


Fig.16 sound recorded after and before edition performed

Second, a synthetic corpus generation was compiled to evaluate highly dynamic prosodic variations, the open-source festival framework (festvox.org) was utilized to automatically render consistent phonetic corpuses across diverse pitch parameters and speaker profiles.

The system maps character inputs to their corresponding audio files asynchronously (async mode for parallel processing tasks) or sequentially (sync mode for word strings) using native MATLAB audio tools. For arithmetic expressions, a dedicated indexing loop matches inputs against a constrained cardinal vocabulary (1 to 100), enabling immediate arithmetic synthesis.

6.2.2 Algorithmic optimization of phase boundaries

When executing the phoneme-based synthesis model, MATLAB introduces distinct boundary latency. The system evaluates a sequence of conditional statements to resolve the phonetic identities of incoming tokens. As the textual array scales, the execution overhead of these conditional checks introduces a micro-delay between consecutive phone arrays.

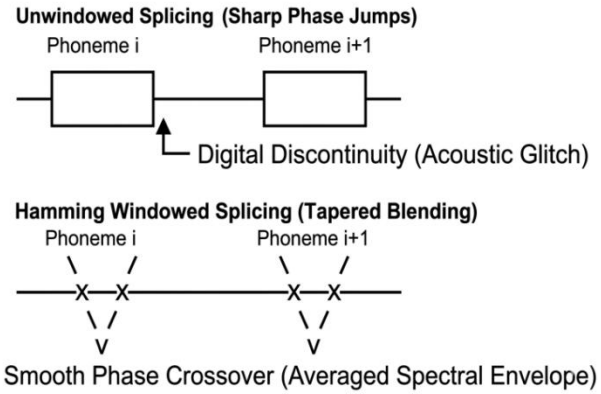


Fig.17 time-domain boundary smoothing

As illustrated in Figure 17, direct time-domain splicing of audio fragments yields sharp digital phase jumps at the boundaries. This discontinuity manifests as harsh, robotic, and unpleasant acoustic clicks. To smooth these boundaries, the system evaluates the length of each phoneme vector and applies an N-point symmetric Hamming [31] window:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$

By tapering the boundary amplitudes to zero, the transition point morphs into an averaged crossover envelope. This attenuation suppresses phase-step distortions, producing a smooth and continuous speech output.

6.3 Graphical user interface (GUI)

The interactive system was split into two distinct operational interfaces designed to coordinate communication between the two users.

6.3.1 Speech recognition interface

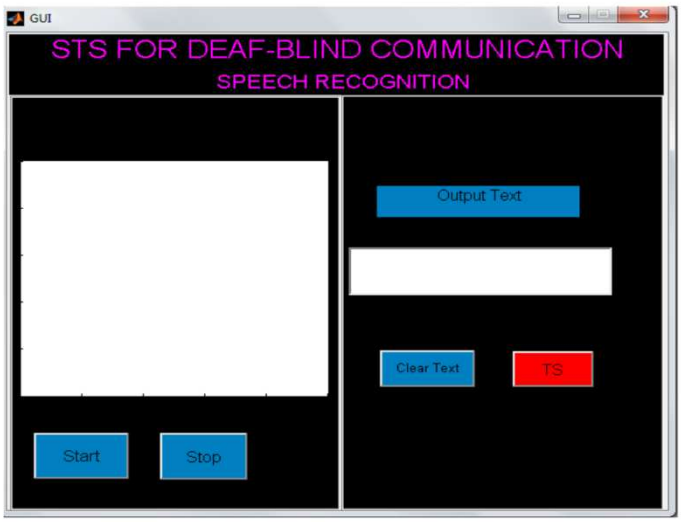


Fig.18 GUI of speech recognition

As shown in Fig 18, the ASR window features a dedicated control interface tailored for the hearing-impaired operator by activating the 'Start' button initializes the audio hardware interface and plots the incoming acoustic waveform in real-time on a time-domain axis. Once the Maximum Likelihood engine resolves decoding, the recognized text string is printed inside a high-contrast output text box. So, an automatic playback system echoes the captured audio loop for local validation. Moreover, pressing the 'TS' button suspends the ASR thread and launches the Speech Synthesis window.

6.3.2 Speech synthesis interface

As shown in Fig 19, the TTS window provides a structured environment for generating verbal output. So first, the user selects the synthesis routing profile (word-based, phoneme-based, or numeric mode). Then, the user types the text string into an input text box. Clicking the 'Synthesize' button parses the string through the text normalization and Hamming-smoothed synthesis engine, generating immediate verbal output via the audio hardware. The 'ST' button suspends the synthesis framework and re-initializes the ASR window, allowing the visually impaired interlocutor to respond.

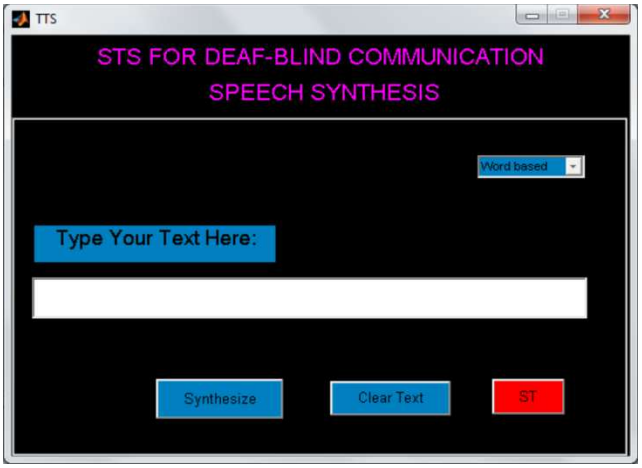


Fig.19 GUI of speech synthesis

7. Conclusion and recommendations

7.1 Conclusion

The primary objective of this study was the development and implementation of an integrated, software-driven speech recognition and synthesis framework to facilitate bidirectional communication between hearing and visually impaired individuals. By engineering a dual-window graphical user interface (GUI) within the MATLAB environment, a functional communication bridge was established. The system robustly transcribes real-time acoustic input into high-contrast digital text for hearing-impaired users while concurrently converting textual string inputs into synthesized, intelligible verbal waveforms via an optimized diphone concatenation engine for visually impaired individuals.

Experimental validation indicates that the prototype meets its design objectives, achieving an average operational efficiency of approximately 90%. The architectural design of the system offers straightforward scalability, allowing the core vocabulary and phonetic database to be expanded using the established parameterization and modeling protocols. Furthermore, current limitations regarding ambient noise susceptibility and computational latency can be mitigated by deploying higher-grade hardware, such as directional, noise-canceling microphone arrays and edge-computing processing units. Because the underlying ASR and TTS modules are decoupled and structurally modular, each sub-framework possesses independent utility and can be adapted for discrete assistive technology applications beyond the scope of this research project.

7.2 Recommendations

The prototype developed in this research demonstrates a baseline framework for language-independent speech processing, and will serve as a foundational architecture for more advanced speech-driven interfaces. First, we recommend integrating a machine translation layer between the ASR and TTS pipelines which would allow the system to function as an automated language translator, expanding its utility to cross-lingual assistive communication. Second, leveraging the underlying GMM and VQ speaker-modeling capabilities, the framework can be extended into speaker verification and voice-biometric security systems. Third, transitioning the front-end string parsing from a sequential conditional layout to deep-learning-based G2P models would reduce boundary latency and improve real-time execution. Therefore, future field deployments should transition from standard omnidirectional transducers to multi-microphone arrays utilizing hardware-level adaptive beamforming to improve speech intelligibility in highly acoustic or noisy environments.

8. Accordance

All procedures were conducted in accordance with the ethics of Aksum University 2018 standard.

9. Competing interests

We declare that all the authors have no competing interests or other interests that might be perceived to influence the results and discussion reported in this paper.

10. Declarations

10.1 Ethical approval

Ethical approval certificate for this study was obtained from the ethics committee of Aksum University with a reference number *AKU-AIT: 01/224/2026*. The involved ethics committee members include Dr.Birhane Gebreyohannes, Mr.Hagos Hadgu and Ms.Eyerusalem Dagnew. All procedures involving participants were conducted in accordance with the ethical standards of the institution.

10.2 Consent to participate

Informed consent was obtained from all individual participants involved in the study, and the authors confirm that participants were aware of the study purpose, risks and benefits.

10.3 Consent to publish

All authors have read and approved the final manuscript and consented to its publication.

11. Data availability

All data generated and analyzed during the research are included within this paper and Science Data Bank with full name of the database “Bridging communication between hearing and visually impaired individuals via speech recognition and synthesis”.

Accession number and Link: DOI https://doi.org/10.57760/sciencedb.38434 or link https://cstr.cn/31253.11.sciencedb.38434

12. Funding

Not applicable

13. Author contribution

Dawit Teklu Weldeslasie conceived the study design. Yonas Desta Sheshegu developed the analytical tools and methodology. All authors conducted the experiments and gathered the data. Mehamed Ahmed Abdurrahman and Yonas Desta Sheshegu managed and organized the research database. Dawit Teklu Weldeslasie developed the GUI and drafted the initial manuscript. All authors critically reviewed, edited, and provided final approval for the published version.

References

[1]. Gayathiri, G., and Mr TJ Raju. *All the basic ideas you need to know about communication*. Wordsswiggle publication, 2025.

[2]. Kravets, Larysa, et al. "Dynamics of language systems in the era of digitalization." (2025).

[3]. Griffiths, L. (2025). Mythical AI: The Long Relationship with AI Through Myth and Alchemy. In: Batt, J.D., Erickson, J. (eds) Depth Psychology, Myth and Artificial Intelligence. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-031-94105-4_3

[4]. Nasirov, S. U. "PRINCIPLES OF DIGITAL SPEECH SIGNAL PROCESSING AND ANALYSIS OF THE ACOUSTIC CHARACTERISTICS OF HUMAN AND SYNTHETIC SPEECH." *International Conference Platform*. No. 6. 2026.

[5]. Jude, G. (2025). Out of the Analog Age: Singing Robots and the Uncanny Valley. In: How Vocaloid Works. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-031-92727-0_2

[6]. Whalen, D. H., and Philip E. Rubin. "Speech synthesis at Haskins Laboratories: From the pattern playback to articulatory synthesis." *The Journal of the Acoustical Society of America* 158.4_Supplement (2025): A23-A23. https://doi.org/10.1121/10.0040002

[7]. Huan Liao; Qinke Ni; Yuancheng Wang; Yiheng Lu; Haoyue Zhan; Pengyuan Xie; Qiang Zhang; Zhizheng Wu; "NVSpeech: An Integrated and Scalable Pipeline for Human-Like Speech Modeling with Paralinguistic Vocalizations", ARXIV-CS.SD, 2025.

[8]. Koratagere Anantha Kumar, S. (2026). Broadband Access and Digital Divide. In: Wireless Broadband Access to Bridge the Digital Divide. Textbooks in Telecommunication Engineering. Springer, Cham. https://doi.org/10.1007/978-3-032-09261-8_1

[9]. Ghazy, R.M., Mohammed AboElela, A., Abou-elyazid, H., S. Al-Qahtani, A. (2025). Sensory Impairment. In: Bennett, G., Goodall, E. (eds) The Palgrave Encyclopedia of Disability. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-031-40858-8_551-1

[10]. Davies, T.C., Desai, S. (2025). Empowering Voices: The Role of Emerging Technologies in Workplace Inclusion for Individuals with Speech Impairments. In: El Morr, C., El-Lahib, Y., da Silveira Gorman, R. (eds) Beyond Tech Fixes. Springer, Cham. https://doi.org/10.1007/978-3-031-93022-5_8

[11]. Schreiter, J., Heinrich, F., Hatscher, B. *et al.* Multimodal human–computer interaction in interventional radiology and surgery: a systematic literature review. *Int J CARS* 20, 807–816 (2025). https://doi.org/10.1007/s11548-024-03263-3

[12]. B G, N., G, T.Y. Automatic speech recognition: challenges, enhancements, and evaluation metrics. *Multimed Tools Appl* 84, 46627–46645 (2025). https://doi.org/10.1007/s11042-025-20998-6

[13]. Y. Deng, "Acoustic Phonetic Temporal Speech Representation," 2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Honolulu, HI, USA, 2025, pp. 1-7, https://doi.org/10.1109/ASRU65441.2025.11434703

[14]. Shabbir, Ahsan, et al. "Enhancing smart home environments: a novel pattern recognition approach to ambient acoustic event detection and localization." *Frontiers in Big Data* 7 (2025): 1419562. https://doi.org/10.3389/fdata.2024.1419562

[15]. Chu, J. Application of intelligent language teaching system based on speech recognition technology. *Int J Speech Technol* 29, 12 (2026). https://doi.org/10.1007/s10772-025-10230-z

[16]. Wang, Tianrui, et al. "Word-level emotional expression control in zero-shot text-to-speech synthesis." *Advances in Neural Information Processing Systems* 38 (2026): 147377-147405.

[17]. A. Piñeiro-Martín, C. García-Mateo, L. Docío-Fernández and M. d. C. López-Pérez, "Robust End-to-End Spoken Language Understanding in Low-Resource Settings," in IEEE Access, vol. 14, pp. 18812-18827, 2026. https://doi.org/10.1109/ACCESS.2026.3660257

[18]. Divyanshu, Ekram Ul Haque, Tanveer Alam, and Atul Kumar. "Impact of Age and Gender on the Accuracy of Computational Model-Based." *Computer Vision, Pattern Recognition, Image Processing, and Graphics: 10th National Conference, NCVPRIPG 2025, Jammu, India, July 16–18, 2025, Revised Selected Papers, Part I*. Springer Nature, 2026.

[19]. Chaurasia, A., Rajagopal, S. (2026). Real-Time Virtual Meeting Architecture for Deaf People: Audio Transcription, Noise Processing and Sign Language Translation Using LSTM Neural Networks. In: Pant, M., Deep, K., Nagar, A.K. (eds) Machine Learning and Its Applications to Healthcare. Mathematics for Sustainable Developments. Springer, Singapore. https://doi.org/10.1007/978-981-95-4831-6_8

[20]. Karibayeva A, Karyukin V, Tukeyev U, Abduali B, Amirova D, Rakhimova D, Aliyev R, Shormakova A. The Development and Experimental Evaluation of a Multilingual Speech Corpus for Low-Resource Turkic Languages. *Applied Sciences*. 2025; 15(24):12880. https://doi.org/10.3390/app152412880

[21]. , Lamkadam, Abdelmajid, and Mohammed Karim. "Time encoded signal processing and recognition with vector quantization: applied to Arabic numerals." *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 24.2 (2026): 481-489. http://doi.org/10.12928/telkomnika.v24i2.27443

[22]. Kret, Meghan. *Probabilistic Temporal Segmentation for Audio Moment Retrieval Using Gaussian Mixture Models*. MS thesis. The Cooper Union for the Advancement of Science and Art, 2026.

[23]. S. M. Ekolama, & C. I. Iyoloma. (2026). Voice Activity Detection in Cellular Networks: An Ekolama Loss–Optimised Deep Q-Network Approach for Adaptive Bandwidth Allocation and Speech Quality Enhancement. Journal of Optoelectronics and Communication, 8(1), 27–49. https://doi.org/10.5281/zenodo.18999249

[24]. P. Janyoi, A. Thangthai and P. Seresangtakul, "Transformer-Based Grapheme-to-Phoneme Conversion With Knowledge Transfer for Isarn," in *IEEE Access*, vol. 14, pp. 45053-45070, 2026. https://doi.org/10.1109/ACCESS.2026.3676694

[25]. John, A., Cardiff, B., John, D. (2026). A Multi-level and Multimodal Data Fusion Algorithm Using Neural Networks. In: Deep Learning and Signal-Processing Methods for Multisensor Data Fusion. Springer, Cham. https://doi.org/10.1007/978-3-031-96724-5_7
[26]. Naik, G.R. (2026). Designing MATLAB App Layouts. In: MATLAB App Development. Springer, Singapore. https://doi.org/10.1007/978-981-95-8797-1_4
[27]. Wong, B.W.L., Samuel, A.G. & Kapnoula, E.C. Speech perception consistency facilitates initial lexical activation, but not speech perception flexibility. *Sci Rep* 16, 16189 (2026). https://doi.org/10.1038/s41598-026-47943-3
[28]. Sachi Gupta, Gaurav Agarwal, Shivani Agarwal, Atul Kumar Rai, Optimizing automatic speech recognition with advanced transformers and metaheuristics, Biomedical Signal Processing and Control, Volume 117, 2026, 109464, ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2026.109464.
[29]. Visalli, M. (2026). Transforming Raw Text into Structured Data: An Overview of Content Analysis and Natural Language Processing Techniques for Sensory and Consumer Science. In: Visalli, M., Galmarini, M.V. (eds) Collection and Analysis of Text Data in Sensory and Consumer Science. Springer, Cham. https://doi.org/10.1007/978-3-032-04851-6_5
[30]. Kobylin, A., Ryazantsev, O. (2026). Processing of Audio Recordings and Feature Extraction for Machine Learning. In: Semenov, A., Yepifanova, I., Kajanová, J. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 272. Springer, Cham. https://doi.org/10.1007/978-3-032-06608-4_9
[31]. Sayood, K. (2026). Signal Representation. In: Biomedical Signal Processing. Synthesis Lectures on Biomedical Engineering. Springer, Cham. https://doi.org/10.1007/978-3-032-08855-0_2
[32]. Bernard J. Baars, Nicole M. Gage, Chapter 11 - Language, Cognition, Brain, and Consciousness (Second Edition), Academic Press, 2010, Pages 370-396, ISBN 9780123750709, https://doi.org/10.1016/B978-0-12-375070-9.00011-5