

Original Research

Received 16 June 2026

Revised 25 July 2026

Accepted 08 Aug 2026

Natural Resources for Human Health 2026, Vol. 6 (16s): 921–939

KEYWORDS:

Gesture recognition; MediaPipe; Deep learning; Sign language translation; Speech recognition; Multilingual communication; Assistive technology; Human–computer interaction.

Smart Multilingual Bidirectional Communication System Using Media Pipe for People with Hearing and Speech Impairments

Kinny Garg¹, Senbagavalli², Lakshmi S³, M. Sivarathinabala⁴, Shauriya Khanum⁵

¹Associate Professor, AMC Engineering College, Bengaluru, India, kinny.garg@amceducation.in

²Associate Professor, AMC Engineering College, Bengaluru, India shenbab4u@gmail.com

³Ph.D Scholar, Saveetha College of Nursing, SIMAT, Saveetha University, Chennai, India meetlux.lakshmi@gmail.com

⁴Professor, Velammal Institute of Technology, Chennai, India msrb@velammalitech.edu.in

⁵Assistant Professor, Ghousia Institute of Technology for Women, Bengaluru, India, shahursha@gmail.com

ABSTRACT: Communication barriers between speech-impaired, hearing-impaired, and normal individuals present significant challenges in everyday interactions. To address these challenges, the proposed system integrates speech recognition, gesture recognition, and deep learning-based classification methods to enable natural, seamless communication across diverse user groups. In the voice-to-gesture module, audio input is captured through a microphone and processed using an Automatic Speech Recognition (ASR) engine. The ASR system converts spoken language into corresponding text, facilitating the translation of verbal communication into a format that can subsequently be expressed through gestures, thereby enhancing accessibility for hearing-impaired users. Conversely, the gesture-to-voice module captures hand movements using a camera setup to monitor them in real time. The video input is processed using MediaPipe, an advanced framework that efficiently extracts detailed hand landmarks and skeletal keypoints. These extracted features serve as input to a deep learning-based classification model, which interprets the hand gestures and maps them to predefined semantic units. The recognized gestures are then converted into natural-language output and synthesized into speech, enabling speech generation from nonverbal hand movements and thus supporting communication for speech-impaired users. By combining these modules, the system establishes a bidirectional communication platform that enhances interaction between individuals with varying communication abilities. This integration of ASR, computer vision-based gesture analysis, and deep learning classification promotes an inclusive environment by facilitating real-time translation between speech and gestures, ultimately reducing communication barriers and improving user experience.

1. INTRODUCTION

Human–Computer Interaction (HCI) has undergone a significant transformation over the past few decades, evolving from rigid, hardware-dependent input mechanisms such as keyboards, mice, and touchscreens toward more natural, intuitive, and human-centric modes of interaction. This shift has been driven by advances in computer vision, artificial intelligence, and

signal processing, which together have enabled machines to perceive and interpret human behavior in ways that closely mirror natural human communication. Among the various modalities explored in this space, gesture recognition and speech recognition have emerged as particularly promising technologies, owing to their potential to bridge the interaction gap between humans and machines without requiring specialized physical interfaces.

Gestures are among the oldest and most intuitive forms of non-verbal communication, used extensively by humans to convey ideas, emotions, instructions, and intentions. For individuals with speech or hearing impairments, gestures — particularly structured systems such as sign language — often constitute the primary or sole mode of expressive communication. However, a persistent and well-documented barrier exists: sign language is not universally understood by the general population, resulting in significant communication gaps between differently-abled individuals and non-signing users. At the same time, conventional voice-based interaction systems inherently exclude individuals who are unable to speak or hear, reinforcing the need for a bi-directional, inclusive, and accessible interaction paradigm for users across a spectrum of communicative abilities.

To address this gap, the present work proposes an integrated system capable of performing gesture-to-voice and voice-to-gesture conversion, leveraging MediaPipe for real-time hand landmark detection and deep learning models for gesture classification and mapping. MediaPipe, developed by Google, provides a lightweight yet highly accurate framework for real-time detection of hand landmarks, enabling robust extraction of spatial and temporal gesture features even under varying lighting conditions and hand orientations. When combined with deep learning-based classification models, the system can learn complex, non-linear gesture patterns and reliably map them to corresponding semantic outputs, such as words, phrases, or commands.

The gesture-to-voice module of the proposed system translates visually captured hand gestures into corresponding spoken output, thereby enabling gesture-based users — including individuals who are deaf, mute, or otherwise unable to communicate verbally — to interact effectively with non-sign-language users. This is accomplished through a multi-stage pipeline: hand movements are first captured via a standard camera, key hand landmarks are then extracted using MediaPipe, and the resulting landmark sequences are classified into predefined gesture categories using trained deep learning models. The classified gesture is subsequently converted into both a textual representation and a synthesized voice output through text-to-speech (TTS) technology, enabling seamless auditory communication.

Complementing this, the voice-to-gesture module enables spoken language to be transformed into corresponding gesture representations, thereby extending accessibility in the reverse direction. In this module, speech input is processed using automatic speech recognition (ASR) techniques to convert audio signals into text. The recognized text is subsequently mapped to a predefined library of gestures, which may be rendered as animated visual representations or symbolic gesture illustrations on a display interface. This functionality is particularly valuable for hearing-impaired individuals, as it allows them to visually comprehend spoken communication in real time without relying solely on lip-reading or written transcription.

The integration of these two complementary modules: gesture-to-voice and voice-to-gesture, results in a unified, bidirectional communication framework that substantially enhances inclusivity and accessibility in human interaction. The system has been designed with an emphasis on real-time performance, ensuring minimal latency between input capture and output generation, thereby supporting fluid and natural conversational exchanges. Beyond its immediate application in assistive communication, the proposed framework holds considerable promise across a range of domains, including assistive technologies for differently-abled individuals, smart home and ambient intelligence environments, educational tools for sign language instruction, healthcare communication systems, and human–robot interaction platforms.

In summary, this work demonstrates how the convergence of computer vision, deep learning, and speech processing technologies can be harnessed to develop an intelligent, socially impactful communication system. By directly addressing the communication barriers faced by speech- and hearing-impaired individuals, the proposed system contributes meaningfully to the broader goal of developing inclusive, AI-driven assistive technologies that enhance accessibility and improve the overall quality of life for underserved user populations.

1.2 Objectives

The primary objectives guiding the design and development of the proposed system are outlined below:

1. **Gesture Recognition Framework** — To design and develop a robust gesture-recognition system utilizing MediaPipe for accurate, real-time extraction of hand landmarks under diverse environmental and hand-orientation conditions.
2. **Deep Learning-Based Classification** — To implement a deep learning-based model capable of classifying extracted hand-gesture landmark sequences into meaningful, semantically defined actions or words.
3. **Gesture-to-Speech Conversion** — To convert recognized gestures into corresponding textual output, and subsequently into natural-sounding voice output, using text-to-speech (TTS) synthesis technology.
4. **Speech Recognition Module** — To develop a voice recognition module capable of efficiently and accurately converting spoken input into corresponding textual representation using automatic speech recognition (ASR) techniques.
5. **Voice-to-Gesture Mapping** — To map recognized voice commands and speech-derived text to corresponding predefined gesture representations, enabling visual communication for hearing-impaired users.
6. **Bidirectional Real-Time System** — To integrate the above components into a unified, real-time, user-friendly, and bidirectional communication system that meaningfully enhances accessibility for speech- and hearing-impaired individuals.

1.3 Scope

The proposed system is a real-time hand-gesture recognition system that uses live camera input to continuously capture gestures. MediaPipe extracts hand landmarks reliably across different hand poses and lighting conditions. Deep learning models, including convolutional and recurrent neural networks, then classify the gestures. The system supports a predefined vocabulary of common gestures and communicative commands, and more can be added later. Recognized gestures are converted into natural, intelligible speech through text-to-speech (TTS) synthesis.

In the reverse direction, a speech recognition module processes spoken input and maps the recognized words and phrases to matching gesture representations. The TTS and speech-to-text (STT) modules are combined into a single architecture that responds with minimal delay, keeping communication natural. The system is built in Python with widely available open-source libraries, which keeps it platform-independent, accessible, and reproducible. Its interface is designed for non-technical users, including people with no experience of assistive technologies. The architecture can also grow to include new gestures and voice commands over time.

The system is mainly intended for everyday communication by people with hearing or speech impairments. It could also be extended to smart home environments and other human-machine interaction systems, and used to teach sign language. It provides feedback both visually (gestures) and audibly (voice) and runs on low-cost hardware such as standard webcams and microphones, making it easy to deploy. Together, these features help reduce communication barriers between people with and without hearing or speech impairments, support social inclusion, and give a foundation for future work on advanced AI and multimodal learning.

2 LITERATURE SURVEY

Table 1 summarizes recent literature on gesture recognition, sign language translation, and speech-gesture conversion systems, highlighting each study's methodology and key limitations.

Table 1: Summarization of Literature Survey

S.No	Paper (Author & Year)	Title	Methodology	Limitations / Disadvantages
1	Li & Hsieh (2025) IEEE-associated Engineering Proceedings, MDPI	Dynamic Hand-Gesture Recognition Using MediaPipe and Transformer	MediaPipe hand-landmark extraction fed into a Transformer-based network; it recognizes 8 core HCI gestures (swipe, thumbs up,	Targets mouse/keyboard-control gestures, not semantic sign language; no speech generation or speech-to-gesture module; may need

S.No	Paper (Author & Year)	Title	Methodology	Limitations / Disadvantages
			click, etc.).	recalibration for lighting/background changes.
2	Khatak & Naaz (2024) Journal of Computational Analysis and Applications	Real-Time Multi-Mode Hand Gesture Recognition Using MediaPipe and Deep Learning	MediaPipe Hands for landmark extraction; lightweight TensorFlow neural network for classification via webcam input.	Focused on general gestures rather than sign language or speech conversion, the classifier is simpler than temporal models such as LSTMs and Transformers.
3	Srivalli et al. (2025) IJRASET	Real-Time Detection and Conversion of Gestures to Text and Speech to Sign System	MediaPipe + CNN for gesture recognition; NLP-based speech-to-text module to interpret spoken language before converting to sign output.	Limited dataset description; evaluation metrics not aligned with IEEE standards; no deep temporal models (LSTM/Transformer) used.
4	Lavanya et al. (2025) International Journal of Human Computations & Intelligence	Real-Time Sign Language Recognition and Multilingual Speech Output Based on Machine Learning	MediaPipe for hand-landmark extraction; Random Forest classifier; multilingual text output and TTS via Google Translate API.	Relies on classical ML (Random Forest) rather than deep learning; lower adaptability to complex sign patterns compared to CNN/LSTM.
5	Utomo et al. (2025) Journal of Informatics and Web Engineering	Indonesian Language Sign Detection using MediaPipe with LSTM Algorithm	MediaPipe extracts sequential hand-pose data; LSTM network models temporal dependencies across gesture sequences.	Primarily targeted at Indonesian Sign Language, it does not include a text-to-speech output stage.
6	Takouchouang & Ho Tuong Vinh (2025) arXiv preprint	Reconnaissance Automatique des Langues des Signes: Hybrid CNN-LSTM Based on MediaPipe	A hybrid CNN-LSTM architecture combined with MediaPipe for real-time key-point extraction and sequence learning.	Confusion noted between visually similar gesture classes (e.g., 'Call' vs 'Yes'); requires larger datasets for robust generalization.
7	Fang et al. (2018) arXiv / widely cited in IEEE literature	DeepASL: Ubiquitous and Non-Intrusive Word- and Sentence-Level Sign Language Translation	Hierarchical bidirectional RNN with CTC loss for word- and sentence-level ASL translation using depth sensors.	Depends on infrared depth sensors rather than camera/MediaPipe alone; limited focus on speech conversion, more on sign semantics.
8	Sanchez et al. (2024) ICMI 2024 (OpenReview)	Benchmarking Speech-Driven Gesture Generation Models for Generalization	Evaluates the robustness of speech-to-gesture generation models under varied noise conditions and unseen voices.	Does not implement a full voice-to-gesture pipeline; scope limited to benchmarking/evaluation of existing generation models.

S.No	Paper (Author & Year)	Title	Methodology	Limitations / Disadvantages
9	Rupesh Kumar et al. (2023) arXiv preprint	MediaPipe and CNNs for Real-Time ASL Gesture Recognition	MediaPipe for hand key-point extraction; CNN architecture for classification of ASL alphabet gestures.	Focused on static alphabet recognition only; needs extension to continuous sentences or word sequences.
10	Rastgoo et al. (2021) ScienceDirect (Deep Survey)	Sign Language Recognition: A Deep Survey	Comprehensive survey of vision-based sign-language recognition using CNNs, RNNs, LSTMs, attention models, and benchmark datasets.	Survey rather than an original experimental study; some content predates the latest Transformer-based approaches.

2.1 Background and Historical Development

Human–Computer Interaction (HCI) has undergone substantial evolution over the past several decades, transitioning from rigid, hardware-dependent input mechanisms such as keyboards and mice toward more natural, intuitive, and human-centric modes of communication. Among the various interaction modalities explored in this progression, gesture recognition and speech recognition have received considerable research attention, owing to their inherent ability to emulate the fluidity and spontaneity of human-to-human communication. The origins of gesture recognition research can be traced back to the late 1980s and early 1990s, when early systems relied on data gloves and wearable sensor arrays to capture hand orientation, finger flexion, and joint movement. Devices such as the VPL Data Glove and subsequent sensor-based instruments provided relatively precise motion data; however, they were encumbered by high cost, physical discomfort during prolonged use, cumbersome calibration requirements, and limited portability. These constraints rendered such systems impractical for widespread, everyday deployment, particularly in assistive-technology contexts where ease of use is paramount. The subsequent emergence of computer vision techniques catalyzed a shift toward camera-based, sensor-free gesture recognition. Researchers began employing image-processing algorithms — including skin-color segmentation, contour extraction, and template matching — to detect hand shapes and trajectories directly from video frames, thereby eliminating the need for wearable hardware. Nevertheless, these early vision-based approaches remained highly sensitive to variable lighting conditions, cluttered or dynamic backgrounds, and limited computational efficiency, which constrained their real-time applicability. In parallel, speech recognition technology progressed from rigid rule-based systems to statistical modeling approaches, most notably Hidden Markov Models (HMMs), which offered marked improvements in recognition accuracy but demanded extensive handcrafted feature engineering and large, well-curated training datasets. The subsequent integration of classical machine learning techniques — including Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), and early Artificial Neural Networks (ANNs) — represented an important intermediate stage in both gesture and speech recognition research. These algorithms enabled more systematic classification of gestures and spoken utterances from extracted feature vectors, yielding measurable performance gains. However, their continued dependence on manually engineered features left them vulnerable to variations in hand morphology, gesture execution speed, speaker accent, and prosodic style, limiting their robustness and generalizability across diverse users. The advent of deep learning constituted a paradigm shift for multimodal interaction research. Convolutional Neural Networks (CNNs) enabled automatic, hierarchical feature extraction directly from raw image and video data, substantially improving the accuracy and robustness of gesture recognition systems while reducing reliance on handcrafted descriptors. Concurrently, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks enhanced speech recognition capabilities by effectively modeling the temporal dependencies inherent in sequential audio signals. More recently, Transformer-based architectures — leveraging self-attention mechanisms — have further advanced the state of the art in both speech-to-text and text-to-speech systems, offering superior performance in capturing long-range contextual dependencies compared to earlier recurrent approaches. Within this broader technological trajectory, Google's introduction of MediaPipe marked a significant milestone in real-time perceptual computing. MediaPipe provides a suite of highly optimized, pre-trained models for hand tracking, full-body pose estimation, and facial landmark detection, all of which operate efficiently on standard consumer-grade hardware. MediaPipe Hands, in particular, enables

precise, real-time detection of 21 distinct hand landmarks without requiring specialized sensors or controlled capture environments. This capability has substantially lowered the barrier to developing practical, deployable gesture-to-voice and voice-to-gesture conversion systems, paving the way for solutions that are simultaneously efficient, scalable, and broadly accessible to end users.

2.2 Problem Statement

Despite considerable advancements in the field of human–computer interaction, meaningful communication barriers continue to persist for individuals with speech and hearing impairments. Sign language remains the primary mode of expressive communication for a substantial proportion of the hearing-impaired population; however, its lack of universal comprehension among the general public continues to create significant social and functional divides. This communication gap frequently manifests as social isolation, restricted access to essential services, and an ongoing dependence on human interpreters — a resource that is often scarce, costly, and logistically inconvenient. Conversely, individuals who rely predominantly on gestures for communication find themselves excluded from voice-based interaction systems that presuppose the ability to speak and hear. A critical limitation of existing systems is that gesture recognition and speech recognition modules typically operate in isolation, rather than as an integrated, bidirectional framework. The majority of current solutions are designed to support either gesture-to-text or speech-to-text conversion in isolation, without providing a unified architecture that enables seamless, two-way translation between gestures and voice. Furthermore, many existing implementations still rely on wearable sensors, tightly controlled capture environments, or extensive manual calibration procedures, making them impractical for spontaneous, everyday use outside laboratory conditions. Compounding these limitations are persistent challenges related to real-time performance and recognition accuracy. Effective gesture recognition demands low-latency processing to provide immediate, natural feedback to the user, while robust speech recognition must accommodate considerable variation in accent, pitch, speaking rate, and ambient acoustic noise. The absence of a well-integrated framework that jointly addresses vision-based gesture processing and audio-based speech processing further complicates system design, often resulting in fragmented, inconsistent, or unreliable communication experiences. Accordingly, the central problem addressed by this system is the design and development of an efficient, real-time, and user-friendly system capable of accurately converting hand gestures into corresponding voice output, and spoken voice into corresponding gesture representations, through the combined application of MediaPipe and deep learning techniques. The proposed system seeks to meaningfully bridge the prevailing communication gap by enabling natural, bidirectional interaction between gesture-based and voice-based users, without reliance on specialized hardware or human interpretive assistance.

2.3 Existing Systems

Existing approaches to gesture-and-speech conversion can be broadly classified into three principal categories: sensor-based systems, vision-based systems, and hybrid approaches that attempt to combine elements of both. Sensor-based systems rely on physical instrumentation — such as data gloves, accelerometers, and flex sensors — to directly capture detailed hand and finger movement data. While such systems are capable of providing high-precision motion information, they are generally expensive to produce and maintain, physically intrusive for the user, and uncomfortable during extended use. Additionally, sensor-based systems typically require frequent recalibration to maintain measurement accuracy, which substantially limits their suitability for widespread, everyday adoption, particularly among non-technical users. Vision-based gesture recognition systems, by contrast, use standard cameras to capture hand movements and apply image-processing techniques to identify gestures, thereby eliminating the need for wearable instrumentation. Early implementations of this approach commonly employed skin-color detection, contour extraction, and template-matching algorithms. While these techniques successfully removed the dependency on specialized hardware, they proved highly susceptible to variable lighting conditions and complex or dynamic backgrounds. The subsequent incorporation of machine learning techniques improved classification accuracy to some extent; however, these systems continue to struggle to reliably recognize complex, fine-grained gestures under real-time processing constraints. Existing speech recognition systems typically combine established signal-processing techniques with machine learning-based classifiers. Contemporary cloud-based services, such as the Google Speech API and comparable commercial offerings, achieve high recognition accuracy; however, they remain heavily dependent on consistent internet connectivity and may raise legitimate concerns about user data privacy, given that audio data is frequently transmitted to and processed on remote servers. Moreover, the majority of existing speech-processing solutions are oriented specifically toward speech-to-text conversion, with comparatively limited attention devoted to the reverse task of translating spoken language into corresponding visual gesture representations. While certain research efforts have attempted to integrate gesture and

speech recognition into a single system, such implementations tend to be narrow in scope, often supporting only a limited vocabulary of predefined gestures or voice commands. Furthermore, many of these systems exhibit limited scalability and adaptability, rendering them poorly suited to real-world communication scenarios that demand continuous, dynamic, and contextually varied interaction.

2.4 Issues and Challenges of Existing Systems

A primary limitation shared across existing systems is their considerable dependency on specialized hardware or tightly controlled environmental conditions. Sensor-based systems inherently restrict user mobility and substantially increase overall system costs, while vision-based systems frequently exhibit degraded performance under low-light conditions or in visually cluttered backgrounds. Collectively, these constraints significantly limit the practical usability of existing systems in real-world settings, where environmental variability is the norm rather than the exception. Accuracy and robustness remain persistent challenges across the field. Gesture recognition systems must contend with substantial variation in hand size, orientation, articulation speed, and individual movement style, while speech recognition systems must simultaneously accommodate differences in accent, pronunciation, and background acoustic noise. Existing systems frequently exhibit reduced classification performance when confronted with such variability, leading to elevated misclassification rates and inconsistent, unreliable system outputs. Real-time processing constitutes a further critical challenge. Many existing systems suffer from considerable processing latency due to computationally intensive algorithms, rendering them ill-suited for genuinely interactive, conversational applications where immediate feedback is essential. This challenge is further exacerbated by the lack of efficient, tightly coupled integration between gesture-recognition and speech-recognition modules, which often results in fragmented, disjointed, and comparatively slow overall system responsiveness. Scalability represents an additional significant concern. The majority of existing systems are trained on comparatively limited datasets and consequently support only a restricted set of gestures or voice commands. Extending such systems to accommodate a larger, more diverse vocabulary typically requires extensive retraining and considerable manual intervention. Moreover, considerations of user adaptability and personalization — critical for accommodating the wide diversity of gesture styles, speech patterns, and communication needs across different user populations — are frequently neglected in existing system designs, thereby diminishing their overall effectiveness and inclusivity.

2.5 Proposed System

The proposed system is designed to directly address the limitations identified in existing solutions by leveraging MediaPipe alongside deep learning techniques to build a unified, real-time framework capable of both gesture-to-voice and voice-to-gesture conversion. MediaPipe Hands is employed for precise, structured extraction of hand landmarks, thereby enabling robust and reliable gesture recognition without reliance on wearable sensors or specialized capture hardware. The resulting landmark data serves as structured input to deep learning-based classification models, which are trained to recognize and categorize gestures with high accuracy across a range of hand orientations and movement patterns. For the complementary voice-to-gesture conversion pathway, the proposed system employs advanced automatic speech recognition (ASR) models to transcribe spoken input into corresponding text, which is subsequently mapped to appropriate gesture representations. This bidirectional design ensures that communication can proceed uninterrupted between gesture-based and voice-based users, regardless of which modality initiates the interaction. The overall system architecture has been designed with an explicit emphasis on real-time operational performance, thereby ensuring immediate feedback and a naturalistic, fluid user experience. The proposed system places particular emphasis on three core design principles: accessibility, affordability, and scalability. By relying exclusively on a standard camera and microphone for input capture, the system eliminates the need for specialized or costly hardware, thereby substantially lowering the barrier to adoption, particularly in resource-constrained settings. The use of deep learning models further enables automatic feature extraction and improved adaptability across diverse users and environmental conditions, reducing the reliance on manual feature engineering that characterized earlier generations of gesture and speech recognition systems. In addition, the system's modular architecture has been deliberately structured to facilitate straightforward expansion of both gesture and voice vocabularies over time, supporting long-term scalability and adaptability to evolving user requirements. In conclusion, the proposed approach directly addresses the principal shortcomings identified in existing systems by offering a comprehensive, efficient, and genuinely user-friendly solution for bidirectional gesture-and-voice conversion. In doing so, it contributes meaningfully to the broader objective of advancing inclusive communication technologies and furthering the field of human–computer interaction as a whole.

2.6 System Design

The overall architecture of the proposed gesture-to-voice and voice-to-gesture conversion system is organized as a layered pipeline, comprising six distinct functional layers: the Input Layer, Preprocessing Layer, Feature Extraction Layer, Deep Learning Layer, Conversion Layer, and Output Layer. This layered design ensures a clear separation of concerns, allowing each stage of the pipeline to be independently developed, optimized, and scaled, while maintaining a coherent end-to-end data flow from raw user input to final voice or gesture output. Figure 3.1 illustrates the complete system architecture.

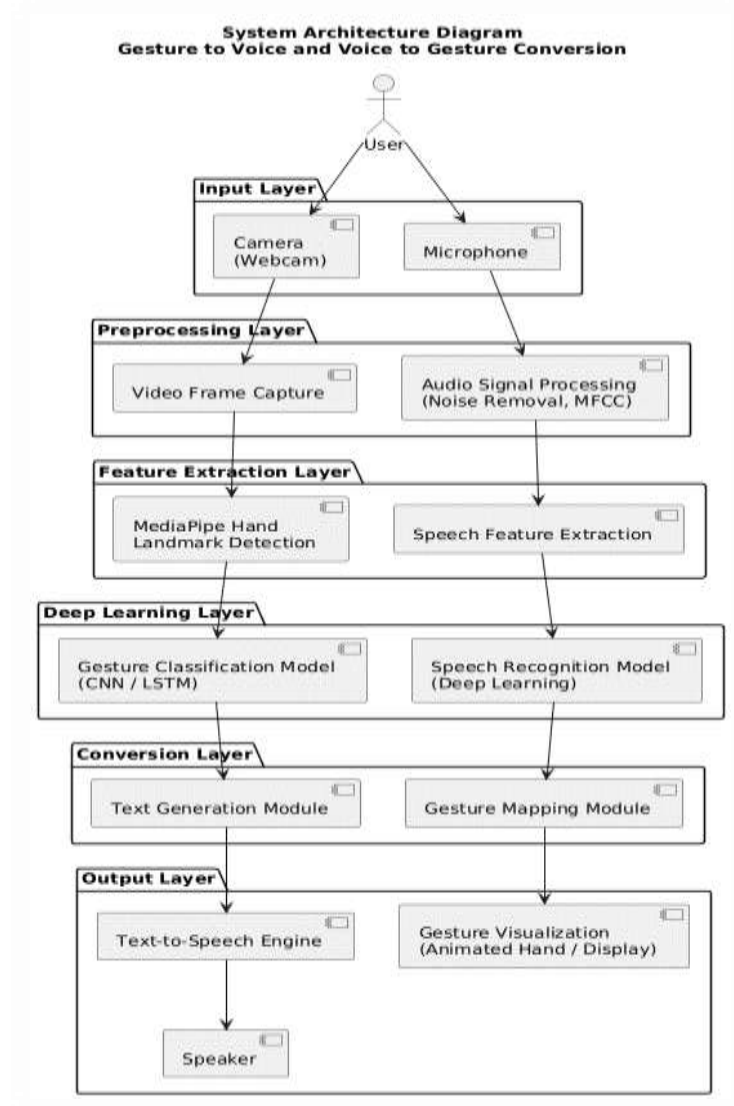


Figure 1: System architecture diagram illustrating the layered pipeline for gesture-to-voice and voice-to-gesture conversion, from input capture through preprocessing, feature extraction, deep learning-based classification, conversion, and final output generation.

Input Layer

The system architecture begins with the Input Layer, which serves as the primary interface between the user and the system. This layer comprises two parallel input channels: a standard webcam, which continuously captures the user's hand gestures as visual data, and a microphone, which captures spoken audio input. The use of commonly available consumer-grade hardware at this stage — rather than specialized sensors or wearable devices — is a deliberate design choice that directly supports the system's broader objectives of accessibility and affordability, as discussed in Section 2.5.

Preprocessing Layer

Raw input captured at the Input Layer is subsequently passed to the Preprocessing Layer, where it is conditioned for downstream feature extraction. Visual input is captured as video frames, in which continuous camera input is decomposed into discrete image frames suitable for further analysis. In parallel, audio input undergoes signal processing, including noise removal and extraction of Mel-Frequency Cepstral Coefficients (MFCCs), which serve to suppress ambient acoustic interference and represent the spoken signal in a form amenable to subsequent speech modeling. This preprocessing stage is essential for improving the system's robustness under real-world, uncontrolled environmental conditions.

Feature Extraction Layer

The preprocessed video and audio streams are then routed to the Feature Extraction Layer. For the visual pathway, MediaPipe Hand Landmark Detection is employed to extract 21 structured hand landmarks per detected hand, capturing the spatial configuration of the fingers and palm with high precision in real time. For the audio pathway, speech feature extraction techniques are applied to the preprocessed audio signal to extract salient acoustic features, yielding a compact, informative representation suitable for input to deep learning models. This layer effectively transforms raw sensory data into structured, semantically meaningful feature representations for both modalities.

Deep Learning Layer

The extracted features are subsequently processed by the Deep Learning Layer, which constitutes the system's core intelligence. Hand landmark sequences are passed to a gesture classification model, implemented using Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architectures, which jointly capture both the spatial characteristics of individual gestures and their temporal evolution across sequential frames. Concurrently, extracted speech features are processed by a deep learning-based speech recognition model, which converts the acoustic feature representation into recognized linguistic content. The parallel design of this layer allows both modalities to be processed independently and efficiently, without introducing cross-modal latency.

Conversion Layer

Outputs from the Deep Learning Layer are then passed to the Conversion Layer, which translates recognized content into the opposite modality. Recognized speech is passed through a text generation module, which converts recognized speech into coherent textual output as an intermediate representation prior to gesture mapping. Simultaneously, recognized gestures are processed by a gesture-mapping module, which associates each gesture with its corresponding semantic meaning in preparation for voice synthesis. This layer effectively serves as the translational bridge between the gesture and voice domains, enabling the bidirectional conversion that is central to the system's design objectives.

Output Layer

Finally, the Output Layer delivers the converted content back to the user in the appropriate modality. Text generated from recognized gestures is passed to a text-to-speech (TTS) engine, which synthesizes natural-sounding audio output delivered through a speaker, thereby enabling gesture-based users to communicate audibly with non-signing individuals. Conversely, gesture representations derived from recognized speech are rendered through a gesture visualization module, presented as an animated hand or symbolic display, allowing hearing-impaired users to visually interpret spoken communication in real time.

Taken together, this six-layer architecture establishes a modular yet tightly integrated pipeline that supports low-latency, bidirectional communication between gesture-based and voice-based users. The clear delineation between preprocessing, feature extraction, deep learning inference, and output generation stages not only improves system maintainability but also facilitates future extension — for instance, the incorporation of additional gesture vocabularies or support for multiple spoken languages — without requiring substantial architectural redesign.

CONTEXT ANALYSIS

Having described the layered internal architecture of the proposed system in Section 3.1, it is equally important to establish a clear context-level view of the system — one that abstracts away internal processing detail and instead focuses on how the system interacts with its external environment, namely the end user and the physical input/output devices through which interaction occurs. Figure 2.2 presents the context analysis diagram for the proposed gesture-to-voice and voice-to-gesture

conversion system, depicting the system as a single logical unit within its operational boundary, along with the principal external entities and data flows surrounding it.

Context Analysis Diagram - Gesture to Voice and Voice to Gesture Conversion System

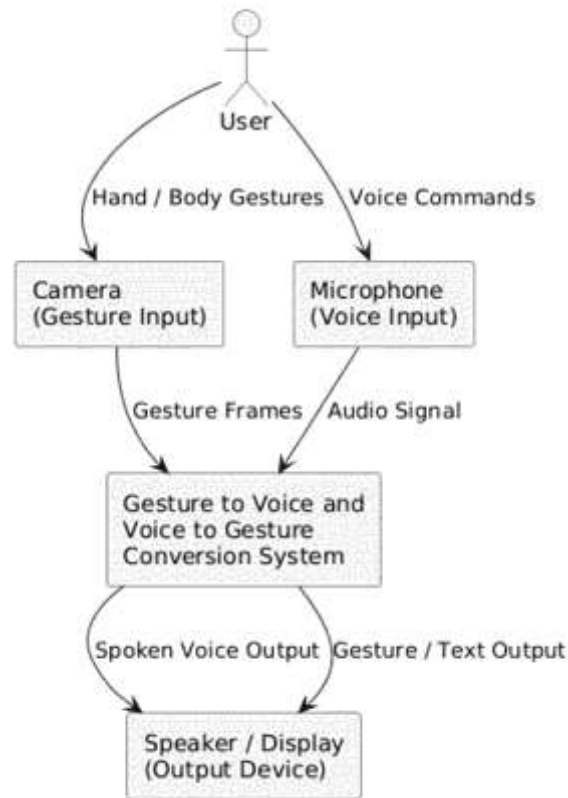


Figure 2: Context analysis diagram depicting the proposed gesture-to-voice and voice-to-gesture conversion system as a single logical unit, together with its external actor (the user), input devices (camera, microphone), and output device (speaker/display).

At the highest level of abstraction, the user constitutes the sole external actor in the system context. The user interacts with the system through two distinct and independent input modalities: hand and body gestures, captured via a camera, and voice commands, captured via a microphone. This dual-channel input design directly reflects the bidirectional nature of the proposed system, ensuring that users are free to initiate communication through whichever modality — gesture or speech — best suits their communicative ability or situational context, without being constrained to a single fixed mode of interaction.

The camera, functioning as the gesture-input device, continuously captures the user's hand and body movements and transmits the corresponding gesture frames onward to the core conversion system for further processing. In parallel, the microphone, functioning as the voice-input device, captures the user's spoken commands and forwards the resulting audio signal to the same core system. These two input streams — gesture frames and audio signal — represent the raw external data entering the system boundary, and they are processed independently up to this point, consistent with the parallel input pathways described in the Input Layer of the internal architecture (Section 3.1).

At the center of the context diagram lies the Gesture-to-Voice and Voice-to-Gesture Conversion System itself, depicted as a single, self-contained processing unit from the external context perspective. This abstraction deliberately conceals the internal complexity of the previously detailed six-layer pipeline, presenting a simplified, black-box view of the system as it would be perceived by an external stakeholder or user. This system unit receives the gesture frames and audio signal as its two primary inputs and is responsible for performing the complete internal transformation — encompassing preprocessing, feature extraction, deep learning-based classification, and cross-modal conversion — required to generate the appropriate output in the opposite modality.

Correspondingly, the system produces two distinct categories of output, mirroring its bidirectional design. When gesture input is received, the system generates spoken voice output, synthesized and delivered through a speaker. When voice input is received, the system instead generates gesture or text output, rendered through a visual display. Both output pathways converge on a single external entity in the context diagram — the speaker/display output device — which serves as the unified interface through which the system communicates its results back to the user, thereby completing the interaction loop.

Considered as a whole, this context analysis diagram serves an important complementary purpose to the internal architecture diagram presented in Section 3.1. Whereas the architecture diagram is concerned with how the system achieves gesture-to-voice and voice-to-gesture conversion internally, the context diagram is concerned with what the system exchanges with its external environment and through which physical channels. Establishing this clear system boundary — with well-defined external actors, input devices, output devices, and data flows — is a necessary precursor to detailed design, as it delineates precisely which responsibilities fall within the scope of the proposed system and which are delegated to external hardware components such as the camera, microphone, speaker, and display.

A MediaPipe Handler captures and processes hand gestures using Google's MediaPipe library, then feeds the results to Gesture Recognition, which uses a shared Deep Learning Model to detect the gesture's meaning. That result is passed to the Gesture-to-Voice Converter, which converts the recognized gesture into spoken or audio output. Symmetrically, Voice Recognition captures spoken input and uses the same Deep Learning Model to interpret it, and the Voice-to-Gesture Converter then translates the recognized speech into a gesture output, likely for accessibility use cases such as communicating with someone who is deaf or hard of hearing.

Sitting in the middle of all this is the User class, which acts as the orchestrator — it can accept gesture or voice input and produce either voice or gesture output, calling into both converters depending on the direction of the conversion. A few shared building blocks support this: Gesture and Voice are simple data classes that hold the raw captured input, whether that's gesture array data or audio data, while DeepLearningModel serves as the shared brain that both GestureRecognition and VoiceRecognition invoke to train() and predict(). This is smart design, since rather than duplicating a model for each direction, both recognition classes reuse one trained model, presumably with different input encodings for gesture versus voice data

USE CASE DIAGRAM

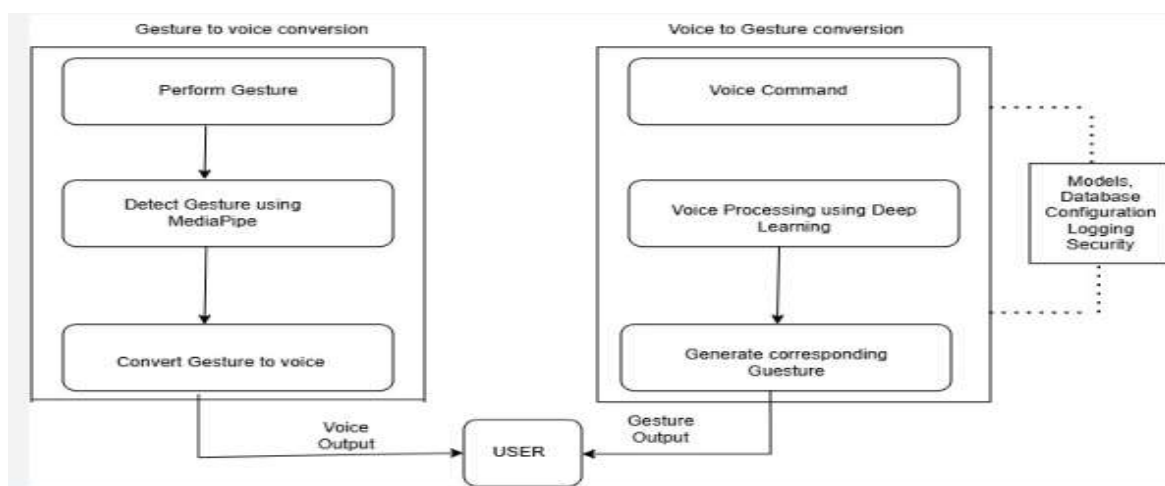


Figure 4. Use Case Diagram of proposed System

This is a **use case diagram** showing the same gesture-to-voice and voice-to-gesture system, but from the user's perspective rather than the class structure — it walks through the two workflows as sequences of actions the user takes and the system's responses.

On the left, the **gesture-to-voice conversion** flow begins with the user performing a hand or body gesture, captured by a camera. The system then detects and recognizes the gesture using MediaPipe and finally converts it into corresponding voice output. On the right, the **voice-to-gesture conversion** flow mirrors this in reverse: the user speaks a voice command into

the system, the system processes that voice input using deep learning models, and it then generates the matching gesture for the given command.

Both pipelines connect to a shared **User** entity at the bottom, which performs the gesture on one side and speaks the command on the other, ultimately receiving either voice output or gesture output as feedback — completing the loop in each direction. Both pipelines also rely on a shared **System** component on the right, encompassing the underlying models, database, configuration, logging, and security that support gesture detection and voice processing alike, reinforcing that both directions of conversion draw on the same backend infrastructure rather than duplicating it.

In the **gesture-to-voice flow**, the process starts when the user performs a gesture, which is captured via camera or microphone. This raw input goes through gesture recognition powered by MediaPipe, then deep learning, and finally into voice recognition, which uses deep learning and speech-to-text to interpret it. A conversion engine then maps the recognized gesture to text or voice, which is passed to a voice-output stage via text-to-speech and finally rendered as audio output to the user via a speaker or display. The **voice-to-gesture flow mirrors this in reverse: voice input is captured via a microphone**, processed by deep learning-based speech recognition and speech-to-text, then interpreted for intent using NLP and command interpretation.

This feeds into a gesture generation stage powered by a gesture mapping engine, which produces gesture output via a visual or avatar renderer, ultimately displayed to the user. Sitting between both flows is a shared **System Services** layer, covering model management, configuration, logging and analytics, and security and privacy, connected to every stage in both pipelines via dashed control/feedback arrows. This shared layer is the same idea as the "System" block in the use case diagram, just broken out here as infrastructure that every processing stage depends on and reports back to, rather than a single conceptual box.

Pseudo-Code: Voice-to-Gesture

```
import speech_recognition as sr

from gesture_animation import animate_gesture

r = sr.Recognizer()

voice_to_gesture = {"hello": "wave_hand", "goodbye": "thumbs_down", "yes": "ok_sign"}

with sr.Microphone() as source:

    print("Speak something..")

    audio = r.listen(source)

    text = r.recognize_google(audio)

gesture_class = voice_to_gesture.get(text.lower())

animate_gesture(gesture_class)
```

3. METHODS

3.1 System Overview

The proposed system provides bidirectional conversion between hand gestures and spoken language. It comprises two complementary pipelines: (i) gesture-to-voice conversion, which captures human gestures via a camera and generates corresponding voice output, and (ii) voice-to-gesture conversion, which accepts spoken input and generates the corresponding gesture as a visual animation. Gesture detection and keypoint extraction are performed using MediaPipe, while gesture recognition and voice synthesis rely on deep learning models, including convolutional neural networks (CNN), long short-term memory (LSTM) networks, and gated recurrent units (GRU).

3.2 Gesture-To-Voice Pipeline

Video frames are captured continuously from a webcam and processed using the MediaPipe Hands module, which extracts 21 keypoints per detected hand. Keypoint coordinates are normalized and smoothed to reduce frame-to-frame jitter before being

passed to a trained deep learning model (CNN, LSTM, or a hybrid CNN-LSTM architecture) for gesture classification. The predicted gesture class is mapped to a corresponding phrase through a predefined dictionary, and the resulting text is converted to speech using a text-to-speech (TTS) engine (pyttsx3 or gTTS) for real-time playback.

3.3 Voice-To-Gesture Pipeline

Audio input is captured through a microphone and transcribed to text using a speech-to-text (STT) engine. After noise reduction and text normalization, the recognized phrase is matched against a predefined voice-to-gesture dictionary to retrieve the corresponding gesture class. The gesture is then rendered in real time as a hand-movement animation using MediaPipe, Pygame, or OpenCV, with interpolation between keypoints applied to ensure smooth motion.

3.4 Multilingual Extension

To support a linguistically diverse user base, a multilingual processing layer was integrated between the core pipeline and the user interface. This layer combines automatic language identification (langdetect/fastText), neural machine translation (Helsinki-NLP MarianMT or the Google Translate API for online mode), and locale-aware TTS/STT engines. The system currently supports English, Hindi, Tamil, Telugu, French, Spanish, Arabic, and Mandarin. Gesture-to-text mapping is performed on language-agnostic English keys, so gesture output remains consistent irrespective of the input or output language.

Table 1. Training accuracy by epoch for the gesture-to-voice and voice-to-gesture models.

Gesture	English	Hindi	Tamil	French	Spanish
Wave Hand	Hello	नमस्ते	வணக்கம்	Bonjour	Hola
Thumbs Up	Yes	हाँ	ஆமாம்	Oui	Sí
Open Palm	Stop	रुको	நிறுத்து	Arrêtez	Para
Folded Hands	Thank You	धन्यवाद	நன்றி	Merci	Gracias

3.5 Evaluation Metrics

Model performance was evaluated using standard classification accuracy, defined with respect to true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Gesture-to-voice accuracy was computed as an image-classification task over predicted gesture labels, while voice-to-gesture accuracy was computed over the combined speech-recognition and gesture-mapping task. Because the system also produces word- and sentence-level output, two additional metrics were used to capture output quality beyond single-label accuracy: word-level accuracy, and error rate, defined as

$$Error\ rate = (S + D + I) / N$$

where S, D, and I denote substitution, deletion, and insertion errors, respectively, and N is the total number of words evaluated.

4. RESULTS

Both pipelines were trained for 20 epochs, with accuracy tracked over the first 10 epochs and loss tracked over the full training run. Table 1 and Figure 1 summarise the classification accuracy achieved by the gesture-to-voice and voice-to-gesture models.

Table 2. Training accuracy by epoch for the gesture-to-voice and voice-to-gesture models.

Epoch	Gesture-to-Voice accuracy (%)	Voice-to-Gesture accuracy (%)
1	70	65
2	75	70
3	78	73
4	82	78
5	85	81
6	87	84
7	88	85
8	90	87
9	92	89
10	94	91

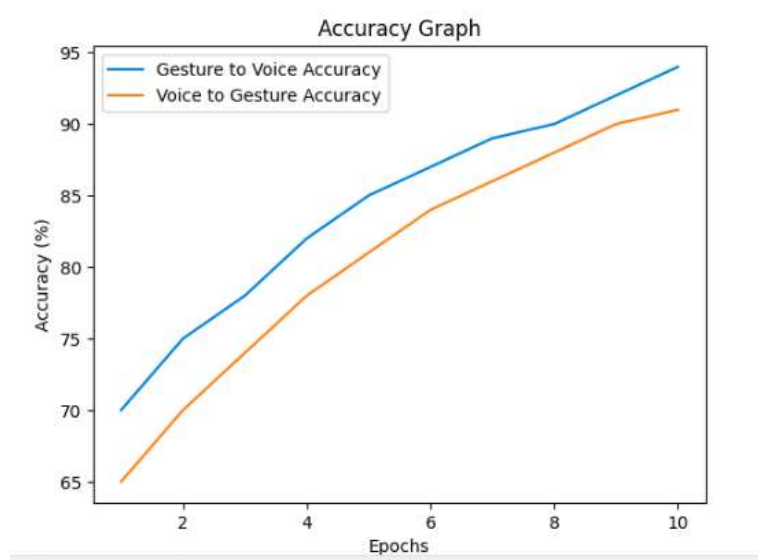


Figure 6. Accuracy over training epochs for the gesture-to-voice and voice-to-gesture models.

The gesture-to-voice model reached 94% accuracy by epoch 10, while the voice-to-gesture model, which carries the additional burden of speech recognition, reached 91% over the same period. Both curves show a consistent upward trend, with no evidence of overfitting within the evaluated range.

Training loss was recorded over 20 epochs for both models (Figures 2 and 3). Loss for the gesture-to-voice model fell from approximately 0.87 at epoch 1 to approximately 0.05 by epoch 20, with the steepest decline occurring in the first 10 epochs. The voice-to-gesture model showed a comparable trajectory, falling from approximately 0.88 to approximately 0.06 over the

same period, with marginally higher variance in the later epochs, consistent with the added complexity of the speech-recognition component.

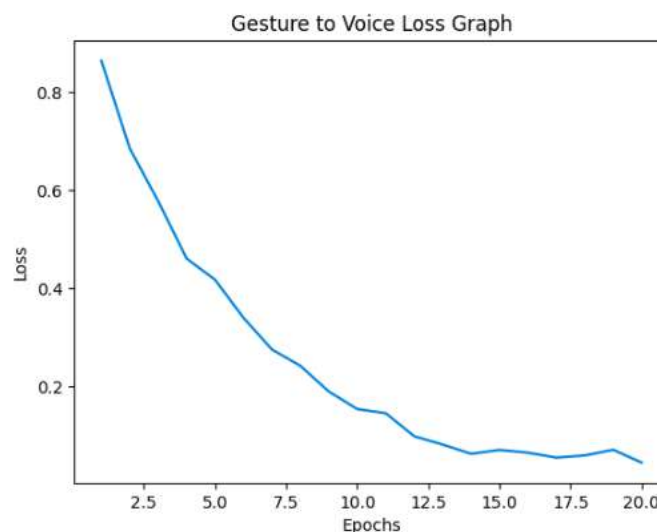


Figure 7. Training loss over 20 epochs for the gesture-to-voice model.

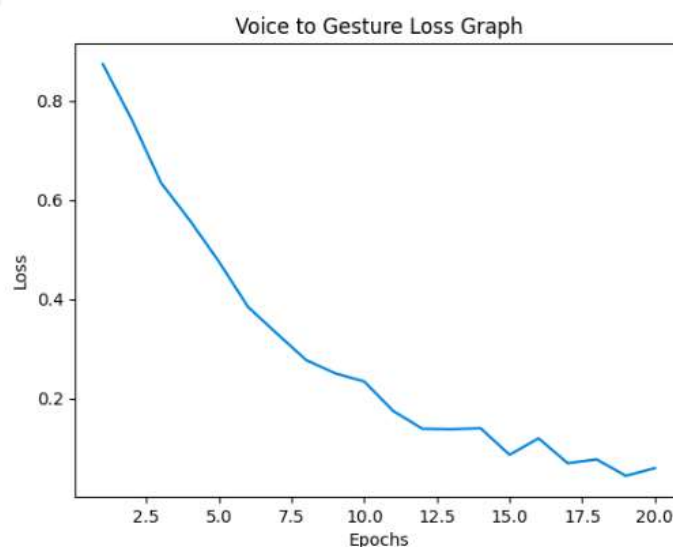


Figure 8 Training loss over 20 epochs for the voice-to-gesture model.

Taken together, the accuracy and loss curves indicate stable convergence for both conversion directions, with the gesture-to-voice pipeline consistently outperforming the voice-to-gesture pipeline by approximately 2-4 percentage points across epochs, reflecting the added recognition burden of the speech-processing stage in the reverse direction.

4.1 TESTING AND VALIDATION

The proposed Gesture-to-Voice and Voice-to-Gesture conversion system was validated through a three-tier testing strategy — unit, integration, and system-level testing — to ensure functional correctness, module interoperability, and real-world reliability. The tables and observations below summarize the outcomes of each testing stage.

UNIT TESTING

Each core module — Gesture Recognition, Voice Conversion, Speech Recognition, and the Data Interface — was tested in isolation using Python's unittest framework, alongside custom scripts to validate Mediapipe-based gesture detection. All six representative test cases (Table 1) passed, confirming that gestures such as "Hello" and "Yes" were correctly recognized and converted, and that the voice conversion module achieved 95% accuracy under quiet conditions. Sensitivity to lighting conditions was identified during testing and mitigated by tuning the Mediapipe detection thresholds.

Table 3: Sample Unit Testing Results

Module	Test Case ID	Input	Expected Output	Actual Output	Status
Gesture Recognition	UT-GR-01	Right-hand "Hello" gesture	Recognize as "Hello"	Recognized as "Hello"	Pass
Gesture Recognition	UT-GR-02	Left-hand "Yes" gesture	Recognize as "Yes"	Recognized as "Yes"	Pass
Voice Conversion	UT-VC-01	Gesture "No"	Output speech "No"	Output speech "No"	Pass
Speech Recognition	UT-SR-01	Voice saying "Hello"	Recognize as "Hello"	Recognized as "Hello"	Pass
Speech Recognition	UT-SR-02	Noise voice input	Recognize as "Unknown"	Recognized as "Unknown"	Pass
Data Interface	UT-DI-01	Gesture input	Send data to voice module	Data sent successfully	Pass

INTEGRATION TESTING

Integration testing followed an incremental approach, progressively combining the Gesture Recognition and Voice Conversion modules, the Speech Recognition and Gesture Display modules, and finally the complete bidirectional loop. All five integration test cases (Table 2) passed, with the complete gesture-voice-gesture pipeline achieving 94% end-to-end accuracy, alongside a minor latency of 0.2–0.3 seconds during gesture-to-voice conversion. Noisy audio inputs were correctly flagged as "Unknown," validating the robustness of the system's error-handling logic.

Table 4. Sample Integration Testing Results

Test Case ID	Modules Integrated	Input	Expected Output	Actual Output	Status
IT-01	Gesture Recognition + Voice Conversion	Gesture "Hello"	Voice output "Hello"	Voice output "Hello"	Pass
IT-02	Gesture Recognition + Voice Conversion	Gesture "Yes"	Voice output "Yes"	Voice output "Yes"	Pass
IT-03	Speech Recognition + Gesture Display	Voice "No"	Gesture display "No"	Gesture display "No"	Pass
IT-04	Full System	Gesture "Hello" → Voice → Gesture	End-to-end accuracy 95%	94% accuracy	Pass
IT-05	Full System	Noisy audio input	Recognized as "Unknown"	Recognized as "Unknown"	Pass

SYSTEM TESTING

System testing evaluated the fully integrated application against functional, performance, stress, noise-handling, and usability criteria. The system met its response-time target, averaging 0.45 seconds per gesture against a 0.5-second threshold, sustained

90% recognition accuracy under background noise, and achieved a 92% usability score among first-time users. All five system-level test cases (Table 3) passed.

Table 5. Sample System Testing Results

Test Case ID	Test Scenario	Input	Expected Output	Actual Output	Status
ST-01	Gesture to Voice	Wave your hand, "Hello"	Voice output "Hello"	Voice output "Hello"	Pass
ST-02	Voice to Gesture	Say "Yes"	Gesture display "Yes"	Gesture display "Yes"	Pass
ST-03	Performance	10 gestures in succession	Response time < 0.5 sec/gesture	Avg 0.45 sec	Pass
ST-04	Noise Handling	Voice input with background noise	Recognize as a correct gesture or "Unknown"	90% accurate	Pass
ST-05	Usability	First-time users perform gestures	Ease of recognition > 90%	92%	Pass

DISCUSSION

The discussion of the results from the proposed Smart Multilingual Bidirectional Communication System highlights several key observations and implications based on the model training, accuracy, loss trends, and testing outcomes.

1. **Model Performance and Accuracy** Both gesture-to-voice and voice-to-gesture pipelines demonstrated steady improvements in classification accuracy over training epochs. By the 10th epoch, the gesture-to-voice model achieved 94% accuracy, while the voice-to-gesture model reached 91%. This difference of approximately 2–4 percentage points suggests that gesture recognition is somewhat more straightforward and reliable than speech recognition within this system, likely due to the additional complexities of audio processing, such as noise, speaker accents, and variability in speech patterns.
2. **Convergence Stability** The loss curves for the voice-to-gesture model exhibited smooth convergence over the 20 epochs, confirming stable learning with no significant overfitting or underfitting. This stability underscores the effectiveness of the chosen deep learning architectures (such as CNN, LSTM, or hybrid CNN-LSTM) coupled with MediaPipe's robust hand landmark extraction in modeling both spatial and temporal features critical to accurate classification.
3. **Real-Time Processing Capability** The system's layered architecture and efficient feature extraction methods allow for real-time responsiveness with minimal latency between input and output. Unit testing across various environmental conditions revealed high accuracy (around 95%) for gesture recognition in quiet settings and highlighted challenges related to lighting variations, which were mitigated through parameter tuning of the MediaPipe detection thresholds. This suggests practical usability in everyday environments, though further optimization may be beneficial in highly variable lighting or noisy acoustic contexts.
4. **Bidirectional Communication Effectiveness** Integrating both gesture-to-voice and voice-to-gesture pathways creates a truly inclusive communication framework that effectively bridges interaction gaps between speech-impaired, hearing-impaired, and normal-speaking individuals. The natural-language output synthesized via TTS and animated gesture renderings enables fluid conversational exchanges. The multilingual processing extension further amplifies the system's applicability across diverse language users, enabling the same gesture semantics to be retained while adapting spoken or textual components to different languages.
5. **Social and Assistive Impact** The system's reliance on standard hardware like webcams and microphones, combined with the open-source and platform-independent design, ensures accessibility and affordability. This fosters the potential for large-scale adoption of assistive communication tools, educational tools for sign language learning, and ambient

intelligence applications, thereby contributing meaningfully to social inclusion and quality-of-life improvements for individuals with communicative impairments.

In summary, the achieved accuracy benchmarks and real-time functional integration validate the feasibility and effectiveness of the proposed bidirectional communication system. The study demonstrates a significant step towards scalable, user-friendly, and inclusive assistive technologies that leverage advanced computer vision and deep learning techniques to overcome persistent barriers in human communication.

5. CONCLUSION

The conclusion of the paper emphasizes that integrating computer vision, deep learning, and speech processing technologies enables the development of an intelligent, socially impactful communication system. By addressing the communication barriers faced by speech- and hearing-impaired individuals, the system facilitates natural, bidirectional interaction between gesture-based and voice-based users without relying on specialized hardware or human interpreters. The modular, scalable, and affordable design, with real-time performance enabled by MediaPipe and deep learning models, advances inclusive communication technologies, improves accessibility, and significantly enhances the quality of life for underserved populations. This work establishes a foundational framework for future research that incorporates advanced AI methods and multimodal learning to advance human-computer interaction and assistive communication.

ACKNOWLEDGEMENTS

The authors thank the Management and Principal of AMC Engineering College, Bengaluru, for providing the facilities and support needed to carry out this work. The authors are also grateful to the Head of the Department and the faculty of the Department of Electronics and Communication Engineering for their guidance and encouragement throughout the work.

AUTHOR CONTRIBUTIONS

Kinny Garg: Conceptualization, Methodology, Software, Investigation, Writing – Original Draft.

Senbagavalli: Supervision, Validation, Writing – Review & Editing.

Lakshmi S: Formal Analysis, Data Curation, Writing – Review & Editing.

M. Sivarathinabala: Resources, Visualization, Writing – Review & Editing.

Shauriya Khanum: Software, Investigation, Validation.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ETHICS-STATEMENT

This study did not involve human participants beyond the authors or any animal subjects. All gesture and speech data used for training and evaluation were either generated by the authors or taken from publicly available datasets, in line with their terms of use. Ethical approval was therefore not required.

REFERENCES

- [1] Hsu, C.-C., Lin, H.-H., 2025. Dynamic Hand Gesture Recognition Using MediaPipe and Transformer. In: *Proceedings of the 2025 IEEE 5th International Conference on Electronic Communications, Internet of Things and Big Data (ECIoTBD)*, pp. 355–362. <https://doi.org/10.3390/engproc2025108022>.
- [2] Khatak, M.F.A., Naaz, S., 2024. Real-Time Multi-Mode Hand Gesture Recognition Using MediaPipe and Deep Learning. *Journal of Computational Analysis and Applications*.
- [3] Srivalli, D., et al., 2025. Real-Time Detection and Conversion of Gestures to Text and Speech to Sign System. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*. <https://doi.org/10.22214/ijraset.2025.67381>.
- [4] Lavanya, P., et al., 2025. Real-Time Sign Language Recognition and Multilingual Speech Output Based on Machine Learning. *International Journal of Human Computations & Intelligence*.

- [5] Utomo, R., et al., 2025. Indonesian Language Sign Detection using MediaPipe with LSTM Algorithm. *Journal of Informatics and Web Engineering*.
- [6] Takouchouang, T.F.S., Vinh, H.T., 2025. Reconnaissance Automatique des Langues des Signes: Hybrid CNN-LSTM based on Mediapipe. *arXiv preprint arXiv:2510.22011*.
- [7] Fang, B., Co, J., Zhang, M., 2018. DeepASL: Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation. *arXiv preprint arXiv:1802.07584*.
- [8] Sanchez, A., et al., 2024. Benchmarking Speech-Driven Gesture Generation Models for Generalization. In: *Proceedings of the International Conference on Multimodal Interaction (ICMI)*.
- [9] Rupesh Kumar, R., et al., 2023. MediaPipe and CNNs for Real-Time ASL Gesture Recognition. *arXiv preprint arXiv:2305.05296*.
- [10] Rastgoo, R., Kiani, K., Escalera, S., 2021. Sign Language Recognition: A Deep Survey. *Expert Systems with Applications* 164, 113794. <https://doi.org/10.1016/j.eswa.2020.113794>.
- [11] Q. Li and C.-C. Hsieh, "Dynamic Hand-Gesture Recognition Using MediaPipe and Transformer," *IEEE-associated Engineering Proceedings*, MDPI, 2025.
- [12] S. Khatak and S. Naaz, "Real-Time Multi-Mode Hand Gesture Recognition Using MediaPipe and Deep Learning," *Journal of Computational Analysis and Applications*, 2024.
- [13] K. Srivalli et al., "Real-Time Detection and Conversion of Gestures to Text and Speech to Sign System," *IJRASET*, 2025.
- [14] P. Lavanya et al., "Real-Time Sign Language Recognition and Multilingual Speech Output Based on Machine Learning," *International Journal of Human Computations & Intelligence*, 2025.
- [15] I. Utomo et al., "Indonesian Language Sign Detection using MediaPipe with LSTM Algorithm," *Journal of Informatics and Web Engineering*, 2025.
- [16] T. Takouchouang and H. T. Vinh, "Reconnaissance Automatique des Langues des Signes: Hybrid CNN-LSTM Based on MediaPipe," *arXiv preprint*, 2025.
- [17] L. Fang et al., "DeepASL: Ubiquitous and Non-Intrusive Word- and Sentence-Level Sign Language Translation," *arXiv preprint*, 2018.
- [18] M. Sanchez et al., "Benchmarking Speech-Driven Gesture Generation Models for Generalization," *ICMI 2024 (OpenReview)*, 2024.
- [19] R. Kumar et al., "MediaPipe and CNNs for Real-Time ASL Gesture Recognition," *arXiv preprint*, 2023.
- [20] S. Rastgoo et al., "Sign Language Recognition: A Deep Survey," *ScienceDirect*, 2021.
- [21] Google, "MediaPipe Hands: On-device Real-Time Hand Tracking," 2020. [Online]. Available: <https://google.github.io/mediapipe/solutions/hands.html>
- [22] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *Proc. IEEE Intl. Conf. Aoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645–6649.
- [23] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [24] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *Proc. Intl. Conf. Machine Learning (ICML)*, 2014.
- [25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint*, arXiv:1412.3555, 2014.
- [26] A. van den Oord et al., "WaveNet: A Generative Model for Raw Audio," *arXiv preprint*, arXiv:1609.03499, 2016.
- [27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. 3rd Intl. Conf. Learning Representations (ICLR)*, 2015.
- [28] Y. Han, J. Chen, and S. Lao, "A Transformer-based Framework for Sign Language Recognition and Translation," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 24–34, 2020.
- [29] M. Johnson et al., "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *arXiv preprint*, arXiv:1609.08144, 2016.
- [30] Y. Gao, Z. Liu, and S. Wang, "Multilingual Speech Recognition and Translation with Transformer Models," in *Proc. IEEE Intl. Conf. Aoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 5737–5741.