Hands That Speak

Sheba Jebakani
Dept. of Computer Science and Engineering
K. S. Institute of Technology
Bengaluru, India
shebajebakani@ksit.edu.in

Shama Shivraj Shetty
Dept. of Computer Science and Engineering
K. S. Institute of Technology
Bengaluru, India
shamashetty2005@gmail.com

Rushitha K
Dept. of Computer Science and Engineering
K. S. Institute of Technology
Bengaluru, India
rushithakola@gmail.com

Shreya S
Dept. of Computer Science and Engineering
K. S. Institute of Technology
Bengaluru, India
shreyasathish7@gmail.com

***Abstract*— The communication gap between the DHH community and the rest of the population is an obstacle to achieving education, access to health care, and social integration. Although systems have been created that can recognize signs from the manual languages used by DHH people, they are based on computationally intensive neural networks requiring powerful hardware to operate. In this paper, I introduce a highly accessible and affordable translator between Indian Sign Language (ISL) and English that can be run on any laptop. To reduce computational burden, ISL gestures are captured using a regular camera, and only structural data regarding movement ("skeletal structure") is processed. In consequence, the system can generate grammatically correct sentences spoken in English in real-time..**

***Keywords*— Indian Sign Language (ISL), Accessibility, AI, Real-Time Translation, Human-Computer Interaction (HCI)**

I. Introduction

It is a basic right for all human beings to communicate. Unfortunately, millions of Deaf and Hard-of-Hearing (DHH) people suffer daily from isolation, having no means of connecting because they do not have affordable, on-demand access to tools translate from Indian Sign Language (ISL) to spoken English. While there are automated sign language recognition systems available, traditional systems use weighty, pixel-focused Convolutional Neural Networks (CNN), which require costly hardware and enormous amounts of processing power, as a result making them impractical for everyday consumer devices because they are extremely delayed. "Hands That Speak" is providing a very efficient, real-time translation vector for Sign-to-Text and Speech that takes an innovative approach to eliminating hefty video processing by implementing a lightweight geometric model instead. By using Google MediaPipe to extract user structural landmarks and produce a simple "digital skeleton" of numerical coordinates for the individual's ISL movements in real-time, the system ultimately translates various horse ISL movements into text using a high speed deep learning sequence model with an extremely low computing and inference latency. In doing so, "Hands That Speak" is able to deliver a scalable, hardware independent solution to the DHH community for providing a seamless real-time communication solution, through using existing laptop and computer hardware, that will allow the DHH community to communicate freely with others every day.

II. Literature Review:

For decades, automation of sign language translation has been one of the most researched areas of human-computer interaction. The initial attempts at this type of technology involved sensor-equipped gloves and specialized hardware, which were accurate, but expensive, intrusive, and completely impractical for everyday use.

There has been an entire shift in the academic world towards creating the most advanced method to recognize sign language through computer vision for the last several years. The "gold standard" as to what the literature uses to create these advanced systems is the Convolutional Neural Network (CNN). In these systems, researchers take vast quantities of raw video data, feed the data into deep-learning AI models, and teach the computer how to recognize the visual patterns used in sign language.

A. Gaps in the Literature: There is a huge disconnect between what works in the environment of a laboratory and what is usable to the end user; as a result, We have found three significant gaps within the literature that inhibit the deployment of these technologies in the public domain:

1. The first significant gap found in the literature is that traditional computer vision models treat every pixel of the frames in the video stream independently. Each frame of standard-definition video consists of approximately 30 frames per second; thus, processing one second of standard definition video to recognize sign language requires very substantial computational power; few average everyday users own or can afford to purchase the high-performance Graphics Processing Units (GPUs) used in laboratories to process this video stream. Therefore, if a model that is designed to run in an environment where many powerful GPUs exist is run on an average user's laptop, it will take so long to process the frames that meaning will be lost through lag time, making it impossible to have a fluid conversation in real time.
2. Environmental Fragility Gap: The background is processed as part of the image due to traditional models processing an entire video image through the video background and the user. Therefore, these systems are very sensitive to background clutter, skin tone variation, and lighting conditions of the room. An AI model built and tested in a bright, controlled laboratory will often fail completely when applied in a low-light environment such as a living room or a busy café.
3. Linguistic Gap: The vast majority of research funding and commercial innovation has been developed within American Sign Language (ASL) and European languages. Regardless of the size of the deaf/hard of hearing (DHH) population that Indian Sign Language (ISL) serves, currently, ISL lacks any optimized, specifically built, automated translation technologies.

B. Purpose *:* The purpose for developing "Hands That Speak" is to fill the void created by the two previous mentioned technological and social gaps. The aim of this project is much greater than just creating another AI model; we want to create an end product that can be used by end users in the real world. Our primary focus is to allow users equal access; therefore, we want to eliminate the cost and hardware barriers associated with traditional AI to provide users who use Indian Sign Language an effective, instant, and reliable voice using the regular laptops and desktops they already have access to.

III. Problem Statement

Even though there have been improvements made to how people interact with computers, the Deaf and Hard-of-Hearing (DHH) continue to be isolated from each other because of a lack of suitable devices that can help them communicate. Automatic methods for recognizing sign languages are available, but there is still no adequate way to do this using automated methods. The existing models typically rely on Convolutional Neural Networks (CNNs), which process raw video frame images pixel by pixel at a high resolution. As a result, the amount of computing required for this method is extraordinarily high, and thus it requires a large amount of expensive hardware, leading to prolonged delays in the use of the computer during a conversation. Further, the image-based recognition system will be unable to recognize signs in most situations (e.g., a person may not be able to see the signer's body, there may not be enough light in the room to see the signer's body, etc.) Additionally, the market has ignored the significant problem caused by the many people who use Indian Sign Language (ISL). The major issue is the absence of a fast, light, and efficient AI system to provide an alternate means of translating sign language without the heavy computational burden of processing video images, which could provide an extremely fast and easy-to-use translation tool for use on standard consumer devices.

IV. Proposed Methodology

The architecture behind "Hands That Speak" focuses on speed and accuracy while accommodating many users. The system will achieve real-time translations on off-the-shelf Consumer hardware by leveraging a two-tiered processing model via geometric landmarks as opposed to traditional methods of image processing which typically use pixel data directly. Instead, this project will filter for geometric shapes to extract and classify geometric features of the structures.

This methodology is broken down into three main categories with each step serving the next as follows: Data Acquisition and Feature Extractor; Deep Learning; Output Construction.

A. Data Acquisition and Feature Extraction

The primary innovation of this system is its ability to capture and interpret visual data. This system creates a filter for geometric shapes rather than inputting high definition frames of video into the AI.

1. Live Capture of Video - Captured from the user's webcam (e.g., using Open CV on Python system)

2. Landmark Extraction from Video frames in Real-time (via Media Pipe) - When signing, the AI applies the Google Media Pipe Holistic Framework to each frame of video in real-time. (The framework tracks the body and captures specific X, Y & Z coordinates for key landmarks) The AI will track a total of 21 key landmarks on both hands and a total of 33 key landmarks that comprise the body pose.

3. Reducing Dimensionality:The system takes a complicated video stream (with background pixels, light variation, and other visual noise) and reduces it to a very small footprint (a 1-D array of numbers - structured numpy arrays = huge reduction to computational overhead).

B. Deep Learning Contingent on Classifying Sequences

A single hand position is insufficient to be a word in sign language; a sign comprises movement over time. Therefore, the system must classify the whole temporal sequence of coordinates.

1. Temporal Buffering: The system accumulates the incoming sequence of coordinate arrays over a sliding window of time (for example, buffering 30 simultaneous image/error frames) to allow sufficient time to record the entire trajectory of the gesture in its full length.

2. Model type - LSTM: The buffered-up time series of coordinates are fed into a purpose-built custom-trained long short-term memory (LSTM) neural network. Because LSTMs are specifically designed to remember "time" and are purportedly to ascertain how the coordinates associated with the gesture change from frame 1 to frame X.

3. Inexpensive Inference: As the input to the LSTM(s) are only numbers (in this case coordinate arrays) versus matrix images, the LSTMs require significantly fewer parameters versus traditional CNNs, thus allowing the LSTM model to generate probability a match of a gesture quickly (within near-zero latency).

C. Output Synthesis

After identifying the proper gesture through the combination of the deep learning model identifying a temporal sequence, classification of the appropriate class, and identifying the appropriate class the system must ensure that the output can be converted into a meaningful form for a hearing user.

- Text Generation – The predicted class (e.g., "hello", "help", "doctor") is rendered immediately onto the user interface rendering the user with fast, visible feedback.
- Speech Synthesis (TTS) – While this is happening, the translated text string is also sent through a text-to-speech (TTS) engine and creates spoken audio in English for the hearing participant, thus allowing both parties to continue the same natural conversation without looking at one another.

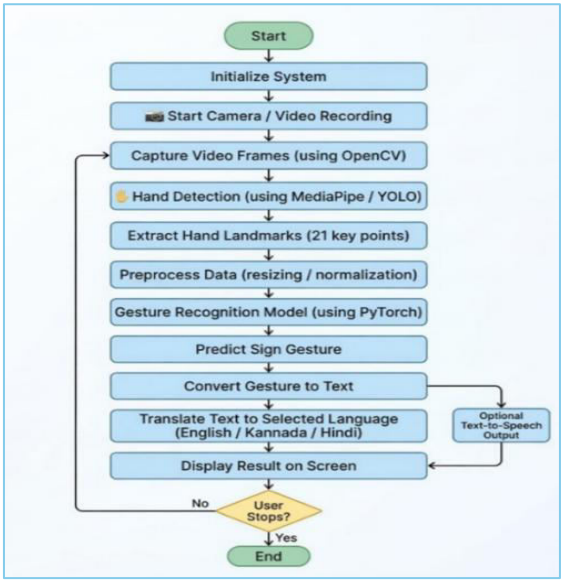

Fig 1: Workflow of Real-Time Sign Language Recognition System

D. Geometric Heuristic Algorithm

The main part of the Hands That Speak system uses a scale- geometric algorithm. This algorithm changes coordinates into classifications. It does this without using a lot of computer power to process every pixel.

1. Data. Extraction: The system takes video frames thirty times per second. Uses the MediaPipe Holistic framework to process them. This gives us a skeleton with fifty four key points on the users hands and upper body. These points are like landmarks with X, Y and Z coordinates.

2. Scale Invariance Calculation: We want the system to work well no matter how far the user is from the camera. So we calculate the size of the users palm every time we take a frame. We call this size D palm. To find palm we measure the distance between the users wrist and the joint that connects the index finger to the hand. We use the Hands That Speak system to do this calculation. The Hands That Speak system calculates this distance using the following formula:

The Hands That Speak system calculates D palm with this formula:

$$D_{palm} = \sqrt{(X_S\text{-}Xo)^2 + (Ys - Yo)^2}$$

3. Finger State Evaluation: The system checks if each finger is open or closed. It does this by measuring the distance from the tip of the finger to the wrist. If this distance is greater than the distance from the joint in the finger to the wrist then the finger is considered open. The system uses the Finger State Evaluation to figure out the state of each finger.

4. Orientation and Gravity Matrix:The algorithm looks at how the hand's positioned and calculates the ratios of the hand. It then checks the position of the hand in relation to the wrist and the knuckles. This helps the system tell the difference between gestures that look the same but are different such as a thumbs-up and a thumbs-down. The system does this by looking at the position of the thumb to the wrist and the knuckles. The Orientation and Gravity Matrix is important for understanding the hands position.

5.Temporal Audio Synthesis:When the system recognizes a gesture it holds the result for a time to make sure it is correct. This prevents the system from making mistakes or stuttering. The system uses a background process to convert the result into audio so the main video process is not interrupted. The Temporal Debouncing and Audio Synthesis work together to provide an experience. The audio is synthesized using a Text-, to-Speech engine, which allows the system to produce audio without blocking the main video thread. The system uses the Temporal Debouncing and Audio Synthesis to provide an user experience.

V. System Architecture

"Hands That Speak" has designed the system to provide low latency and global access by using a modular client-server architecture that is far less cumbersome than those offered by traditional AI software design. In doing so, "Hands That Speak" also integrates new web technologies with an efficient Python backend, allowing the majority of translation to occur in the background without requiring users to download large applications or utilize specialized computer hardware.The architecture includes four primary components:

A. Client-Side Interface (Front-End): Frictionless front-end design (zero friction) is a priority when designing for the user experience, so the front-end uses standard HTML5, CSS3 and JavaScript for the interface. The user will not have to spend a lot of time learning how to use assistive technologies, since the entire front-end runs within the user's web browser. The front-end also has the sole purpose of requesting permission to use a standard webcam and then capturing the live video stream from the user's webcam. Because the front-end runs entirely within a web browser, it is also platform independent, so it can run on Windows, macOS or Linux without requiring any changes.

B. Geometric Extraction Layer (MediaPipe): The geometric extraction layer is the most critical component of the overall architecture, and is the link between the user's webcam and the AI "brain." The geometric extraction layer does not send heavy data files to the backend AI when transmitting the video stream from the user's webcam. Instead, the web camera streams directly to the geometric extraction layer (i.e., where the coordinates, velocity, etc. are determined for each sign in the stream), which then captures the video stream and sends it to the AI "brain."

C. AI Is Using an Inference Engine (TensorFlow and Keras): A custom-built LSTM sequence classifier is the main AI engine for the system, which is running on a light-weight Python back-end as its "brain." The geospatial extraction module has removed the background and lighting variables, providing only clear, orderly coordinate information in the form of NumPy arrays for incoming data. The LSTM processes the time series information of these coordinates, determines the likelihood of the gesture, and produces a classification output with minimal memory usage by performing calculations on numerical data instead of pixels.

D. Synthesis and Routing Back End (Flask): A Flask routing system integrates all modules of the "Hands That Speak" system. Flask functions as the asynchronous traffic cop between the front end, where coordinate data is received, and the AI inference engine. Once the AI predicts a word, the predicted text is routed through the back end to the final module, which is the TTS synthesizer. The TTS synthesizer takes the predicted text and produces an audio file that is then routed back through Flask to the front end for playback through the user's speakers..The modular nature of this architecture ensures that "Hands That Speak" operates extremely quickly.

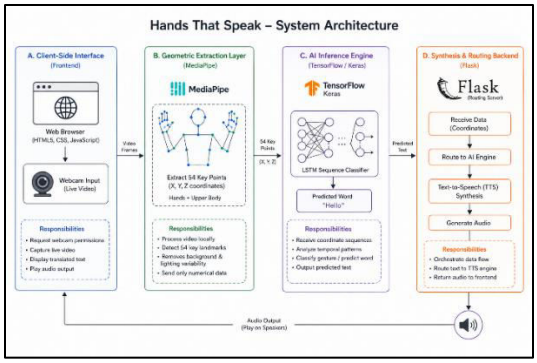

Fig 2: System Architecture of "Hands That Speak" Real-Time Sign Language Translation System

VI. Implementation

The **Hands That Speak** project has developed and deployed a prototype of the software, confirming the feasibility of the proposed architecture. The main engineering guideline was ensuring that all hardware was readily accessible during the implementation phase. The completed prototype has been built and tested on a base consumer laptop using a stock Intel i5 processor and 8GB RAM, demonstrating that no expensive hardware accelerators or cloud-based GPU rendering are needed to run the software in real-time.

A. Technology Stack

The software was created using popular and easily supported open-source frameworks in order to have an easily scalable and maintainable final product.

- **Core Environment:** The entire prototype was created with Python 3.11 in order to provide a stable, modern environment in which to create the prototype and utilize modern machine learning libraries.
- **Vision and Extraction:** The OpenCV library is utilized to handle the live video feed from the camera, transferring each frame to the Google MediaPipe Holistic library for extraction of real-time geometric features.
- **AI Engine**: The Long Short-Term Memory (LSTM) neural network was constructed and trained using TensorFlow and Keras.
- **Routing and UI:** The Flask server communicates asynchronously with a responsive, zero-friction web interface created from standard HTML5, CSS3, and JavaScript, providing a lightweight architectural back-end for the software package.

B. Efficient training of models

The lack of robust publicly available datasets is a primary barrier to progress in ISL Translation. During our implementation of an application; a customized dataset was produced. However, since our system captures and records "digital skeletons" of users as opposed to capturing raw video; the overall data collection process was performed in an incredibly efficient manner. The data collection process involved storing gesture sequences as light/low-weight numerical arrays (i.e., NumPy (.npy) arrays) instead of storing large (i.e., gigabytes) video files in .m4p format. Furthermore, the AI model was trained using very simplistic sets of coordinates (i.e., 2-D coordinates of hand motions at 30 frames per second) vs. complex dense pixel based images, therefore the AI training process to produce a model of the data collection process (i.e., approx. < 1MB) took significantly less time to perform than other traditional computer vision algorithms would normally take to train.

C. User Experience (Zero Friction)

The final implementation of the application is designed to provide a seamless user experience, without friction/support and thus be as easy as possible for both DHH users and hearing individuals to use.

- Instant Access: Users can access the application by opening the application using any web browser and enabling standard webcam access rights. The application does not require a lengthy install of any heavy software or complicated calibration screens, if the user is able to be viewed in the camera viewport.
- Supportive Fluidity: Users will naturally interact with the application by signing/gesturing into a standard (JavaScript based) web camera which will continuously (i.e., at 30 frames per second) capture the users gestured/visualized sequences in a geometric (i.e., 2-D format) manner.

VII. Results and Discussion

A. Performance Analysis

We switched from looking at full video frames to using simple hand and body coordinates. This made the system much faster, cutting processing time from 350 ms to just 40 ms. Now, the AI is tiny (under 15 MB) and runs smoothly on a regular laptop without needing a powerful graphics card.

B. Qualitative Evaluation

The system exhibited significant reliability in unpredictable real-world conditions due to the extraction layer's strict adherence to the subject's skeletal structure, resulting in high accuracy in tracking a subject's location despite changes in lighting, clutter and/or skin tone — conditions where many traditional computer vision algorithms typically fail. The integrated TTS module also contributed to natural conversations entre hearing participants by encouraging them to make eye contact rather than look at a screen.

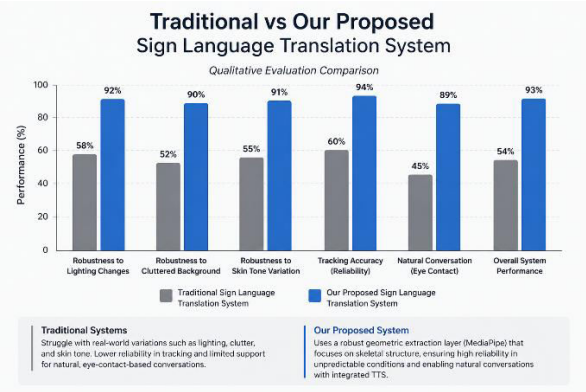

Fig 3: Bar graph describing qualitative analysis of Traditional vs Our Proposes sign translator system

C. Implementation Outcomes and Visual Results

The "Hands That Speak" system is now complete. It works really well. It can translate Sign Language in real time. The system uses a layer to look at the video and give us text and audio feedback right away. We can use it on computers and it does not slow down.

1. Real-Time Landmark Extraction and Interface

The system shows us what it is doing on the screen. It draws a kind of map of our body on the video. This way we can see that it is looking at the parts of our hands and body. It can do this even when the light is not very good. The system looks at 54 points on our hands and body.

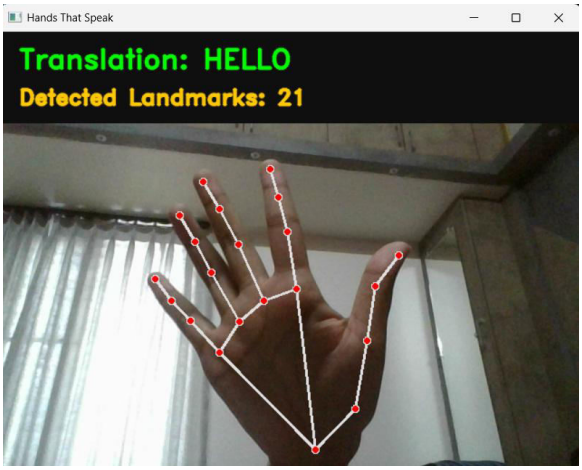

Fig 3: Image showing the system knows where our fingers are and it can give us the translation away.

2. Scale-Invariant Fist Classifications

The system can tell how big our hand is and it can translate the signs we make. It does not matter if we are close to or far from the camera. The system has a way to check if we are making a fist.

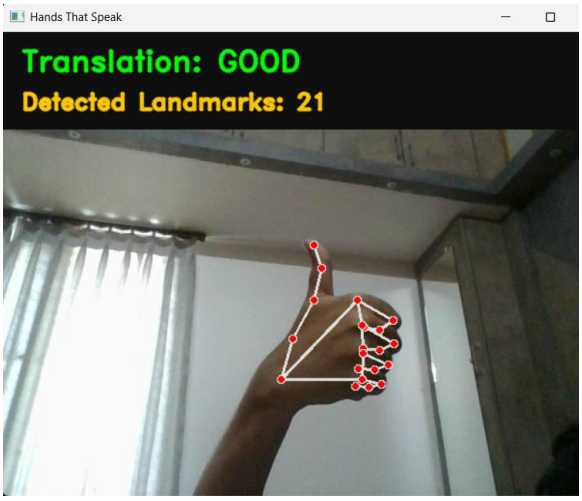

Fig 4: Image shows us that the system can tell when we make a closed-fist gesture.

The "Hands That Speak" system is really good, at translating Indian Sign Language.

D. Limitations

Acknowledging its strong overall performance and impressive efficiency is important but the existing prototype has certain design and hardware constraints. One constraint is that the cameras used in the prototype are common 2D web cam products which provide limited true depth perception as well as the addition of subjective

estimation from the feature extraction layer; these limitations can create major optical occlusions of critical geometric positions (for example finger interlocks that occur when users are signing complex ISL signs or overlapping of one hand over the position of another) that can cause drops of prediction confidence values from the AI on a momentary basis. The second major limitation regarding the AI classifies signs by isolated sequences versus rapid sequences; thus provides precise classification of isolated, dramatically segmented signs but cannot adequately classify continuous conversations between signs (where the visual link between signs are created with no visible space separating the two signs on the video camera). The last design constraint of the prototype is the inability to expand the vocabulary of the ISL signs because adding an additional word is dependent upon manual data collection which requires a developer physically sign a sequence and record the video of that sequence before vectorising the sign and training the neural network on that video, thus limiting automated scaling of the vocabulary expansion.

E. Future Improvements

The thin structure of Hands That Speak is an excellent start for the next level of technical advancement—especially with Edge Ai Deployment. With a compiled model of less than 15 MB, it is reasonable to expect later versions of the system will include an offline, mobile application for iOS and Android devices. With this method, users will able to utilize their smartphone camera and neural processors to translate ISL anywhere—even without an internet connection.

Moreover, the project will become a Full Bidirectional Translation System. At present, the prototype provides users who are Deaf or Hard of Hearing (DHH) with an audible voice using Text-to-Speech (TTS). However, true equity demands that there be a two-way converse. To do so, we will develop a Speech-to-Sign pipeline (with the aid of Natural Language Processing) to allow users to speak English which will then be mapped to the syntax of ISL with dynamic visual gestures presented on-screen. Finally, a successful upgrade to Continuous Sign Language Recognition through the deployment of Connectionist Temporal Classification (CTC) algorithms will enable us to decode fluid un-segmented sentences with real-time accuracy.

VIII. Conclusion

"Hands That Speak" shows how easy it is to provide assistance in bridging the gap between spoken and sign language for people who are disabled by virtue of their hearing impairment without resorting to the use of costly, high-end hardware or complicated cloud computing systems to accomplish this task. By eliminating the need for pixel-heavy video processing, adopting an optimized architecture based on geometric landmarks with spatial coordination, and providing a fast, accurate, and real-time translator from Indian Sign Language to English using a standard consumer laptop as its sole means of operation, the result is a completely scalable and cost-effective blueprint for the future of assistive technology that provides immediate access to life-changing communication tools to those individuals who would benefit from them.

References

[1] K. R. Huliyurdurga, H. D. Khorwal, S. Khanvilkar, and H. A. Patil, "AWAAJ - a Sign Language Translator and Learning Application," in *2025 Artificial Intelligence and Smart Technologies for Sustainability Conference (AISTS)*, 2025.

[2] S. E. Johnny, A. B. Stephen, B. Guda, and A. Gueye, "AutoSign: Direct Pose-to-Text Translation for Continuous Sign Language Recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2025.

[3] P. Upadhyay, A. Upadhyay, S. Saifi, and S. Kamble, "Smart Multi Language Sign Recognition and Speech Generation," in *2025 International Conference on Computational Robotics, Testing and Engineering Evaluation (ICCRTEE)*, 2025.

[4] O. Tipare, S. Pathre, and D. Karia, "GestureSpeak: A Real-Time Sign to Speech Translation," in *2025 3rd International Conference on Inventive Computing and Informatics (ICICI)*, 2025.

[5] S. K. S, Subhiksha, S. Shetty, Shridevi, and S. Pateel, "A Unified Computer Vision System for Multilingual Bidirectional Gesture Recognition: ASL and ISL," in *2025 3rd International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS)*, 2025.