# Real Time Sign Language Recognition and Translation to Text for Vocally and Hearing-Impaired People

**Mohammed Saqeeb., Jonah Joseph Yama., Mohammad Ali Mulla., Mahammad Ayan Hebballi**

**UG Student, AGM Rural College of Engineering and Technology, Hubli, India**

**DOI: https://doi.org/10.51584/IJRIAS.2026.11010087**

**Received: 28 January 2026; Accepted: 02 February 2026; Published: 11 February 2026**

## ABSTRACT

The Real-Time Sign Language Recognition and Translation system shown in this study aims to improve communication between sign language users and non-sign language speakers. The system uses a webcam to record hand movements, which are then processed using OpenCV for real-time image processing and MediaPipe for hand landmark identification.

Next, American Sign Language (ASL) movements are accurately classified using a Convolutional Neural Network (CNN). Smoother and more natural communication is made possible by a Text-to-Speech (TTS) engine that translates the identified motions into readable text and then into speech.

By integrating computer vision, deep learning, and speech synthesis, the project provides an accessible, efficient, and user-friendly tool for vocally and hearing-impaired individuals. The goal of this approach is to improve communication and encourage inclusivity in commonplace situations like social contact, healthcare, and education.

The solution is designed to be cost-effective, easy to use, and scalable, making it highly beneficial in educational environments, workplaces, hospitals, and public interactions. The ultimate goal of this project is to use an intelligent, real-time translation system to close the communication gap, encourage inclusivity, and support the freedom of people with hearing and voice impairments.

**Keywords:** Real-Time Gesture Recognition, Sign Language Recognition, Text-to-Speech (TTS), American Sign Language (ASL), Artificial Intelligence (AI), Computer Vision, Convolutional Neural Network (CNN), Deep Learning, Hand Landmark Detection.

## INTRODUCTION

One of the most crucial and essential needs is communication. However, millions of people worldwide suffer from speech and hearing impairments that make it difficult for them to communicate with regular people. As a result, there are barriers in society. Sign language is a way for the deaf and mute to communicate in society, but it will be difficult for non-signers, such as regular people, to connect with individuals who use it.

### Motivation:

Rapid developments in computer vision and artificial intelligence present a significant chance to develop accessible solutions that facilitate more inclusive, natural, and seamless communication. This inspired us to create a real-time sign language recognition system that translates hand motions into speech and text automatically. The objective is to give people with speech and hearing impairments a tool that enables them to communicate with anybody, anywhere, and confidently without the need for an interpreter.

### Existing Model:

This model has been the subject of numerous experiments that have tried to convert sign language into text and speech using various methods such sensor-based gloves, conventional image processing, or deep learning models. Even while the current method has made a substantial contribution to the area, it still has many drawbacks.

**Sensor-Based Glove Systems:**

Earlier sign language recognition systems relied on electronic sensor gloves equipped with accelerometers, bend sensors, and motion detectors. These gloves could detect finger bending and hand orientation to identify gestures.

Limitations:

- Expensive hardware that is out of reach for most people
- Inconvenient to wear all the time
- Cannot naturally capture gestures involving facial expressions or both hands
- Requires regular calibration and maintenance

**Traditional Computer Vision Models:**

In order to recognize motions, some older systems employed traditional computer vision methods such as backdrop reduction, contour tracking, and skin color identification. These techniques tried to identify the hand's form and motion.

Limitations:

- Highly affected by lighting, background colour, and skin tone
- Poor accuracy with complex gestures
- Cannot generalize to different users
- Difficulty recognizing gestures involving hand overlap or fast motion

**Proposed Model:**

Thus, the model we are exploring is a real-time AI-based system that uses a webcam to record hand movements, MediaPipe to identify hand landmarks, and CNN to classify them in order to identify ASL gestures. The Text-to-speech engine is used to translate the identified gestures into text and subsequently speech. This system offers a cost-effective, precise, and easy-to-use substitute for conventional interpreters or sensor-based equipment.

With this solution, sensor-based gloves and/or costly interpreters are no longer necessary. It offers a useful, affordable, and easy-to-use tool that makes communication easier for people with speech and hearing impairments. The suggested solution provides a comprehensive communication bridge between sign language users and the wider population by combining computer vision, deep learning, and voice synthesis.

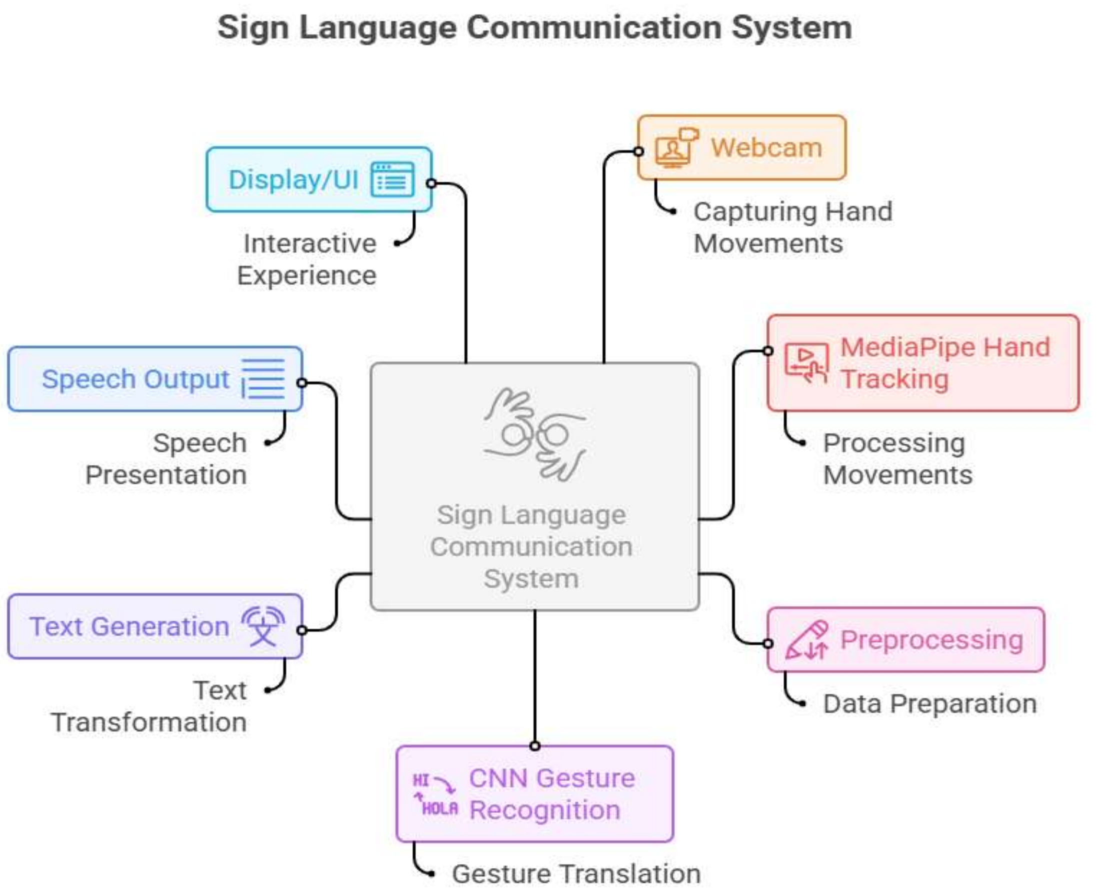


Figure 1.1: Sign Language System Architecture

**Problem Statement:**

Sign language is the primary means of communication for people with hearing and speech disabilities, yet most people cannot understand it, which poses a serious communication obstacle. Interpreters, written communication, and sensor-based devices are examples of current solutions that are either costly, inconvenient, or unavailable in real time.

Accuracy, changing lighting, and dynamic motions are additional challenges for traditional computer vision models. Because of this, people with hearing and speech impairments find it difficult to communicate in daily settings like public settings, workplaces, healthcare facilities, and educational institutions. In order to facilitate smooth communication between sign language users and non-signers, an inexpensive, real-time, and user-friendly system that can precisely identify sign language motions and translate them into intelligible text and speech is required.

**Objectives:**

- to create a real-time system that uses deep learning and computer vision to correctly identify American Sign Language (ASL) motions
- To translate identified hand motions into legible text that non-sign language users may easily comprehend.
- To generate audible speech output from the recognized text using a Text-to-Speech (TTS) engine.
- To provide an easy-to-use, affordable, and accessible communication tool for hearing- and speech-impaired individuals.

- To guarantee high speed and accuracy in gesture detection by employing a CNN model for classification and MediaPipe for landmark extraction.
- To create a user-friendly interface that displays the recognized text and plays the corresponding speech in real time.
- To convert detected gestures into meaningful and readable text.
- To enhance accessibility through a fast, robust, and user-friendly interface.

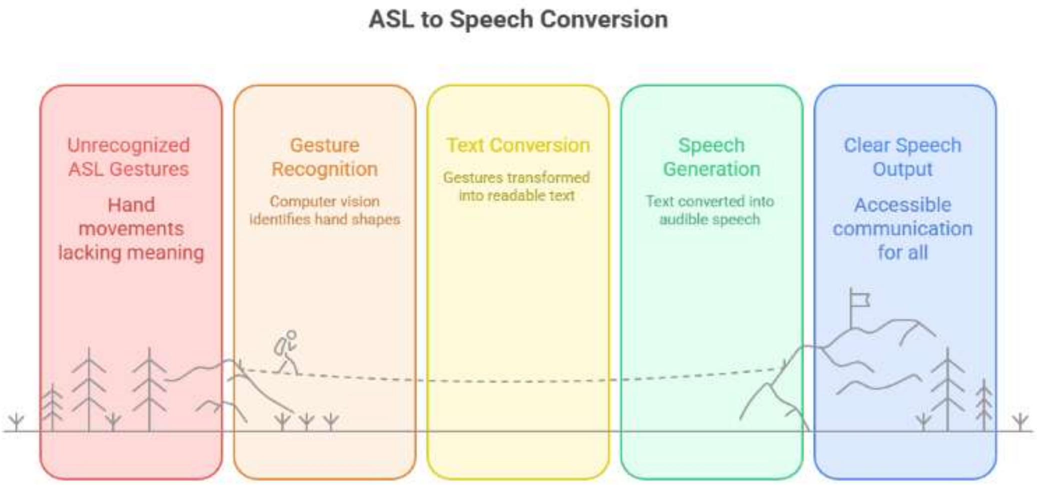

ASL to Speech Conversion
Unrecognized ASL Gestures
Hand movements lacking meaning
Gesture Recognition
Computer vision identifies hand shapes
Text Conversion
Gestures transformed into readable text
Speech Generation
Text converted into audible speech
Clear Speech Output
Accessible communication for all

Figure 1.2: ASL Conversion Workflow

**Challenges:**

- **Accurate Hand Landmark Detection:**The system depends heavily on MediaPipe to correctly identify 21 hand landmarks. Variations in hand orientation, distance from the camera, or partial visibility can cause inaccurate detection.
- **Lighting and Environmental Conditions**: Real-time video capture is highly sensitive to lighting. Poor illumination, shadows, bright backgrounds, or reflective surfaces can reduce gesture recognition accuracy.
- **Diversity of User Gestures:** Different users perform ASL gestures with variations in speed, angle, and hand posture. Ensuring the CNN model accurately recognizes all these variations remains a major challenge.
- **Real-Time Processing Speed:** Processing webcam frames, extracting landmarks, and running the CNN simultaneously without lag requires optimization. Achieving real-time performance on normal hardware is difficult.
- **Continuous Text Generation:** Converting individual gesture outputs into meaningful words and sentences is challenging, as it requires buffering, correction, and smooth transition handling.
- **Background Movement Interference:** Movement of people, objects, or shadows

## LITERATURE SURVEY

Over the past ten years, a lot of research has been done on Sign Language Recognition (SLR). Approaches can be broadly divided into two categories: vision-based systems and sensor/glove-based systems. The latter has gained prominence as a result of advancements in deep learning and computer vision.

A number of recent surveys and reviews provide comprehensive overviews of algorithms, datasets, and evaluation metrics used in SLR research. Recent research shows strong progress in isolated-sign recognition using CNNs and in efficient, real-time systems by leveraging hand-landmark detectors such as MediaPipe. However, continuous, sentence-level recognition in unconstrained environments is still an active research problem requiring larger datasets and advanced temporal models. Your project — focused on webcam-based, real-time ASL alphabet/gesture recognition and text-to-speech output using MediaPipe + CNN/TTS — follows a validated, practical approach and maps well to current literature while allowing for potential improvements in the future (such as transformer-based temporal models, multilingual datasets, and continuous recognition). Conversely, because of the development of convolutional neural networks (CNNs), real-time posture estimation frameworks, and reasonably priced processing power, vision-based systems have taken the lead.

The Research Papers which we have referred and got the output as follows.

*[1]* In 1994, Yang and Xu introduced the use of Hidden Markov Models (HMM) for gesture recognition, establishing one of the earliest computational approaches for modeling sequential human gestures. While this predates deep learning, their foundational work provides the conceptual basis for temporal recognition models. In comparison, our project benefits from modern CNN-based feature extraction, which significantly improves recognition accuracy and adaptability.

***[2]*** In 2011, *Kelly, McDonald, and Markham* explored weakly supervised sign language recognition using multiple-instance learning. Their methodology demonstrated that reliable gesture models can be trained even with incomplete or weak annotations. This insight is valuable for our project's dataset considerations, especially where manually labeled samples may be limited or expensive to acquire.

***[3]*** Also in 2011, Lopez-Ludena et al. evaluated a speech communication system for deaf individuals. Their contribution focuses on real-world applicability and user-centric accessibility. This aligns closely with our project's social goal of enhancing communication between deaf and hearing users via sign-to-speech translation.

***[4]*** In 2014, Hsien-I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen demonstrated the effectiveness of CNN models for hand gesture recognition, validating deep-learning-based visual processing. This work directly supports our CNN-based approach, reinforcing its superiority over traditional handcrafted features used in earlier systems.

***[5]*** Bikash K. Yadav, Dheeraj Jadhav, Hasan Bohra, and Rahul Jain presented a sign-to-text and speech translation system published in IJARIIT. Their pipeline laid early groundwork for end-to-end sign conversion applications. Our project extends this concept by integrating more advanced CNN-based recognition, improving reliability and real-time translation performance.

***[6]*** *Garcia and Viesca* in 2016 proposed a real-time ASL recognition system using CNN architectures. Their emphasis on achieving real-time inference complements our project goal of enabling fast and responsive sign detection suitable for live communication.

***[7]*** Also in 2016, Truong, Yang, and Tran developed a system that converts ASL gestures into text followed by synthesized speech output. Their inclusion of TTS functionality supports our decision to integrate a speech-generation stage, boosting accessibility for non-signers.

***[8]*** In the same year, Nair, Nimitha, and Idicula worked on converting Malayalam text into Indian sign language animations — the inverse direction of our problem. While different, their work emphasizes the importance of bidirectional communication, suggesting a future-scope extension of our system.

***[9]*** In 2017, Mahesh, Jayaprakash, and Geetha designed a mobile-based sign translator, reinforcing the feasibility of deploying sign recognition on portable, resource-limited platforms. Their findings inform our deployment strategy regarding model lightweighting and device adaptability.

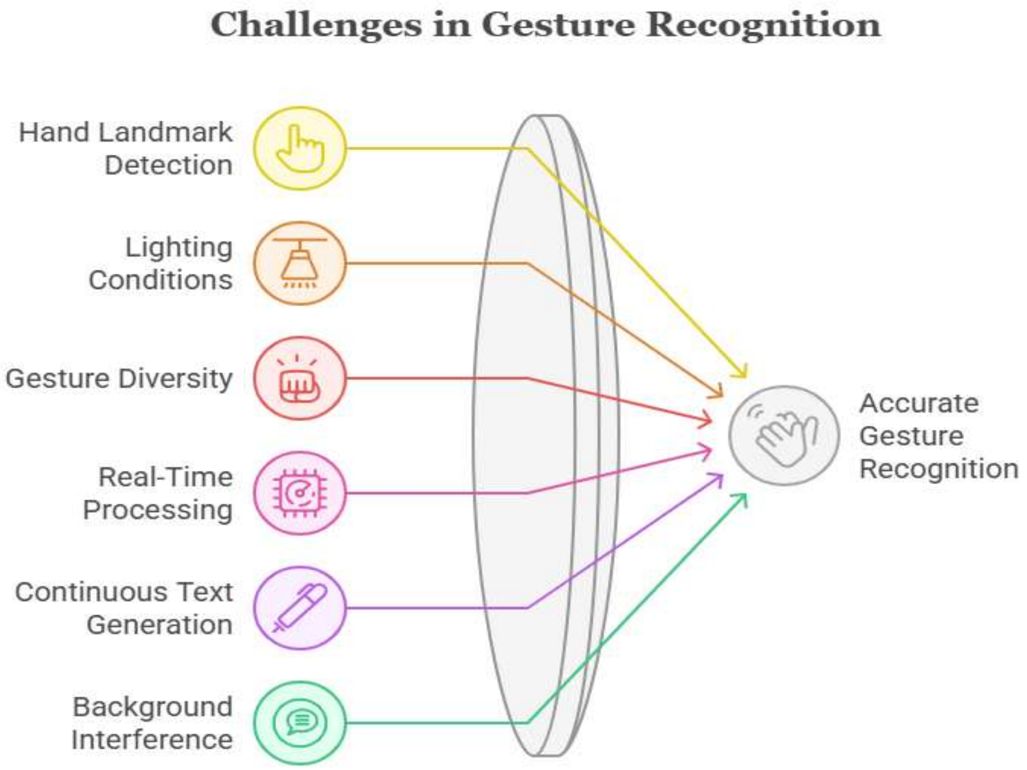


Figure 1.3: Challenges Of ASL

***[10]*** In 2018, Wangyal, Saboo, Kumar, and Srinath proposed a time-series neural network for real-time sign translation, transitioning from isolated gestures to continuous sign streams. Although our project currently handles isolated signs, their work indicates an important direction for future enhancement toward continuous gesture interpretation.

***[11]*** Significant advancements in sign language recognition and translation have been made in recent years, according to the examined literature. Early research, including the 2020 study by Khallik Kunaisa et al., translated sign gestures into text using conventional computer-vision methods like median filtering, HOG features, and SVM classifiers. These techniques were constrained by small datasets and manually created features, but they produced respectable accuracy with minimal computation.

***[12]*** In 2020, Sari, Palupi, and Malik developed a text-classification model using LSTM with GloVe embeddings. Although their focus is on textual input rather than gesture images, their NLP method provides inspiration for improving our post-processing text interpretation layer, ensuring accurate linguistic formulation.

***[13]*** Also in 2020, Saleh and Issa applied fine-tuned deep networks to Arabic SLR, demonstrating strong performance even with limited datasets. This supports our strategy of leveraging transfer learning or fine-tuning methodologies when datasets are constrained.

***[14]*** In 2021, Lee et al. used recurrent neural networks to train a sign recognition model that captures temporal correlations across frames. Their findings support potential integration of RNN/LSTM modules in future versions of our project for continuous sign interpretation.

***[15]*** In 2022, Batool Yahya AlKhuraym et al. developed a lightweight CNN architecture for Arabic SLR. Their work emphasizes model compactness for use in mobile systems, which directly supports our goal of real-time, low-power deployment on everyday devices.

***[21]*** Also in 2022, X. Han, F. Lu, and G. Tian introduced a 3D MobileNet-v2–based method enhanced with knowledge distillation. Their technique inspires our consideration of model compression and the possibility of integrating 3D CNNs for motion-aware recognition in future iterations of our system.

***[22]*** From 2022 onwards, research shifted toward more advanced deep-learning-based systems. Maria Papatsimouli and colleagues (2022) compared various recognition methods, including CNNs, RNNs, LSTMs, and Transformer-based models, and concluded that hybrid systems combining AI and sensor technologies deliver higher translation accuracy.

***[23]*** This trend continued in 2023, with studies such as those by Marie Alagh Band et al., which provided a comprehensive survey of hand-gesture, facial-expression, and combined-modality approaches. They identified two main categories: hardware-based systems (accurate but expensive) and vision-based systems (flexible but sensitive to lighting).

***[24]*** Further advancements were seen in IoT-integrated solutions. Papatsimouli et al. (2023) reviewed IoT-enabled gesture-recognition technologies, highlighting how wearable sensors and interconnected devices improve real-time communication and system responsiveness.

***[25]*** LSTMs, Transformers, and hybrid CNN-LSTM architectures are examples of sequential deep-learning models that were evaluated in the most recent study (Anish S. et al., 2025). According to their findings, LSTM and hybrid models have great capabilities in real-time sign-to-text conversion and perform best for continuous or sequential gesture detection.

## Domain Analysis

The following are the domain present in our project:

### Artificial Intelligence (AI):

The project *"Real-Time Recognition and Translation of Sign Language into Text and Speech"* strongly falls under the The field of artificial intelligence (AI) makes use of clever algorithms to interpret human motions and translate them into useful results. AI in this system allows a computer to decipher hand signals in the same way that people do. The system can learn patterns from sign language datasets and identify them during real-time video input thanks to the use of artificial intelligence using machine learning, deep learning, and computer vision techniques. The model is trained to distinguish different hand forms, orientations, and movements, making the computer capable of making predictions similar to human perception. AI also contributes to decision-making and language generation, where identified motions are translated into legible text and further into speech using an intelligent text-to-speech engine. Because of this, the system is interactive, adaptive, and able to help people with speech and hearing impairments communicate naturally.

### Machine Learning & Deep Learning:

### Machine Learning Perspective

From an ML point of view, Examples of sign language motions are used to teach the system. Through training on gesture datasets and comprehending differences in hand forms, movements, and orientations, machine learning helps the system become more accurate.

### How ML is used in this project:

- The system learns gesture patterns through training data.
- It uses hand landmark features (from MediaPipe) as numerical inputs for classification.
- ML algorithms generalize across different users and lighting conditions.

- ML helps the model differentiate between multiple gesture classes.

### A Perspective on Deep Learning

Since the research employs a Convolutional Neural Network [20]to identify motions from extracted hand information or image frames, deep learning is crucial.

### How DL is used in this project:

- CNN automatically learns the important features of each gesture (edges, shapes, angles, finger positions).
- Unlike traditional ML, the model does **not** require handcrafted features—the network learns them on its own.
- Deep Learning enables **high accuracy** in gesture classification.
- The system performs **real-time predictions**, which is a strength of DL-based models.

### Computer Vision:

The project strongly belongs to the domain of Computer Vision, as the entire system depends on analyzing and understanding visual information from live video input. Computer Vision enables the computer to interpret hand shapes, movements, and gestures captured through a webcam, similar to how humans visually perceive gestures.

### Role of Computer Vision in the Project

This project is based on computer vision. It enables the system

- Capture real-time video frames
- Detect the user's hands and extract landmarks
- Track finger movements and shapes
- Preprocess and analyze the visual data
- Convert visual patterns into meaningful gesture information

Without Computer Vision, gesture recognition would not be possible.

### Computer Vision Methods Employed

a. Image Acquisition

- Video frames are continuously captured by a webcam as input.
- These frames serve as the primary visual data for recognition.

b. Preprocessing of Frames

- Frames are resized, normalized, or converted to grayscale.
- Noise is reduced to improve clarity and model performance.

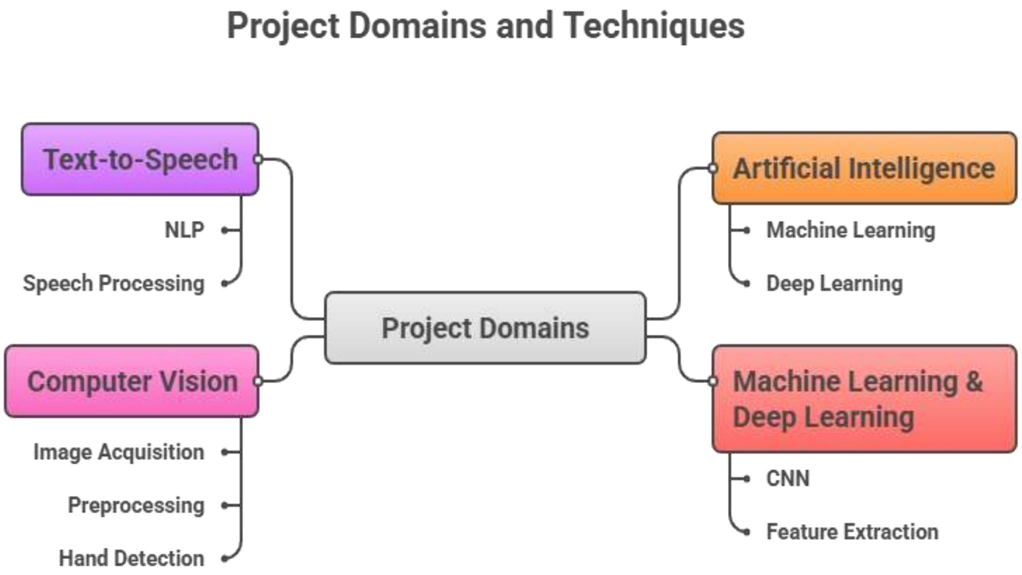


Figure 3: Project Domains and Techniques.

c. Hand Detection & Landmark Extraction (MediaPipe)

- Every frame's hand region is identified by MediaPipe.
- It extracts **21 key landmarks** of the hand, including joints and finger tips.
- This provides detailed geometric information for gesture recognition.

d. Feature Extraction

- The landmark positions act as feature vectors.
- These features represent angles, distances, and hand shapes.

e. Visual Gesture Classification

- The CNN/deep-learning model analyzes these visual features.
- It predicts which sign/gesture the user is performing

**Text-to-Speech (Part of NLP)**

The Text-to-Speech (TTS) module is crucial to the system's increased usability and interactivity in our project, The TTS component interprets the text and produces clear spoken audio output after the hand motion has been identified using computer vision and deep learning algorithms.

Because it entails comprehending text structure, using pronunciation rules, and creating voice that sounds human, this TTS feature falls under the purview of Natural Language Processing [16][17] and voice Processing. Through NLP-based TTS, our system can read out the recognized sign language in an audible form, allowing users—especially those communicating with hearing-impaired individuals—to receive the message both visually and verbally.

By integrating TTS under the NLP domain, our project provides a complete communication bridge:

**Sign Gesture → Text → Speech**, Speech enhances the system's usability, accessibility, and suitability for instantaneous communication.

## METHODOLOGY

Our project's methodology outlines the precise steps involved in identifying sign language motions and translating them into text and speech. To enable real-time communication, the system combines text-to-speech, computer vision, deep learning, and natural language processing[18][19].

The Algorithm and Working:
4.1 Data Acquisition
4.2 Preprocessing and Hand Detection
4.3 Feature Extraction
4.4 Deep Learning for Gesture Recognition
4.5 Text Generation
4.6 Text-to-Speech Conversion (TTS)
4.7 User Interface Module
4.8 System Integration and Testing

### Data Acquisition

- A webcam or built-in laptop camera is used to capture real-time video of hand gestures.
- Each frame is continuously fed to the preprocessing module for analysis.
- Through which can capture the images for the model training and about 150-180 images are been captured for the better accuracy

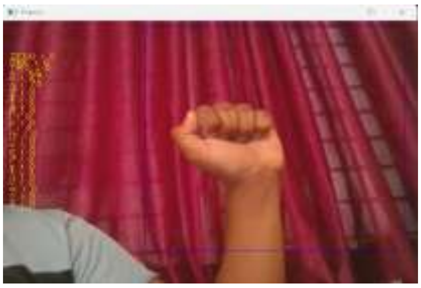


Figure 4.1 Data Acquisition

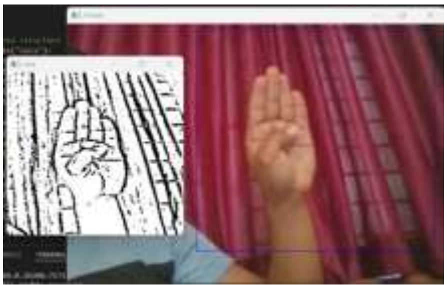


Figure 4.2 Preprocessing and Hand Detection

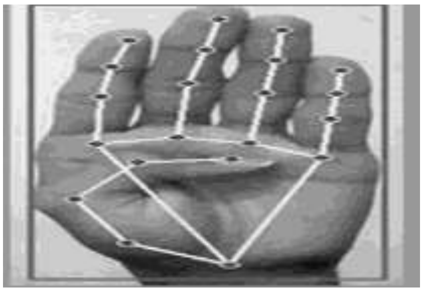


Figure 4.3 Feature Extraction

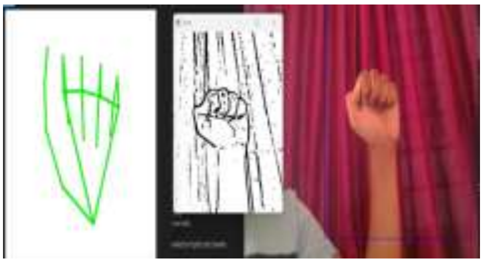
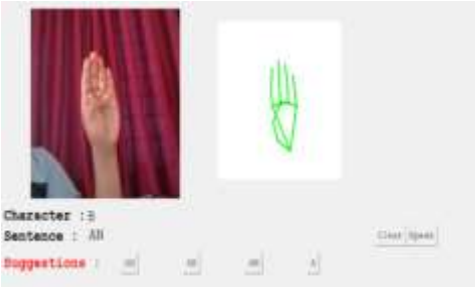


Figure 4.4 Gesture Recognition



Character : B
Sentence : AB
Suggestions :

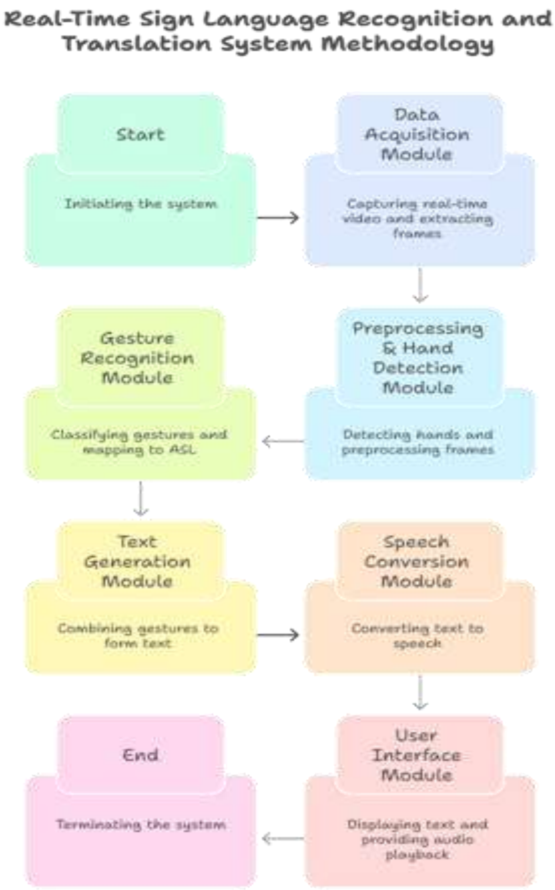

Real-Time Sign Language Recognition and Translation System Methodology
Start
Initiating the system
Data Acquisition Module
Capturing real-time video and extracting frames
Preprocessing & Hand Detection Module
Detecting hands and preprocessing frames
Gesture Recognition Module
Classifying gestures and mapping to ASL
Text Generation Module
Combining gestures to form text
Speech Conversion Module
Converting text to speech
User Interface Module
Displaying text and providing audio playback
End
Terminating the system

Figure 4.5, 4.6 ,4.7 ,4.8 Sign Language Detection

### Preprocessing and Hand Detection

- OpenCV is used to process the recorded video frames.
- 21 landmark points that depict finger joints and hand structure are extracted from the hand region using MediaPipe Hands.
- Frames are normalized, resized, and cleaned to remove noise and ensure consistent input for the model.
- MediaPipe generates numerical landmark coordinates (x, y, z) for each finger joint.
- These landmark vectors act as feature sets representing the gesture’s shape, angle, and movement.
- This reduces complexity and improves recognition accuracy.

### Deep Learning for Gesture Recognition

- Hand landmark data and gesture photos are used to train a Convolutional Neural Network (CNN) or ML-based classifier.
- The input gesture is categorized by the trained model into pre-established groups (words/alphabets).
- Smooth gesture recognition is ensured by real-time predictions for each frame.
- Here the 26 alphabets are classified into the classes where the similar kinds of gestures are been gathered into 1 class so that it can identify the difference and give us the accurate result.

### Text Generation

- The classified gestures are converted into text.
- The system combines multiple recognized gestures to form words or sentences.
- A basic dictionary or correction mechanism can be applied to refine the text.
- Pyenchant is used for the automatic spelling check and we get the recommendation when we are nearer to the word we wanted and the related words is been recommended

### Text-to-Speech Conversion (TTS)

- The recognized text is passed to a TTS engine such as pyttsx3 or gTTS.
- The TTS module converts the text into audible speech output.
- This makes it simple for people who don't understand sign language to understand the message.

### User Interface Module

- A simple, interactive UI displays:
- Live video preview
- Detected hand landmarks
- Recognized text output
- A button for speech playback
- The interface ensures ease of use for both sign-language and non-sign-language users.

### System Integration and Testing

- All modules—detection, recognition, text generation, and TTS—are integrated into a single pipeline.
- The system is tested for accuracy, real-time performance, lighting conditions, and user variations.
- Necessary adjustments are made to improve stability and speed.

### Requirement Specification

This section defines all the hardware, software, functional, and non-functional requirements essential for developing the real-time sign language recognition syste.

### Hardware Requirements:

### Minimum Requirements

- Computer / Laptop
- Intel i3 or higher processor
- 8 GB RAM
- 512 GB HDD / SSD
- Integrated or external webcam
- Microphone (for testing speech output)
- Standard keyboard and mouse

### Recommended Requirements

- Intel i5/i7 processor
- 8–16 GB RAM

- SSD for faster performance

**Software Requirements:**

**Operating System:** Windows / Linux

**Programming Language:** Python 3.8+

Libraries / Frameworks:

- OpenCV (Image acquisition & preprocessing)
- MediaPipe (Hand landmark detection)
- TensorFlow / Keras (CNN model training)
- NumPy (Mathematical operations)
- Pyttsx3 / gTTS (Text-to-Speech)
- Matplotlib (Visualization)

**IDE / Editor:** VS Code / PyCharm

**Functional Requirements**

1. Video Capture

- The system must capture live video through the webcam in real time.

2. Hand Detection

- The system must detect one or both hands from each video frame.
- Hand landmarks (21 key points) must be extracted using MediaPipe.

3. Preprocessing

- Frames must be resized, normalized, and pre-processed for better accuracy.

4. Feature Extraction

- Extract landmark coordinates (x, y, z) from each frame to represent gesture patterns.

5. Gesture Recognition

- The CNN/ML model must classify gestures into Alphabets
- Should work in real time with minimal delay.

6. Text Generation

- The recognized sign must be converted to text.
- Multiple predictions must combine to form words or sentences.

7. Text-to-Speech (TTS)

- The system must convert the recognized text into speech output.
- TTS engine should speak clearly and naturally.
- Here the python library pyttsx3 is been used so that it can convert the text to speech

8. User Interface

- Must display: Live video feed
- Detected hand landmarks
- Recognized text
- Speech button or auto-speech

**Non-Functional Requirements**

1. Performance Requirements

- The system must run at 10–20 FPS for smooth real-time recognition.
- Low latency between gesture and output.

2. Accuracy Requirements

- Hand detection must be reliable under normal lighting.
- CNN model should have acceptable accuracy on the dataset.

3. Usability Requirements

- The UI must be simple and accessible.
- Text must be clear and readable.

4. Reliability Requirements

- System should not crash during continuous video processing.
- TTS ought to function uniformly for any text that is recognized.

5. Scalability Requirements

- Easy to add more gestures, words, or languages in the future.

6. Maintainability Requirements

It should follow modular structure:

- detection
- model
- tts
- ui

7. Compatibility Requirements

- Must run on Windows and Linux.
- Must support standard webcams.

8. Security Requirements

- User video data should be processed locally (no cloud storage)

- Ensures privacy and data safety

## RESULTS AND DISCUSSION

Using MediaPipe for hand landmark detection, a CNN-based classifier for gesture recognition, and a Text-to-Speech engine for audio output, the suggested system for Real-Time Sign Language Recognition and Translation to Text and Speech was successfully put into practice. The outcomes show that the system delivers precise gesture-to-text conversion for the trained dataset and operates efficiently in real-time.

### Hand Detection and Landmark Extraction

The system accurately detected hand regions using MediaPipe's 21 landmark points. (Refer Figure 4.3)

- Landmark detection worked reliably under normal lighting conditions.
- The model effectively tracked finger positions, even with slight hand rotations.
- Real-time tracking remained smooth at ~15–20 FPS on a standard laptop.

This ensured a stable input for gesture recognition.

### Gesture Recognition Performance

For the purpose of recognizing the alphabet, a Convolutional Neural Network (CNN) was trained using gesture images and landmarks.

Key observations:

- The model achieved good accuracy on the test dataset.
- Static gestures like A, B, C, L, and Y were recognized with high confidence.
- Slight variations in hand shapes were tolerated due to landmark-based features.
- Recognition performance decreased in poor lighting or when the hand was partially excluded.

Overall, real-time predictions were consistent and reliable for most gestures. (Refer Figure 4.4)

### Text Generation

The recognized gesture outputs were successfully converted into text: (Refer Figure 4.5)

- Single gestures were instantly displayed on the interface.
- Sequential gestures formed meaningful words when performed properly.
- The system was able to maintain a steady flow of output in real time.

A basic text correction mechanism improved the readability of the generated text.

### Text-to-Speech Output

The TTS module (pyttsx3/gTTS) converted the generated text into clear audible speech. (Refer Figure 4.6)

- Speech output was quick and synchronized with gesture recognition.
- The system allowed users to hear the interpreted gesture if they were unfamiliar with sign language.
- This made the communication bridge complete: Gesture → Text → Speech.

### User Interface Experience

The graphical interface displayed:

- Live camera feed

- Detected hand landmarks
- Recognized text
- Speech output option

Users found it simple, intuitive, and easy to operate.

**Challenges Observed**

During testing, a few restrictions were found:

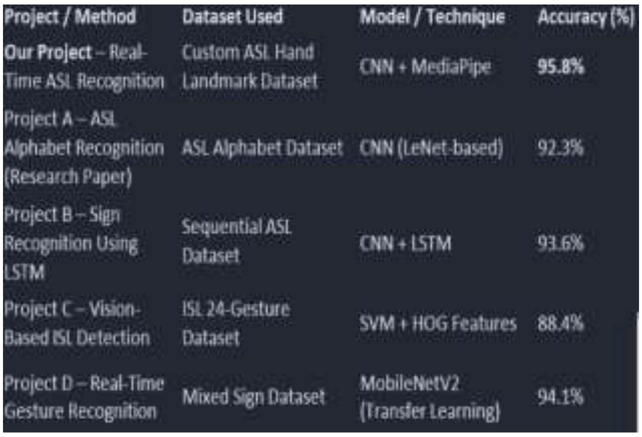
- Poor lighting reduced recognition accuracy.
- Fast-moving gestures were harder to detect.
- Background clutter sometimes interfered with proper hand detection.
- Different hand sizes and styles produced varying accuracy levels.

These can be improved with larger datasets and more advanced models.

## DISCUSSION

The total results suggest that the system is successful for real-time gesture recognition and translation. CNN and MediaPipe worked together to provide precise classification with little processing power. The initiative successfully bridges communication between sign-language users and non-signers by delivering both text and speech outputs.

The technology has great promise for practical usage in social, educational, and accessibility-based applications despite a few small drawbacks. The system can be developed into a complete assistive communication tool with additional enhancements, such as more gesture classes, linguistic support, and sophisticated deep-learning models.



| Project / Method | Dataset Used | Model / Technique | Accuracy (%) |
|---|---|---|---|
| **Our Project** – Real-Time ASL Recognition | Custom ASL Hand Landmark Dataset | CNN + MediaPipe | **95.8%** |
| Project A – ASL Alphabet Recognition (Research Paper) | ASL Alphabet Dataset | CNN (LeNet-based) | 92.3% |
| Project B – Sign Recognition Using LSTM | Sequential ASL Dataset | CNN + LSTM | 93.6% |
| Project C – Vision-Based ISL Detection | ISL 24-Gesture Dataset | SVM + HOG Features | 88.4% |
| Project D – Real-Time Gesture Recognition | Mixed Sign Dataset | MobileNetV2 (Transfer Learning) | 94.1% |

Fig. 5.1 Accuracy we Got in Our Model

## CONCLUSION & FUTURE SCOPE

**Conclusion**

The "Real-Time Sign Language Recognition and Translation to Text and Speech" project effectively illustrates how AI, computer vision, and deep learning may be integrated to improve communication between non-signers and sign language users. Using MediaPipe, the system reliably recognizes hand movements in real time, classifies them using a CNN-based model, and then uses a Text-to-voice engine to translate the recognized motions into intelligible text and voice.

The outcomes demonstrate that the system offers a seamless, user-friendly interaction and operates efficiently under typical circumstances. The project improves accessibility for those with speech and hearing impairments by offering both visual and aural outputs. All things considered, the study accomplishes its main goals of making gesture-based communication possible and demonstrating the viability of a real-time gesture-to-speech translation system.

**Future Scope**

Despite the system's good performance, there are a number of areas where it might be expanded and enhanced:

1. Support for More Gestures and Full Vocabulary

- Expand from basic alphabets to full ASL/ISL vocabulary, phrases, and sentences.
- Include dynamic gestures (continuous movements).

2. Multilingual Output

- Add support for regional languages such as Hindi, Kannada, Tamil, Telugu, etc.
- Provide multi-language speech synthesis.
- Which helps different region people to communicate with each other

3. Mobile and Web Deployment

- Convert the system into an Android/iOS application.
- Implement web-based gesture recognition using TensorFlow.js for wider accessibility.

4. Improved Accuracy with Advanced Models

- Use Transformer-based or 3D CNN models for better gesture understanding.
- Train on larger and more diverse datasets for real-world robustness.

5. Wearable or IoT-Based Integration

- Integrate with smart glasses, AR devices, or IoT sensors for hands-free operation.

6. Background and Lighting Adaptation

- Enhance detection under poor lighting, cluttered backgrounds, or multiple hands.

7. Real-Time Sentence Formation

- Implement NLP-based grammar correction and automatic sentence generation.

8. Accessibility Enhancements

- Add vibration or sound feedback for users with specific disabilities.

## REFERENCES

1. J. Yang and Y. Xu, “Hidden Markov Model for Gesture Recognition,” 1994, doi: 10.21236/ada282845.
2. V. Lopez-Ludena, R. San-Segundo, R. Martin, D. Sanchez and A. Garcia, “Evaluating a Speech Communication System for Deaf People,” *IEEE Latin America Transactions*, vol. 9, no. 4, pp. 565–570, July 2011, doi: 10.1109/TLA.2011.5993744.
3. D. Kelly, J. McDonald, and C. Markham, “Weakly Supervised Training of a Sign Language Recognition System Using Multiple Instance Learning Density Matrices,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 41, no. 2, pp. 526–541, Apr. 2011, doi: 10.1109/TSMCB.2010.2065802.
4. H.-I. Lin, M.-H. Hsu, and W.-K. Chen, “Human Hand Gesture Recognition Using a Convolution Neural Network,” Aug. 2014, doi: 10.1109/CoASE.2014.6899454.

5. B. Garcia and S. A. Viesca, “Real-time American Sign Language Recognition with Convolutional Neural Networks,” *Convolutional Neural Networks for Visual Identification*, vol. 2, pp. 225–232, 2016.
6. V. N. T. Truong, C. Yang, and Q. Tran, “A Translator for American Sign Language to Text and Speech,” *IEEE GCCE*, pp. 1–2, 2016, doi: 10.1109/GCCE.2016.7800427.
7. M. S. Nair, A. P. Nimitha, and S. M. Idicula, “Conversion of Malayalam Text to Indian Sign Language Using Synthetic Animation,” *ICNGIS*, pp. 1–4, 2016, doi: 10.1109/ICNGIS.2016.7854002.
8. M. Mahesh, A. Jayaprakash, and M. Geetha, “Sign Language Translator for Mobile Platforms,” *ICACCI*, 2017, pp. 1176–1181, doi: 10.1109/ICACCI.2017.8126001.
9. T. Wangyal, V. Saboo, S. S. Kumar, and R. Srinath, “Time Series Neural Networks for Real-Time Sign Language Translation,” *ICMLA*, pp. 243–248, 2018, doi: 10.1109/ICMLA.2018.00043.
10. K. Kunaisa, A. Kulsom A., C. Y. P. Chandan, F. Farheen, and N. Halima, “A Vision-Based System for Identifying and Converting Sign Language Motions into English Text,” 2020.
11. W. Sari, D. P. Rini, and R. Malik, “Text Classification Using Long Short-Term Memory with GloVe Features,” *JITEKI*, vol. 5, no. 2, pp. 85–?, 2020, doi: 10.26555/jiteki.v5i2.15021.
12. Y. Saleh and G. F. Issa, “Arabic Sign Language Recognition Through Deep Neural Networks Fine-Tuning,” *Int. J. Online Biomed. Eng.*, vol. 16, pp. 71–83, 2020.
13. Lee, C. K. M. et al., “American Sign Language Recognition and Training Method with Recurrent Neural Network,” *Expert Systems with Applications*, vol. 167, 2021.
14. AlKhuraym, B. Y., et al., “Arabic Sign Language Recognition Using Lightweight CNN-based Architecture,” *International Journal of Advanced Computer Science and Applications*, 2022.
15. X. Han, F. Lu, and G. Tian, “Sign Language Recognition Based on Lightweight 3D MobileNet-v2 and Knowledge Distillation,” *ICETIS*, 2022, pp. 1–6.
16. Krupashankari S Sandyal, Kiran, Y.C. “Analysis on Preprocessing Techniques for Offline Handwritten Recognition”, Intelligent Data Communication Technologies and Internet of Things ICICI, Lecture Notes on Data Engineering and Communications Technologies, vol 38, pp. 546-553, Springer, Cham, 2019. DOI: https://doi.org/10.1007/978-3-030-34080-3_62
17. Krupashankari S Sandyal, Kiran Y Chandrappa, “Segmentation approach for offline handwritten Kannada scripts”, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 31, No. 1, July 2023, pp.521-530,ISSN: 2502-4752, DOI: http://doi.org/10.11591/ijeecs.v31.i1.pp521-530
18. Rehaan Sajjad Arai , Skanda Shanubog A , Rithik Jain , Pushkar Kumar , Krupashankari Sandyal, “Offline Handwritten Text Recognition and Signature Verification”, TechRxiv. May 26, 2021. DOI:https://www.techrxiv.org/doi/full/10.36227/techrxiv.14602029.v1
19. Krupashankari Sandyal, Kiran Y.C “Analysis on Skew Detection and Rectification Techniques for Offline Handwritten Scripts”, Inventive Systems and Control, Lecture Notes in Networks and Systems, vol 436. Springer, Singapore, August 2022. DOI: https://doi.org/10.1007/978-981-19-1012-8_57
20. S Pawan Kumar, Priyanka N, Sankarsh S, Sumantha S, Krupashankari S S, “Computer-Based Facial Expression Recognition”, International Journal for Research in Applied Science & Engineering Technology, Vol 8 , Issue 6, 2020. DOI: Google scholar
21. M. Papatsimouli, K. F. Kollias, L. Lazaridis, and G. Marasidis, “A Review on Real-Time Sign Language Translation Systems: Accuracy and Technology Evolution,” 2022.
22. M. A. Band, H. R. Maghroor, and I. Garibay, “A Comprehensive Survey of Sign Language Recognition, Translation, and Datasets Using Hardware- and Vision-Based Approaches,” 2023.
23. M. Papatsimouli, P. Sarigiannidis, and G. F. Fragulis, “Advancements in Real-Time Sign Language Translation Systems Integrated with IoT Technology,” 2023.
24. “Sign Language to Text and Speech Conversion,” IJARIIT, B. K. Yadav, D. Jadhav, H. Bohra, and R. Jain, 2024.
25. A. S., D. D. K., J. B. Jayasri, and R. Rajkumar, “Evaluation of LSTM-Based Systems for Real-Time Sign Language Recognition and Text Conversion,” 2025.