Journal of Science, Computing and Engineering Research (JSCER)
Volume-9, Issue- 3, March-April 2026.
DOI: *https://doi.org/10.46379/jscer.2026*

REAL-TIME SIGN LANGUAGE TO SPEECH CONVERSION

[1] Dr. Anusuya , [2]Priyansh Khandelwal, [3]Udit Aggarwal, [4]Sahil Khan, [5]Taniya Jangid

[1]Professor, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India.

[2,3,4,5]UG Student, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

Article Information

Received : 29 March 2026

Revised : 30 March 2026

Accepted : 31 March 2026

Published : 03 April 2026

Corresponding Author:

Priyansh Khandelwal

***Abstract*—** Communication barriers between hearing-impaired individuals and the general population remain a significant social challenge. This project presents an AI-powered Real-Time Sign Language to Text and Speech Conversion system designed to bridge this gap using computer vision and deep learning techniques. The proposed system captures static American Sign Language (ASL) gestures through a standard webcam without relying on external sensor-based hardware. Captured frames are preprocessed through Region of Interest (ROI) extraction, grayscale conversion, Gaussian blurring, adaptive thresholding, resizing, and normalization. A Convolutional Neural Network (CNN) trained on a structured ASL alphabet dataset classifies gestures with high accuracy, achieving 95–98% recognition under controlled conditions.

Keywords:American Sign Language, Convolutional Neural Network, Computer Vision, Natural Language Processing, Text-to-Speech.

Citation: Dr. ANUSUYA, Priyansh Khandelwal, Udit Aggarwal, Sahil Khan, Taniya Jangid, "Real-Time Sign Language to Speech Conversion", Journal of Science, Computing and Engineering Research, 9(4), April 2026.

I INTRODUCTION

Communication is one of the most essential elements of human interaction and social development. Individuals with hearing and speech impairments encounter significant challenges in daily communication, particularly when interacting with people unfamiliar with sign language. This communication gap often leads to misunderstanding, dependency, reduced opportunities, and social isolation. Sign language, such as American Sign Language (ASL), is a structured visual communication system; however, its effectiveness is limited because most people do not understand it.

The system adopts a vision-based approach that eliminates the need for expensive sensor-equipped gloves or motion-tracking hardware. Instead, it utilizes a standard webcam to capture hand gestures and processes them using image preprocessing techniques and a Convolutional Neural Network (CNN). This approach makes the system cost-effective, portable, and scalable for real-world deployment.

A. The Core Mechanism

The system workflow consists of five sequential stages: Image Capture, Preprocessing, Prediction, Text Generation, and Output Conversion. Gestures are performed within a predefined Region of Interest (ROI), preprocessed to enhance clarity, classified using a trained CNN model, validated across multiple frames for stability, and finally converted to text and optional speech. By combining computer vision, deep learning, and natural language processing techniques, the system provides a comprehensive assistive communication solution.

II PROBLEM STATEMENT

Despite significant advancements in artificial intelligence and computer vision, communication barriers between hearing-impaired individuals and non-signing populations remain largely unresolved. While sign language serves as an effective and expressive medium for individuals with speech and hearing impairments, it is not commonly understood by the majority of society. This linguistic disconnect limits participation in education, employment, healthcare, and daily interactions.

A. Existing System Limitations

Existing solutions can broadly be classified into sensor-based systems and vision-based systems. Sensor-based approaches rely on wearable gloves embedded with flex sensors, accelerometers, or motion detectors. Although these systems can achieve high accuracy, they are often expensive, uncomfortable, and impractical for everyday use. Vision-based systems utilize cameras to capture hand gestures but many existing implementations suffer from sensitivity to lighting variations, background interference, high computational

requirements, inconsistent real-time performance, and limited integration of linguistic refinement mechanisms.

B. Research Gaps

There is a clear need for a cost-effective, hardware-independent, real-time sign language recognition system that not only identifies gestures accurately but also ensures prediction stability, meaningful word formation, linguistic correctness, and optional speech output. Furthermore, the system must operate efficiently on standard CPU-based hardware to ensure accessibility and widespread adoption.

III PROPOSED MODEL

The proposed work model presents a real-time, vision-based Sign Language to Text and Speech Conversion system. The architecture follows a structured and sequential processing pipeline integrating computer vision, deep learning, natural language processing, and speech synthesis into a single streamlined pipeline.

A. Image Acquisition and ROI Extraction

The system initializes a live video stream from the webcam. A predefined Region of Interest (ROI) is displayed on the screen to guide users in positioning their hand gestures consistently. Static ASL gestures are performed within this bounded region to ensure a uniform background and optimize detection accuracy. Restricting gesture detection to a predefined rectangular area improves both processing speed and prediction accuracy.

B. Preprocessing Pipeline

Once captured, the ROI undergoes a series of enhancement operations using OpenCV. The preprocessing pipeline consists of the following stages:

- **Grayscale Conversion:** RGB frames are converted to grayscale to reduce computational complexity while preserving essential structural information such as edges and contours.
- **Gaussian Blurring:** Applied to remove high-frequency noise and smooth the image, helping the model focus on the primary hand structure.
- **Adaptive Thresholding:** Dynamically separates the hand region from the background by converting the grayscale image into a binary format, effective under uneven lighting conditions.
- **Resizing and Normalization:** The processed binary image is resized to a standardized 128×128 pixels and pixel values are normalized to the range [0, 1] to ensure consistent model input.

C. CNN-Based Gesture Classification

The core component of the proposed model is a Convolutional Neural Network (CNN) specifically designed for static gesture classification. The architecture consists of:

- Convolutional layers for hierarchical spatial feature extraction (edges, contours, hand configurations).
- MaxPooling layers to reduce spatial dimensions and improve generalization.
- ReLU activation functions to introduce non-linearity for complex pattern learning.
- Fully connected dense layers for final classification.
- Softmax output layer generating probability scores for each of the 27 classes (26 alphabets + blank)

.

The model is trained using categorical cross-entropy loss and optimized with the Adam optimizer over 20–25 epochs with batch normalization to prevent overfitting.

D. Frame-Based Validation and Word Formation

To ensure stability in real-time recognition, a count-based validation mechanism confirms a character only if it is predicted consistently over approximately 50 consecutive frames. Once a character is validated, it is appended to a text buffer. The detection of a blank gesture (absence of a hand) acts as a word delimiter. A Hunspell-based spell correction module then evaluates the generated word against a dictionary and suggests the most probable correction, ensuring linguistically coherent output.

Component Diagram

The Component Diagram illustrates the structural relationship among the software components.

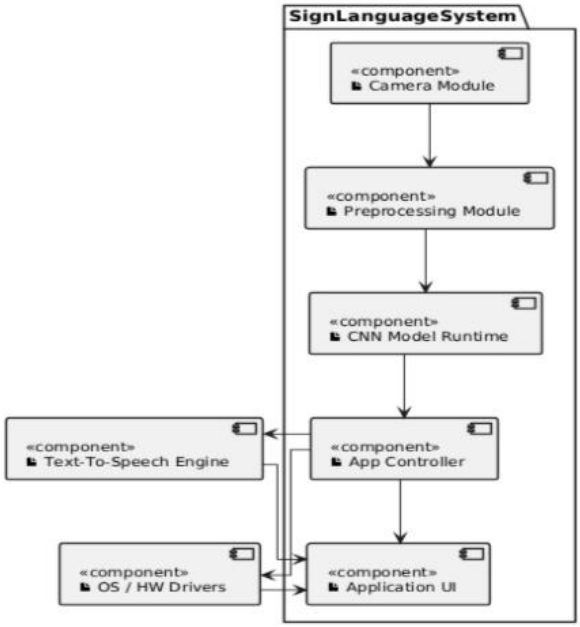

Fig. 1. Component Diagram

IV. TECH STACK

The system is developed using a carefully selected combination of open-source technologies spanning computer vision, deep learning, natural language processing, and speech synthesis.

A. OpenCV

OpenCV serves as the foundational library for real-time image processing. It handles webcam capture, ROI extraction, grayscale conversion, Gaussian blurring, adaptive thresholding, resizing, and normalization. Its optimized image transformation functions ensure clean and consistent input to the neural network.

B. TensorFlow and Keras

TensorFlow provides the underlying machine learning framework for efficient tensor operations, model training, and inference. Keras, as a high-level API on top of TensorFlow, simplifies CNN design through modular layer-based construction. The trained model is serialized and reloaded for real-time inference without retraining.

C. Hunspell

Hunspell is integrated for spell checking and autocorrection. It uses dictionary-based morphological analysis to validate recognized words and suggest corrections, improving overall communication clarity and enhancing user experience.

D. pyttsx3 / Google TTS

These Text-to-Speech libraries convert recognized text into natural-sounding audio. pyttsx3 provides offline synthesis ensuring functionality without internet access, while Google TTS offers more natural voice output when connectivity is available.

E. Python 3.10+

Python serves as the central programming language connecting all components. Its extensive support for AI and data science libraries enables seamless integration between OpenCV, TensorFlow, Hunspell, and TTS modules.

V. ESULT SCREENSHOTS

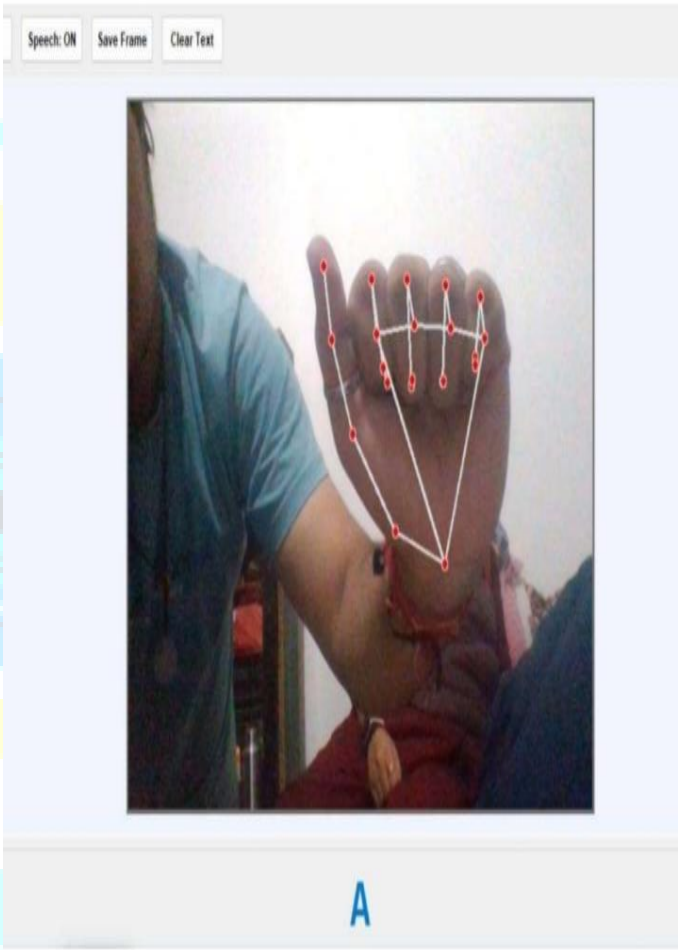

Fig. 2. Output Page

VI. CONCLUSION

The development of the Real-Time Sign Language to Speech Conversion system represents a successful intersection of modern computer vision, deep learning, and natural language processing. The primary objective—to bridge the communication barrier between hearing-impaired individuals and non-signing populations using a cost-effective, hardware-independent solution—has been fundamentally achieved.

The system effectively transforms static ASL hand gestures captured via a standard webcam into meaningful text and speech output without requiring specialized sensor hardware or GPU acceleration. Key achievements include:

- **High Recognition Accuracy:** 95–98% classification accuracy under controlled lighting conditions using a lightweight CNN architecture.
- **Real-Time Performance:** 30–40 FPS processing on standard CPU-based hardware with frame-based validation ensuring stable, reliable output.
- **Linguistic Correctness:** Hunspell-based autocorrection refines recognized text, ensuring grammatically coherent communication.
- **Multimodal Output:** Optional TTS synthesis converts recognized text into audible speech, completing the communication loop.

While the current version focuses on static ASL alphabet gestures, the rapidly evolving landscape of Large Language Models (LLMs), dynamic gesture recognition (LSTM/Transformer architectures), and mobile deployment (TensorFlow Lite) provides several exciting avenues for future expansion. The system demonstrates how artificial intelligence can be applied as a powerful tool for social inclusion—ensuring that every individual, regardless of ability, can communicate, connect, and be understood.

REFERENCES

[1] T. Starner, J. Weaver, and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998.

[2] L. Pigou, S. Dieleman, P. J. Kindermans, and B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," ESANN, 2015.

[3] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015.

[4] F. Chollet, "Keras," GitHub, 2015. Available: https://github.com/fchollet/keras

[5] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[6] O. M. Sincan and H. Y. Keles, "AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods," IEEE Access, 2020.

[7] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.

[8] O. Koller, J. Forster, and H. Ney, "Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems," Computer Vision and Image Understanding, 2015.

[9] Python Software Foundation, "Python 3.10 Documentation," 2026. Available: https://docs.python.org/3.10/

[10] Kaggle, "Sign Language MNIST Dataset," 2026. Available: https://www.kaggle.com/datasets/datamunge/sign-language-mnist

[11] pyttsx3, "Offline Text-to-Speech Library for Python," 2026. Available: https://pyttsx3.readthedocs.io/

[12] Hunspell Project, "Hunspell Spell Checker and Morphological Analyzer," 2026. Available: https://hunspell.github.io/