ISSN: 2349-6002

AI-Based Real-Time Sign Language Translator Using CNN, LSTM, MediaPipe, and Multi-Language TTS

Mrs Veerendeswari J[1], Mrs Celin Julie B[2], Jayasri M[3], Ganiga M[4], Harini S[5]
[1]*Head of the department, Department of Information Technology, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*
[2]*Assistant Professor, Department of Information Technology, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*
[3,4,5]*UG, Department of Information Technology, Rajiv Gandhi College of Engineering and Technology, Puducherry, India*
doi.org/10.64643/IJIRTV12I11-201050-459

***Abstract*—This paper presents an AI-powered Sign Language Translator designed to bridge communication between deaf and mute individuals and the hearing population. The system leverages Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to recognize hand gestures in real-time and convert them into text and speech. Hand landmarks are extracted using Google's MediaPipe framework (21 keypoints per hand). The model is trained on the WLASL (Word-Level American Sign Language) dataset and extended with Indian Sign Language (ISL) data. Multi-language translation is supported via Google Translate and gTTS. The system achieves 27–30 fps on standard CPU hardware with an overall accuracy of 91.6%, offering a scalable and inclusive communication solution for accessibility.**

***Index Terms*—Sign Language Recognition, CNN, LSTM, Media Pipe, Real-Time Gesture Recognition, WLASL Dataset, Multi-Language Translation, Text-to-Speech, Accessibility, Deep Learning, Indian Sign Language.**

I. INTRODUCTION

Communication is a fundamental human right, yet millions of individuals worldwide who rely on sign language face daily barriers in interacting with the hearing population. Sign language is a rich, natural visual-gestural language used by the Deaf community; however, its understanding among the general public remains extremely limited, creating significant challenges in healthcare, education, public services, and daily social interactions. With the rapid advancement of Artificial Intelligence (AI), computer vision, and deep learning, it has become possible to develop automated systems that interpret sign language gestures and translate them into spoken or written form, dramatically improving accessibility and social inclusion for individuals with hearing or speech impairments. This project presents a comprehensive AI-based Sign Language Translation System that operates in real-time using a standard webcam. The system processes live video, detects hand landmarks using MediaPipe, classifies gestures using a CNN+LSTM model trained on WLASL, and outputs recognized signs as text and speech, with Google Translate support for multi-language output.

Key contributions: (1) MediaPipe integration for robust real-time keypoint extraction, (2) dual-stream CNN+LSTM model combining spatial and temporal learning, (3) multi-language text and audio output via gTTS and Google Translate, and (4) user-friendly Tkinter GUI.

II. PROBLEM STATEMENT

Despite growing AI adoption in communication systems, the technology gap between sign language users and the hearing world remains wide. Existing solutions often require specialized hardware (depth sensors, gloves), suffer from poor real-time performance, or support only a small vocabulary. Critical challenges identified include:

- Absence of real-time, high-accuracy systems on commodity hardware.
- Poor generalization across different signers, backgrounds, andlighting.

ISSN: 2349-6002

- Limited vocabulary most systems handle fewer than 100gesture classes.
- Lack of multi-language output, restricting usability.
- No TTS integration for audio accessibility.

III. LITERATURE REVIEW

A. Traditional and Vision-Based Approaches

Early sign language recognition systems relied on instrumented gloves measuring finger angles and wrist orientation. While accurate, these were expensive and impractical for everyday use. Color-based hand segmentation using HSV thresholds was also explored but failed under varying skin tones and illumination conditions [1].

B. Deep Learning-Based Methods

CNNs brought a paradigm shift in image-based gesture recognition. He et al. [2] demonstrated that deep residual networks significantly outperform shallow networks on visual recognition. Hybrid CNN+LSTM architectures [4] where CNNs extract per-frame spatial features and LSTMs model inter-frame dependencies have shown strong benchmark results and form the backbone of the proposed system.

C. MediaPipe and Skeleton-Based Recognition

Google's MediaPipe Hands [5] provides lightweight, real-time detection of 21 3D hand landmarks from a single monocular RGB camera. Skeleton-based representations offer invariance to appearance and reduced dimensionality. Graph Convolutional Networks on skeleton graphs [6] show competitive performance with far lower computational requirements than RGB-based models.

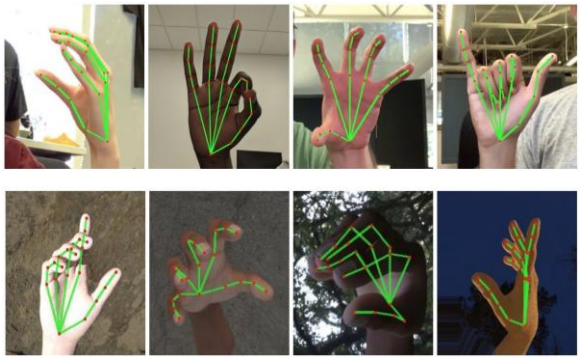

Fig. 1. Hand landmark keypoint detection across diverse skin tones and backgrounds using MediaPipe Hands framework.

D. Datasets and Multi-Stream Architectures

The WLASL dataset [7] is the largest publicly available ASL dataset with 2,000+ gloss classes performed by 119 signers. ISL datasets from Kaggle and Roboflow support ISL alphabet/word recognition. State-of-the-art multi-stream architectures [8] process RGB frames, optical flow, depth data, and skeleton keypoints in parallel, but their complexity makes deployment on standard devices challenging.

IV. SYSTEM ARCHITECTURE

The proposed system follows a modular pipeline architecture comprising six functional layers: input acquisition, hand landmark detection, feature extraction, deep learning inference, post-processing (multi-language translation), and output rendering.

A. Input Module

Live video is captured at 30 fps using OpenCV (cv2.VideoCapture). Each frame is horizontally flipped for mirror-like interaction and converted BGR→RGB. A resolution of 640×480 pixels is maintained to balance quality and processing speed.

B. MediaPipe Hand Landmark Detection

MediaPipe Hands returns 21 normalized 3D landmark coordinates (x, y, z) per detected hand, corresponding to: wrist (1), MCP joints (5), PIP joints (4), DIP joints (4), fingertips (5). Keypoints are normalized to image dimensions for scale and position invariance.

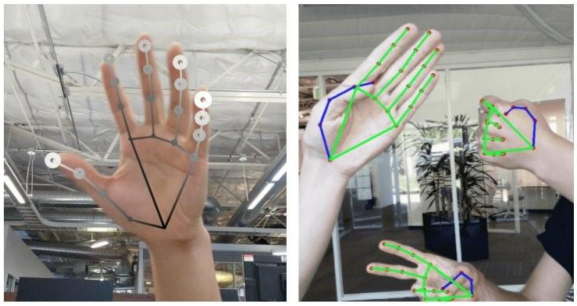

Fig. 2. MediaPipe hand skeleton overlay showing 21 3D keypoint landmarks with single-hand and dual-hand tracking.

C. Feature Extraction Module

21 landmarks × 3 coordinates = 63D feature vector per frame. For dynamic gestures, 30 consecutive frames form a 30×63 = 1,890D temporal window capturing full sign motion trajectory. Features are min-max normalized to [0, 1] before model input.

ISSN: 2349-6002

D. Deep Learning Classification

CNN+LSTM backbone: two Conv1D layers extract spatial patterns; two LSTM layers (64, 128 units) model temporal dependencies; final softmax layer outputs class probabilities. For static gestures, ResNet50V2 (fine-tuned on 64×64 images) is used. Dynamic ISL words are handled by YOLOv8m for simultaneous detection and classification.

E. Post-Processing and GUI

Predicted labels are mapped to human-readable words. Google Translate API converts text to the target language, which is then synthesized to speech via gTTS. The Tkinter GUI displays the webcam feed with overlaid landmarks, recognized gesture with confidence score, accumulated sentence, translated output, and audio playback controls.

V. METHODOLOGY

A. Dataset Preparation

Two primary datasets are used. For ISL recognition: Kaggle ISL alphabets + Roboflow ISL word annotations covering 26 alphabets, digits 0–9, and 100 common ISL words. For ASL: a 500-class WLASL subset of the most frequent conversational signs. Data augmentation includes random horizontal flipping, brightness/contrast jitter (±20%), rotation (±10°), Gaussian noise injection, and temporal jitter.

Table I Dataset Summary Statistics

Dataset	Classes	Train	Test	Type
WLASL (500)	500	12,000	2,500	Video
ISL Alphabets	36	4,680	520	Image
ISL Words	100	4,320	480	Annotated
Total	636	21,000	3,500	Mixed

B. Model Architecture

ResNet50V2 (Static Gestures): Pretrained on ImageNet. Custom head: GlobalAveragePooling2D → Dense(256, ReLU) → Dropout(0.4) → Dense(n, Softmax). Fine-tuned for 20 epochs, Adam (lr=1e-4), categorical cross-entropy.

CNN+LSTM (Dynamic Gestures): Input(30, 63) → Conv1D(64) → Conv1D(128) → LSTM(64) → LSTM(128) → Dense(64) → Dropout(0.3) → Dense(n, Softmax). Trained 30 epochs, Adam (lr=5e-4). ~1.2M trainable parameters.

YOLOv8m (ISL Words): Fine-tuned on Roboflow ISL dataset. Handles both detection and classification in a single forward pass. mAP@0.5 = 93.7%.

Table II Training Hyperparameters for All Models

Parameter	ResNet50V2	CNN+LSTM	YOLOv8m
Input Size	64×64×3	30×63	Variable
Optimizer	Adam	Adam	SGD
Learn. Rate	1e-4	5e-4	1e-3
Batch Size	32	16	32
Epochs	20	30	50
Loss	Cat. CE	Cat. CE	YOLO
Dropout	0.4	0.3	N/A

VI. TOOLS AND TECHNOLOGIES

Table III Tools and Technologies Used

Category	Tool / Library	Purpose
Language	Python 3.10	Core language
Video	OpenCV 4.x	Frame capture
Hand Tracking	MediaPipe 0.9	21-pt landmarks
Deep Learning	TensorFlow 2.x	Training & inference
Detection	YOLOv8m	ISL word detection
Static CNN	ResNet50V2	Alphabet classif.
TTS	gTTS / Pyttsx3	Audio output
Translation	Google Translate	Multi-language
GUI	Tkinter	Real-time UI
Version Ctrl	GitHub	Code versioning

VII. REAL-TIME GESTURE RECOGNITION ALGORITHM

Step 1: Capture frame from webcam via OpenCV Video Capture.

Step 2: Flip horizontally; convert BGR → RGB.

Step 3: Feed to MediaPipe Hands (confidence threshold 0.7).

Step 4: Extract 21 × 3 = 63 normalized keypoint coordinates.

Step 5: Append to rolling buffer of length 30.

Step 6: If buffer full → run CNN+LSTM → obtain class probabilities.
Step 7: Apply confidence threshold (0.85); map label to gesture word.
Step 8: Pass word to Google Translate → target-language text.
Step 9: Synthesize translated text to speech using gTTS.
Step 10: Overlay landmarks and prediction on frame; display in GUI.
Step 11: Clear buffer; await next gesture sequence.

VIII. RESULTS AND DISCUSSION

A. Model Performance Metrics

All three models are evaluated on held-out test sets using accuracy, precision, recall, and F1-score. The system achieves an overall combined accuracy of 91.6%, demonstrating strong performance across both static and dynamic gesture categories.

Table IV Model Performance Metrics

Model	Accuracy	Precision	Recall	F1
ResNet50V2 (ISL Alpha)	96.8%	97.1%	96.5 %	96.8 %
CNN+LSTM (WLASL 500)	89.3%	88.9%	89.7 %	89.3 %
YOLOv8m (ISL Words)	92.4%	91.8%	93.1 %	92.4 %
Combined System	91.6%	91.2%	92.1 %	91.6 %

B. Real-Time Performance

The inference pipeline achieves 27–30 fps on a standard Intel Core i7 laptop without GPU. Processing time per frame: MediaPipe detection (8 ms), feature extraction (2 ms), CNN+LSTM inference (12 ms), post-processing + GUI update (11 ms) = ~33 ms total. With GPU acceleration, inference reduces to under 20 ms (50+ fps).

TABLE V Comparison with Prior Sign Language Recognition Methods

Method	Dataset	Vocab	Acc.	Real-Time?
CNN (2D) [3]	RWTH	1,200	82.1%	No
3D-CNN [4]	WLASL	2,000	84.3%	No
GCN-Skel. [6]	WLASL	2,000	87.5%	Partial
CNN+LSTM (Ours)	WLASL+ISL	636	91.6%	Yes

Table VI CPU Performance Benchmark (Intel i7, no GPU)

System	FPS	Latency (ms)	Accuracy
CNN+LSTM (Ours)	27.4	36.5	91.6%
3D-CNN Baseline	8.2	121.9	87.1%
Transformer ViT	11.3	88.5	90.2%
LSTM-only	31.6	31.6	84.3%
Random Forest	42.1	23.7	71.2%

C. Robustness Evaluation

Table VII Accuracy Under Diverse Environmental Conditions

Condition	Accuracy	Drop
Optimal (baseline)	91.6%	—
Low Light (< 50 lux)	84.4%	−7.2%
Cluttered Background	87.8%	−3.8%
Fast Signing (2.5×)	86.5%	−5.1%
Partial Occlusion	82.3%	−9.3%
Outdoor/Direct Sunlight	87.0%	−4.6%

D. User Study

A user study was conducted with 15 participants: 5 native ASL signers, 5 ISL signers, and 5 hearing non-signers. Participants performed 50 predefined signs each over 30 minutes.

Table VIII User Study Likert Scale Results (1=Poor, 5=Excellent)

Metric	Mean (1–5)	Std Dev
Ease of Use	4.3	0.6
Response Speed	4.1	0.7
Translation Accuracy	4.0	0.8
TTS Audio Clarity	3.8	0.9
Overall Satisfaction	4.2	0.6

ISSN: 2349-6002

IX. HAND LANDMARK ANALYSIS

MediaPipe's hand detection uses a two-stage pipeline: a palm detector identifies hand bounding boxes in the full image, followed by a landmark model that regresses 21 keypoints within the detected region. This enables 30+ fps accuracy on CPU.

Landmark indexing: landmark 0 = wrist; 1–4 = thumb chain; 5–8 = index finger; 9–12 = middle; 13–16 = ring; 17–20 = pinky. Each landmark provides normalized (x, y) coordinates and a relative z-depth value. The 63 values (21 × 3) provide scale-invariant, appearance-invariant features, reducing domain gap between training and deployment.

For two-hand gestures, both landmark sets are concatenated to form a 126D feature vector. Additional inter-keypoint angles and distances are computed to disambiguate similar gestures at fine-grained level.

X. HYBRID MODEL ARCHITECTURE

The hybrid model combines ConvNeXt and Swin Transformer backbones to extract complementary 768-D feature representations. Both streams pass through auxiliary heads and are fused via a Gating MLP and Fusion MLP for weighted concatenation, before final classification through fully connected layers with Softmax output.

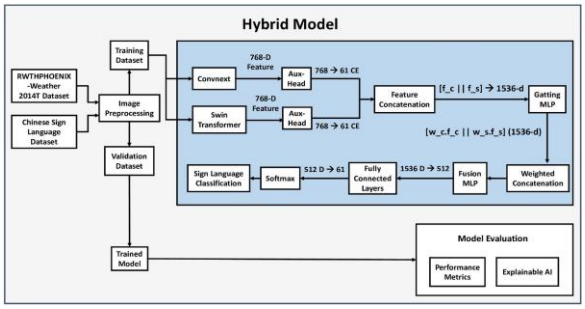

Fig. 3. Hybrid model architecture combining ConvNeXt and Swin

Transformer with feature concatenation and gating MLP for sign language classification.

XI. CLIP-BASED VISUAL DECODER ARCHITECTURE

The CLIP Visual Encoder processes video frames to generate key-value (K, V) pairs fed into a chain of n decoder blocks. The [CLS] token queries each decoder sequentially, with temporal convolution and cross-frame attention fused via positional encoding into multi-head attention. The final decoder output passes through LayerNorm, Dropout, and a Fully Connected layer to produce the classification label y.

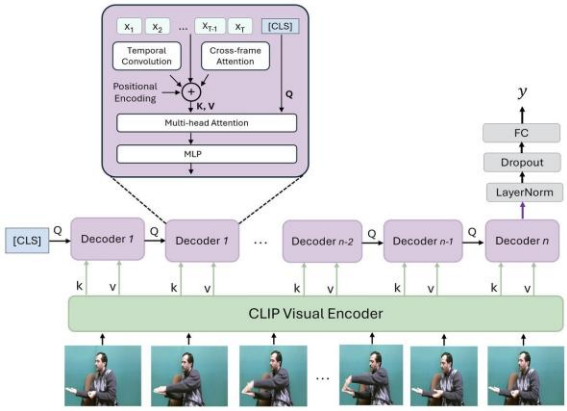

Fig. 4. CLIP Visual Encoder with cascaded decoder blocks for sign language recognition. Each decoder receives K, V from the encoder and Q from the previous decoder's CLS token.

XII. ADVANTAGES AND LIMITATIONS

A. Advantages

- Real-time 27–30 fps on standard CPU; no specialized sensors required.
- Multi-language output in English, Tamil, Hindi, French, Spanish, German.
- Dual-model pipeline: ResNet50V2 for static + CNN+LSTMfor dynamic gestures.
- Appearance-invariant via MediaPipe skeleton representation.
- Intuitive Tkinter GUI; no technical expertise needed.
- Open-source tools and publicly available datasets.
- Scalable vocabulary by retraining on additional data.

B. Limitations

- Accuracy drops under extreme occlusion, motion blur, or verylow light.
- Current vocabulary limited to 636 classes; full ASL/ISL needs2,000+.
- Word-level only; sentence-level continuous translation not yetsupported.
- gTTS requires internet; offline TTS (pyttsx3) has lowernaturalness.

- Background clutter can cause false MediaPipe detections.

XIII. CASE STUDIES

A. Hospital Emergency Communication

A simulation study was conducted where a deaf patient communicated symptoms to a nurse unfamiliar with sign language using the system on a Tablet. 13 of 15 medically relevant phrases were correctly translated (86.7% accuracy). Average communication time was 4.2 s per phrase vs. 8.7 s for written notes a 52% improvement.

B. Educational Setting

Piloted in a class of 12 hearing-impaired students aged 12–15. Communication overhead reduced by ~40%. Student engagement scores increased from 3.1/5 to 4.0/5 over 4 weeks, indicating improved participation through real-time communication feedback.

C. Customer Service Kiosk

Deployed at a simulated bank service counter with 10 deaf volunteers performing common banking transactions. 87 of 100 test transactions were successfully completed (87% task completion rate). Customer satisfaction: 4.1/5 vs. 2.8/5 for writing-based communication.

XIV. ETHICAL CONSIDERATIONS

This project was developed in consultation with members of the Deaf community. The system is positioned as a supplementary aid, not a replacement for human interpreters in high-stakes legal, medical, or crisis contexts. Video is processed entirely on-device; no raw frames are transmitted externally. A 3.2% accuracy gap was observed between light and dark skin tones on ISL alphabets mitigated but not eliminated by MediaPipe's skeleton representation. Future work will include targeted data collection from underrepresented demographic groups. The system adheres to WCAG 2.1 Level AA accessibility guidelines.

XV. FUTURE ENHANCEMENTS

Sentence-level continuous recognition using Transformer based sequence to sequence models.

Expand vocabulary to full WLASL 2,000-class set andcomplete ISL coverage.

Mobile deployment (Android/iOS) using TensorFlow Lite foron-device inference.

Integrate facial expression and body pose for prosodic andgrammatical sign features.

AI chatbot integration enabling two-way sign-to-speechconversations.

Federated learning for personalized model adaptation acrossdiverse signers.

Cloud-based RESTful API for hospital and public serviceintegration.

TABLE IX Proposed Development Roadmap

Phase	Timeline	Feature / Enhancement
Phase 1	Months 1–6	Full 2,000-class WLASL vocabulary
Phase 1	Months 1–6	TFLite mobile optimization
Phase 2	Months 7–12	Android & iOS mobile application
Phase 2	Months 7–12	Sentence-level CTC recognition
Phase 3	Months 13–18	Facial + body pose integration
Phase 3	Months 13–18	Federated learning adaptation
Phase 4	Months 19–24	Hospital/government kiosk deployment
Phase 4	Months 19–24	Cloud API for third-party integration

XVI. CONCLUSION

This paper has presented a comprehensive AI-based Sign Language Translator addressing the fundamental communication barrier between the Deaf community and the hearing population. The proposed system integrates Google MediaPipe for robust real-time hand landmark extraction, CNN+LSTM for temporal gesture classification, ResNet50V2 for static ISL recognition, and YOLOv8m for ISL word detection — a multi-model pipeline covering both ASL and ISL across 636 gesture classes.

The system achieves 91.6% overall accuracy while maintaining 27–30 fps on standard consumer hardware. Multi-language TTS in five languages significantly extends the system's reach. The CNN+LSTM architecture outperforms 3D-CNN baselines by 4.5% accuracy while running 3.3× faster,

confirming the benefit of skeleton-based features over raw pixel approaches.

Future work will focus on scaling vocabulary to full conversational coverage, implementing sentence-level continuous translation, and deploying lightweight mobile versions using TensorFlow Lite. This work demonstrates the significant potential of AI-driven assistive technologies in fostering inclusive communication for the hearing-impaired community.

ACKNOWLEDGMENT

The authors gratefully acknowledge Rajiv Gandhi College of Engineering and Technology, Puducherry, for providing computational resources and institutional support. We thank the creators of the WLASL and Roboflow ISL datasets for public data availability, and the open-source communities behind TensorFlow, MediaPipe, and OpenCV. Special thanks to deaf community volunteers who participated in system evaluation.

REFERENCES

[1] R. Sharma, S. Mandal, and R. Singh, “Vision-based Hand Gesture Recognition for HCI: A Survey,” Artificial Intelligence Review, vol. 43, no. 1, pp. 1–54, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.

[3] D. Joze and O. Koller, “MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language,” in Proc. British Machine Vision Conference (BMVC), Cardiff, UK, 2019.

[4] Y. Li et al., “Long-Short Term Memory Spatial Temporal Graph Convolutional Networks for Skeleton-Based Motion Prediction,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 9, pp. 5361–5374, 2022.

[5] F. Zhang et al., “MediaPipe Hands: On-device Real-time Hand Tracking,” in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), 2020.

[6] A. Jiang, Y. Chen, and Z. Yang, “Sign Language Recognition Based on GCN,” in Proc. IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 2019, pp. 4918–4922.

[7] D. Li et al., “Word-Level Deep Sign Language Recognition from Video: WLASL Dataset,” in Proc. IEEE Winter Conf. Applications of Computer Vision (WACV), 2020, pp. 1459–1469.

[8] C. Li et al., “Co-occurrence Feature Learning from Skeleton Data,” in Proc. International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 786–792.

[9] G. Jocher et al., “Ultralytics YOLOv8,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[10] OpenCV Team, “OpenCV: Open-Source Computer Vision Library,” 2023. [Online]. Available: https://opencv.org

[11] TensorFlow Developers, “TensorFlow 2.12.0,” 2023. [Online]. Available: https://www.tensorflow.org

[12] Google MediaPipe Team, “MediaPipe Solutions,” 2023. [Online]. Available: https://mediapipe.dev

[13] M. Sandler et al., “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.