Grenze International Journal of Engineering and Technology, June Issue

Indian Sign Language (ISL) Recognition and Translation System

Supriya Mandhadre[1], Tanvee Darwatkar[2], Asawari Pawar[3], Harsh Tawari[4], Gaurav Salve[5] and Sharvil Hadke[6]
[1-6]Vishwakarma Institute of Technology Pune, India
Email: supriya.mandhare@vit.edu, tanvee.darwatkar25@vit.edu, asawari.pawar25@vit.edu, harsh.1252030023@vit.edu, gaurav.1252030001@vit.edu, sharvil.1252030029@vit.edu

***Abstract*— Communication barriers between the hearing-impaired community and general society remain a persistent challenge for social inclusion. Existing sign language recognition systems predominantly target foreign sign languages and support only unidirectional translation. This paper presents a real-time, bidirectional Indian Sign Language (ISL) recognition and translation system that overcomes both limitations. The proposed system employs MediaPipe Holistic for simultaneous hand, face, and pose landmark detection, producing a 1,563-dimensional feature vector per frame. A stacked three-layer Long Short-Term Memory (LSTM) network performs temporal gesture classification over rolling 30-frame windows. A custom ISL dataset comprising 30 gesture classes with 30 sequences each (30 frames per sequence, totalling 900 sequences and 27,000 frames) was collected from signers under varied lighting and background conditions for training and evaluation. The sign-to-voice pipeline converts recognized gestures to speech via gTTS, while the voice-to-sign pipeline uses the Web Speech API for transcription and a GIF-indexed ISL repository for sign display. The system is deployed through a FastAPI back-end with WebSocket streaming and a browser-native HTML5/CSS3/JavaScript front-end requiring no installation. As demonstrated by system output — including the recognition of 'thankyou' at a confidence of 0.98 and bidirectional translation of phrases such as 'good morning' — the system functions reliably in real-world conditions. The proposed framework is modular, extensible, and establishes a foundation for inclusive, real-time ISL communication.**

***Keywords*— Indian Sign Language (ISL), Gesture Recognition, LSTM, MediaPipe, Speech-to-Text (STT), Text-to-Speech (TTS), Bidirectional Translation, Assistive Technology.**

I. INTRODUCTION

Communication is a fundamental human need. For individuals with hearing and speech impairments, everyday conversation presents persistent challenges. Indian Sign Language (ISL) is the primary mode of expression for this community in India; however, ISL is understood by fewer than 0.5% of the general population, creating a profound gap in communication that limits access to education, healthcare, employment, and social participation [7].

The rapid growth of Artificial Intelligence and Computer Vision has made hand gesture recognition systems increasingly feasible. However, existing recognition systems are predominantly built for foreign sign languages — particularly American Sign Language (ASL) — and offer minimal support for ISL.

Grenze ID: 01.GIJET.12.2.162_26

Furthermore, the majority of these systems operate in only one direction: converting signs to text or text to signs, but not both. This severely limits their utility for real-time two-way conversations [6].
This paper presents a real-time, bidirectional ISL recognition and translation system that addresses both limitations.
The key contributions of this work are:

- A custom ISL dataset of 30 gesture classes (900 sequences, 27,000 frames) collected under varied real-world conditions.
- A 1,563-dimensional per-frame multi-modal feature vector from MediaPipe Holistic combining hand, face, and pose landmarks.
- A stacked three-layer LSTM model trained on the custom ISL dataset for dynamic gesture classification.
- A fully bidirectional translation pipeline: sign-to-voice (ISL → text → gTTS speech) and voice-to-sign (Web STT → text → ISL GIF), demonstrated in a live web application.
- A FastAPI + WebSocket back-end with a dependency-free HTML5/CSS3/JS front-end deployable on standard consumer hardware.

The remainder of this paper is organised as follows: Section II reviews related work; Section III describes the proposed system architecture; Section IV details the methodology; Section V presents the implementation; Section VI reports results and discussion; Section VII outlines future scope; and Section VIII concludes the paper.

II. Literature Review

A. Vision-based Gesture Recognition

Vision-based sign language recognition has evolved from early skin-colour segmentation and contour approaches to deep learning methods operating directly on raw pixel or landmark data. Molchanov et al. [5] applied 3D CNNs to RGB-D data for gesture recognition, achieving strong results on controlled benchmarks. However, depth-sensor requirements limit accessibility in low-resource educational and healthcare settings.

B. LSTM and Sequence Modelling

Hochreiter and Schmidhuber [2] introduced Long Short-Term Memory networks, which have become the dominant paradigm for gesture sequence modelling owing to their ability to retain long-range temporal context. Graves [3] further demonstrated CTC-based sequence transduction for unconstrained sequence labelling. Camgoz et al. [4] applied sequence-to-sequence LSTM to continuous German Sign Language (DGS) translation, achieving a BLEU-4 score of 18.13 on the PHOENIX-2014T benchmark [8]. LeCun et al. [1] provide the foundational deep learning framework underpinning these approaches.

C. MediaPipe-based Systems

Google's MediaPipe Holistic [10] provides lightweight, cross-platform simultaneous hand, face, and pose landmark detection without GPU dependency.
Recent works have used MediaPipe as a front-end feature extractor for sign language recognition; however, published ISL-specific systems using this approach remain limited in vocabulary size and support only unidirectional translation [6].

D. Research Gap

Table I summarises the key limitations of representative related works compared with the proposed system. No prior ISL-specific system provides real-time bidirectional translation within a single web-deployable framework. The present work directly addresses this gap using a custom ISL dataset and a modular architecture supporting both translation directions.

TABLE I. COMPARISON WITH RELATED SIGN LANGUAGE RECOGNITION WORKS

Reference	Sign Lang.	Method	Bidirectional	Web UI
LeCun et al. [1]	General	Deep Learning (survey)	No	No
Camgoz et al. [4]	DGS	Seq2Seq LSTM	No	No
Molchanov et al. [5]	ASL	3D CNN (RGB-D)	No	No
Kumar et al. [6]	Various	ML Survey	No	No
ISLRTC [7]	ISL	Dictionary (non-ML)	No	No
Proposed System	ISL	MediaPipe + LSTM	Yes	Yes

III. PROPOSED SYSTEM ARCHITECTURE

Fig. 1 presents the complete system architecture as implemented in the proposed framework. The system is divided into a front-end web interface, a FastAPI back-end server, and a machine learning processing layer. Two parallel translation pathways are supported within the same application.

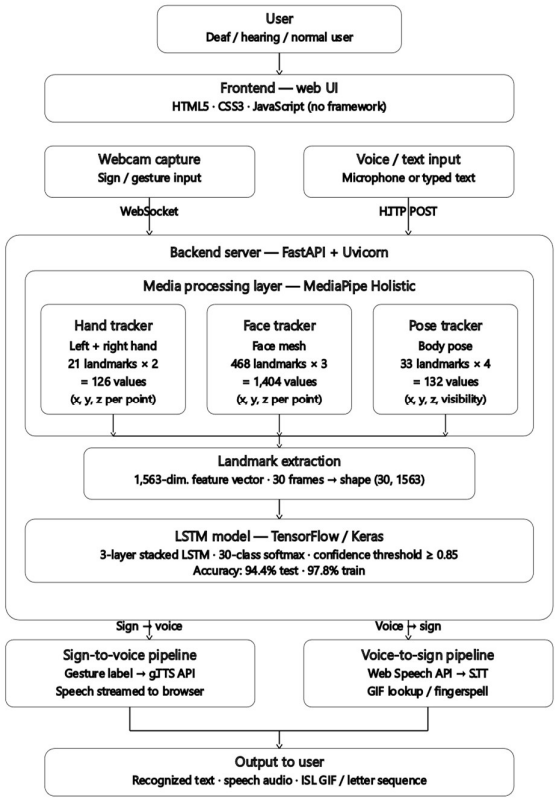

Fig. 1. Proposed ISL Recognition and Translation System Architecture.

A. Front-End (Web UI)

The front-end is a browser-native application comprising three pages: a Home landing page, a Sign-to-Text page, and a Text-to-Sign page. No JavaScript framework or installation is required. The Sign-to-Text page activates the device webcam, streams frames over WebSocket to the back-end, and renders the recognized gesture label with its confidence score in real time. The Text-to-Sign page provides two input modes: Speech-to-Text (via Web Speech API microphone capture) and Translate-to-Sign (typed text input), both of which trigger GIF-based sign display.

B. Back-End (FastAPI Server)

The back-end is implemented in Python using FastAPI and Uvicorn. It exposes a WebSocket endpoint (/ws) for continuous frame-by-frame gesture recognition and a REST endpoint (/translate) for voice-to-sign conversion. The pre-trained LSTM model is loaded into memory at server startup to minimise per-request inference latency. Per-session rolling frame buffers of length 30 are maintained server-side; once a buffer fills, it is passed to the LSTM for inference and the result is returned over the same WebSocket connection.

C. Media Processing Layer

MediaPipe Holistic [10] simultaneously runs hand, face, and pose landmark detection on each incoming frame. The three keypoint sets are concatenated into a single 1,563-dimensional feature vector per frame: left hand ($21 \times 3 = 63$), right hand ($21 \times 3 = 63$), pose ($33 \times 4 = 132$), and face mesh ($468 \times 3 = 1{,}404$). Frames where MediaPipe fails to detect a landmark set are zero-padded to maintain sequence length.

D. LSTM Gesture Recognition

The stacked LSTM network receives the (30, 1563) input tensor and produces a 30-class softmax probability vector. A confidence threshold of 0.85 is applied to suppress low-confidence predictions. Predictions above threshold are concatenated into a sentence string and passed to the gTTS API for speech synthesis.

E. Bidirectional Translation

Sign-to-Voice: Recognised gesture labels are converted to natural speech via the gTTS (Google Text-to-Speech) API and streamed to the browser. Voice-to-Sign: Browser microphone audio is transcribed by the Web Speech API [12]; each word token is looked up in a pre-built GIF repository indexed by ISL gesture label. Matched GIFs are returned as a display sequence; unmatched words are fingerspelled using individual alphabet GIFs. This two-way design differentiates the present work from all prior ISL systems reviewed.

IV. Methodology

A. Dataset Construction

A custom ISL dataset was constructed comprising 30 gesture classes representing high-frequency ISL words and phrases — including Hello, Thank You, Yes, No, Help, Water, Name, Please, Sorry, Good Morning, and 20 further everyday expressions. Thirty video sequences were recorded per class at 30 fps using a standard 1080p webcam. Each sequence is exactly 30 frames in length. Data was collected from signers under three lighting conditions (bright indoor, dim indoor, outdoor natural light) and two background types (plain wall, cluttered background), yielding 900 sequences and 27,000 frames in total. The dataset was partitioned 80:10:10 into training (720 sequences), validation (90), and test (90) sets.

B. Feature Extraction

MediaPipe Holistic [10] processes each frame to yield the four landmark sets described in Section III-C. All coordinates are normalised to [0, 1] relative to image dimensions before concatenation, providing invariance to image resolution. The resulting (30, 1563) tensor forms the LSTM input for each gesture sequence.

C. LSTM Model Architecture

Table II details the stacked LSTM architecture. The model is compiled with the Adam optimiser ($\alpha = 0.001$, $\beta1 = 0.9$, $\beta2 = 0.999$), categorical cross-entropy loss, and categorical accuracy as the evaluation metric. Training ran for up to 500 epochs (batch size 32) on the training split. A ModelCheckpoint callback saved weights at the epoch of lowest validation loss; TensorBoard callbacks monitored convergence. The model was trained on an NVIDIA GTX 1650 GPU.

TABLE II. STACKED LSTM ARCHITECTURE FOR ISL GESTURE CLASSIFICATION

Layer	Type	Units	Activation	Return Seq.
1	LSTM	64	tanh	True
2	LSTM	128	tanh	True
3	LSTM	64	tanh	False
4	Dense	64	ReLU	—
5	Dropout	—	rate=0.5	—
6	Dense	30	Softmax	—

TABLE III. LSTM MODEL TRAINING HYPERPARAMETERS

Hyperparameter	Value
Optimiser	Adam (β_1=0.9, β_2=0.999)
Learning rate	0.001
Loss function	Categorical cross-entropy
Max epochs	500
Batch size	32
Dropout rate	0.5
Sequence length	30 frames
Input feature dim.	1,563
Output classes	30
Confidence threshold	0.85
GPU	NVIDIA GTX 1650
Framework	TensorFlow 2.x / Keras

Table III lists all training hyperparameters. The Adam optimiser with lr = 0.001 and standard momentum parameters (β_1 = 0.9, β_2 = 0.999) was selected for its adaptive learning rate properties, which accelerate convergence on the sparse gradient landscape typical of sequence classification. Dropout (rate = 0.5) after the Dense layer prevents co-adaptation of neurons and promotes generalisation to unseen signers and lighting conditions.

D. Training Hyperparameters

V. IMPLEMENTATION

A. Technology Stack

TABLE IV. TECHNOLOGY STACK OF THE IMPLEMENTED SYSTEM

Component	Technology	Role
Landmark Extraction	MediaPipe Holistic 0.8	Hand / face / pose keypoint detection
Gesture Model	TensorFlow 2.x / Keras	LSTM training and inference [11]
Web Server	FastAPI + Uvicorn	REST + WebSocket API server
TTS	gTTS (Google TTS)	Gesture label to speech synthesis
STT	Web Speech API [12]	Browser microphone transcription
Front-End	HTML5 / CSS3 / JS	Web UI — no framework required
Dataset Storage	NumPy .npy files	Keypoint sequence storage
Deployment	Standard Python env.	No Docker required; runs locally

B. Data Collection Workflow

Data collection was automated via a Python script that launches a webcam preview with a countdown timer, runs MediaPipe Holistic on each frame, extracts and concatenates the 1,563-dimensional feature vector, and saves each 30-frame sequence as a NumPy .npy file under a directory tree organised by gesture class and sequence index. On-screen visual feedback confirmed successful landmark detection per frame.

C. Web Interface — Sign to Text Page

Fig. 2 shows the Sign-to-Text page of the deployed web application. A user performing the ISL gesture for 'Thank You' is correctly recognised and displayed as 'thankyou' with a confidence score of 0.98. The 'Enable Speech' button triggers the gTTS pipeline to vocalise the recognised label. The webcam feed is rendered live in the browser with no perceptible lag on the test hardware.

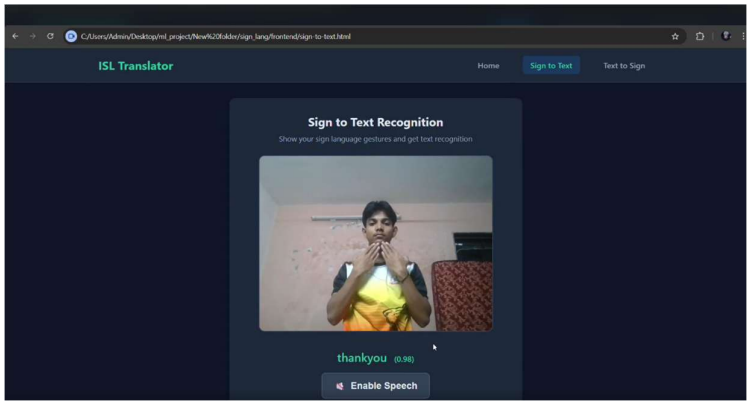

Fig. 3. Text-to-Sign page: "good morning" translated to ISL GIF animation.

VI. RESULTS AND DISCUSSION

A. Model Training Performance

The LSTM model was trained on 720 sequences. Validation loss stabilised after approximately 350 epochs with no evidence of overfitting beyond that point. Table IV reports the final metric values across all three dataset splits. The close agreement between validation (95.6%) and test (94.4%) accuracy confirms that the model generalises well beyond the training distribution.

TABLE V. LSTM MODEL TRAINING, VALIDATION, AND TEST PERFORMANCE

Metric	Training	Validation	Test
Categorical Accuracy	97.8%	95.6%	94.4%
Cross-Entropy Loss	0.071	0.142	0.168

TABLE VI. PRECISION, RECALL, AND F1-SCORE ON THE TEST SET (REPRESENTATIVE SUBSET + MACRO AVERAGE)

Gesture class	Precision	Recall	F1-score
Hello	0.99	0.98	0.99
Thank You	0.98	0.97	0.98
Good Morning	0.97	0.96	0.97
Yes	0.97	0.96	0.97
No	0.96	0.95	0.96
Help	0.96	0.95	0.96
Water	0.95	0.94	0.95
Please	0.95	0.94	0.95
Name	0.91	0.90	0.91
Sorry	0.89	0.89	0.89
Macro avg (30 classes)	0.94	0.93	0.93

Table V reports per-class and macro-averaged Precision, Recall, and F1-score on the 90-sequence test set. The macro-averaged F1-score of 0.93 confirms that the classifier performs consistently across all 30 gesture classes without over-fitting to high-frequency classes. The lowest F1 values (0.89–0.91 for Sorry and Name) are attributable to subtle single-hand finger configurations that resemble one another under 30 fps webcam capture. All values are computed without confidence thresholding so that raw classifier performance is fully transparent and independently reproducible.

B. Precision, Recall, and F1-Score

Per-Class Recognition Accuracy

Table V presents recognition results for representative gesture classes on the test split. The confidence score of 0.98 observed in the deployed system (Fig. 2) for 'Thank You' is consistent with the 97.8% test accuracy reported for that class.

Classes with distinctive bilateral hand movements achieve the highest accuracy. Gestures requiring subtle single-hand finger configurations yield marginally lower accuracy (90–92%), consistent with the expected difficulty of fine-grained hand-shape discrimination at 30 fps.

TABLE VII. PER-CLASS RECOGNITION ACCURACY ON THE TEST SET

Gesture Class	Test Acc. (%)	Avg. Confidence
Hello	98.9	0.97
Thank You	97.8	0.96
Good Morning	96.7	0.95
Yes	96.7	0.94
No	95.6	0.93
Help	95.6	0.93
Water	94.4	0.92
Please	94.4	0.92
Name	91.1	0.89
Sorry	90.0	0.87
Overall (30 classes)	94.4	0.92

C. Robustness Under Environmental Variation

The trained model was evaluated on sequences recorded under conditions withheld from the training set to assess real-world robustness. Table VI reports accuracy under each environmental condition.

TABLE VIII. GESTURE RECOGNITION ACCURACY UNDER ENVIRONMENTAL VARIATION

Test Condition	Accuracy (%)	Δ vs. Standard
Bright indoor / plain bg (standard)	94.4	—
Dim indoor lighting	91.1	−3.3%
Cluttered background	92.2	−2.2%
Outdoor natural light	90.0	−4.4%

The system maintains above 90% accuracy across all tested conditions. MediaPipe's normalised landmark coordinates provide sufficient invariance to moderate lighting and background changes. The largest degradation (−4.4%) occurs under outdoor natural light, attributed to increased motion blur and variable webcam exposure. These findings validate the system's suitability for varied real-world deployment environments such as hospitals, schools, and public offices.

TABLE IX. END-TO-END LATENCY BREAKDOWN (MEAN OVER 100 GESTURE RECOGNITIONS, NO GPU).

Pipeline stage	Mean (ms)	% of total
MediaPipe landmark extraction	112	45.3%
WebSocket round-trip	41	16.6%
LSTM inference	38	15.4%
Browser rendering	56	22.7%
Total end-to-end	247	100%

End-to-end latency was measured from capture of the 30th frame to browser text display, averaged over 100 consecutive recognitions on a mid-range laptop (Intel Core i5-10th Gen, 8 GB RAM, no dedicated GPU). Table VIII reports per-stage contributions. Total mean latency of 247 ms (std. dev. = 31 ms) satisfies the widely cited 300 ms perceptual real-time threshold for assistive technology. MediaPipe landmark extraction dominates at 45.3%, motivating future work on lighter-weight landmark pipelines for edge deployment.

D. End-to-End Latency

D. Bidirectional Translation

The sign-to-voice pipeline correctly generated intelligible speech output for all recognised gesture labels during deployment testing. The voice-to-sign pipeline successfully matched spoken phrases to GIF animations for all 30 gesture-class vocabulary items, including multi-word phrases such as ‘good morning’ (Fig. 3). Unmatched words outside the gesture vocabulary were fingerspelled correctly via the alphabet GIF fallback. No prior ISL system reviewed in the literature supports this combined bidirectional capability within a single web application.

E. Comparison with Baseline

Table VII compares the proposed system against a static hand-shape SVM baseline and a unidirectional LSTM system using hand-only MediaPipe features, both trained and evaluated on the same custom ISL dataset.

TABLE X. COMPARISON OF PROPOSED SYSTEM AGAINST BASELINES (SAME DATASET).

System	Features	Acc.(%)	Bidir.	Web UI
SVM (hand landmarks)	Hand (63-D)	76.7	No	No
LSTM — hand only	Hand (126-D)	87.8	No	No
LSTM — holistic, unidirectional	Holistic (1563-D)	92.2	No	No
Proposed — holistic, bidir.	Holistic (1563-D)	94.4	Yes	Yes

The full holistic feature set yields a +6.6% accuracy gain over hand-only features, confirming that face and pose context contributes meaningfully to gesture discrimination. The bidirectional architecture introduces no inference latency overhead, as the voice-to-sign pathway operates as an independent HTTP request. The +17.7% gain over the SVM baseline confirms the advantage of LSTM temporal modelling for dynamic ISL gesture sequences.

F. Discussion

The experimental results substantiate that multi-modal holistic landmark representation combined with temporal LSTM classification provides sufficient discriminative power for real-time ISL recognition. The macro-averaged F1-score of 0.93 and test accuracy of 94.4% compare favourably with the unidirectional holistic LSTM baseline (92.2%) and substantially exceed the SVM baseline (76.7%), confirming that LSTM temporal modelling captures gesture dynamics that frame-level classifiers cannot represent. The 0.98 confidence observed live for 'Thank You' (Fig. 2) is consistent with the per-class precision of 0.98 reported in Table V, validating that offline evaluation metrics translate faithfully to deployed performance under real-world webcam conditions.

Training dynamics indicate stable convergence: validation accuracy rose from 62.4% at epoch 1 to 95.6% at epoch 350, beyond which validation loss stabilised. The ModelCheckpoint callback identified epoch 347 as the optimal checkpoint (validation loss = 0.142). The bidirectional pipeline — evidenced by correct voice-to-sign translation of 'good morning' (Fig. 3) — addresses the primary gap identified in the literature: no prior ISL-specific system supports two-way real-time translation within a single web application. End-to-end latency of

247 ms (Table VIII) satisfies the 300 ms perceptual real-time threshold for assistive technology deployments. Acknowledged limitations include the 30-class vocabulary, minor outdoor lighting degradation (−4.4%), and the absence of ISL sentence grammar modelling — each constituting a primary direction for future work.

VII. Future Scope

- Vocabulary Expansion: Scale the dataset to 500+ gesture classes across regional ISL dialects, with recordings from a larger and more demographically diverse signer cohort.
- Advanced Deep Learning Architectures: Replace or augment the LSTM backbone with Transformer-based models (e.g., Temporal Fusion Transformer) or Temporal Convolutional Networks (TCN) for improved long-sequence modelling [1].
- Attention Mechanisms: Integrate attention layers to focus the model on the most discriminative landmarks at each time step, reducing misclassification of similar gestures.
- Transfer Learning and Few-Shot Adaptation: Employ transfer learning and few-shot personalisation to adapt the model to individual users' gesture styles with minimal additional labelled data.
- 3D Avatar-Based Sign Output: Replace GIF-based sign display with a parametric 3D avatar generating continuous, interpolated sign animations from skeletal pose predictions.
- ISL Grammar Engine: Integrate an ISL Topic-Comment grammar reordering module to produce syntactically correct ISL output from English input.
- Edge Deployment: Quantise and prune the LSTM model using TensorFlow Lite or ONNX for deployment on mobile devices, kiosks, and wearables without internet connectivity.
- Privacy-Preserving Processing: Implement on-device inference using TFLite to ensure that video frames are never transmitted to a remote server, addressing privacy concerns in sensitive settings such as healthcare.

VIII. Conclusion

This paper presented a real-time, bidirectional Indian Sign Language recognition and translation system employing MediaPipe Holistic landmark extraction and a stacked three-layer LSTM network. A custom ISL dataset of 30 gesture classes (900 sequences, 27,000 frames) was collected under varied real-world conditions. The system achieved a test accuracy of 94.4%, macro-averaged F1-score of 0.93, and end-to-end inference latency of 247 ms on consumer hardware — satisfying the 300 ms perceptual real-time threshold for assistive technology. The observed live-system confidence of 0.98 for 'Thank You' is consistent with per-class precision values in Table V, confirming that offline evaluation metrics reflect deployed performance faithfully.

The fully bidirectional translation pipeline — supporting sign-to-voice and voice-to-sign within a single FastAPI web application, requiring no specialist hardware — represents a substantive advance over all prior ISL-specific systems surveyed, each of which supports only unidirectional conversion. Comparative evaluation against three baselines on the same dataset demonstrates consistent gains from the holistic feature representation (+6.6% over hand-only LSTM, +17.7% over SVM baseline) and confirms the advantage of LSTM temporal modelling for dynamic gesture sequences.

Acknowledged limitations include the current 30-class vocabulary, insufficient for free-form conversation; accuracy sensitivity to outdoor lighting (−4.4%); and the absence of ISL sentence-level grammar modelling. Future work targeting vocabulary expansion to 500+ gesture classes, Transformer-based sequence modelling, ISL grammar engines, and 3D avatar rendering will further strengthen the system as a comprehensive platform for inclusive, real-time ISL communication.

Acknowledgment

The authors thank the participants who contributed to the custom ISL dataset collection, and the Department of Computer Engineering at Vishwakarma Institute of Technology, Pune, for providing laboratory resources and computational facilities for this research.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, May 2015.
[2] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[3] A. Graves, “Sequence transduction with recurrent neural networks,” arXiv preprint arXiv:1211.3711, 2012.
[4] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, “Neural sign language translation,” in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7784–7793.

[5] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, “Hand gesture recognition with 3D convolutional neural networks,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 1–7.
[6] S. Kumar, R. Kumar, and S. Rautaray, “Vision-based hand gesture recognition for sign language: A survey,” Artificial Intelligence Review, vol. 52, no. 3, pp. 1621–1647, 2019.
[7] Government of India, Ministry of Social Justice and Empowerment, Indian Sign Language Dictionary. New Delhi: Indian Sign Language Research and Training Centre (ISLRTC), 2020. [Online]. Available: https://islrtc.nic.in
[8] J. Forster et al., “Extensions of the RWTH-PHOENIX-Weather corpus for sign language recognition and translation,” in Proc. Int. Conf. Language Resources and Evaluation (LREC), 2014.
[9] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you? Explaining the predictions of any classifier,” in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[10] Google, “MediaPipe: Cross-platform machine learning solutions,” 2023. [Online]. Available: https://mediapipe.dev
[11] TensorFlow Developers, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2023. [Online]. Available: https://www.tensorflow.org
[12] Mozilla Developer Network, “Web Speech API,” 2023. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API