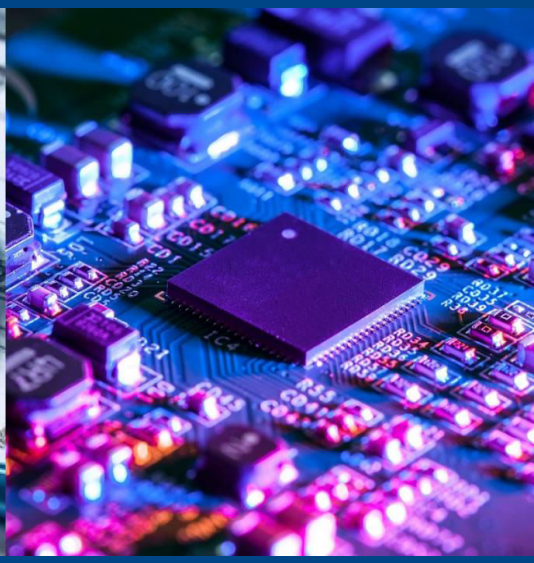
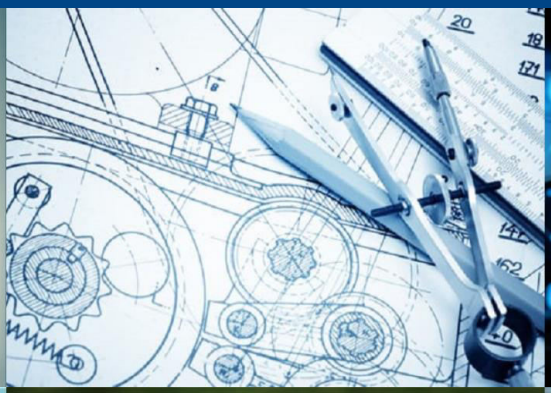
ISSN: 2582-7219

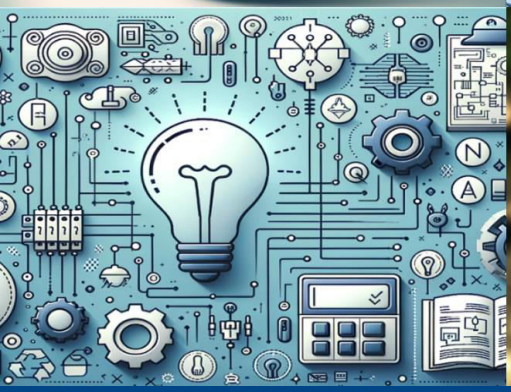




# International Journal of Multidisciplinary Research in Science, Engineering and Technology

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 9.864**

**Volume 9, Issue 5, May 2026**

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

# Real-Time Sign Language Conversion into Text and Audio

**Prof. Dr. Aniruddha S Rumale[1], Vishakha Sarnaik[2], Anushka Gosavi[3], Aryan Chapne[4], Pratap Tupe[5]**

Assistant Professor, Dept. of Information Technology, Sandip Institute of Technology and Research Centre, Nashik, India[1]

UG Students, Dept. of Information Technology, Sandip Institute of Technology and Research Centre, Nashik, India[2,3,4,5]

**ABSTRACT:** The system is an end-to-end deep learning system that converts Indian Sign Language (ISL) gestures into text and speech in real time. The system captures live hand motion through a standard webcam, extracts 21-landmark skeletal data from each hand using Google MediaPipe, assembles 30-frame temporal sequences, and classifies them with a stacked Long Short-Term Memory (LSTM) neural network built on TensorFlow/Keras. Predicted text is displayed on a modern web dashboard (Flask + Socket.IO) and optionally spoken aloud via text-to-speech. The project addresses a critical accessibility gap: approximately 5 million deaf and hard-of-hearing people in India use ISL as their primary language, yet real-time, affordable translation tools remain scarce. By combining computer vision and sequence modelling, ISL·Bridge demonstrates that a complete gesture-recognition pipeline can be implemented with open-source tools and consumer hardware.

**KEYWORDS:** Indian Sign Language, ISL, LSTM, MediaPipe, real-time gesture recognition, deep learning, accessibility, computer vision, Flask, TensorFlow

## I. INTRODUCTION

### 1.1 Background & Motivation

Communication is a fundamental human right. For the approximately 5 million deaf and hard-of-hearing individuals in India, Indian Sign Language (ISL) serves as the primary medium of expression. However, the overwhelming majority of the hearing population has no knowledge of ISL, creating a significant communication barrier in healthcare, education, employment and daily life.

Existing solutions — human interpreters, text-relay services, and commercial applications — suffer from high cost, limited availability, and language-pair restrictions. A software system capable of real-time, device-local ISL recognition could dramatically reduce this gap without ongoing infrastructure costs.

### 1.2 Problem Statement

How can hand gestures conforming to the Indian Sign Language lexicon be automatically translated into readable text (and speech) in real time on consumer hardware, with no internet dependency?

### 1.3 Scope

| IN SCOPE | OUT OF SCOPE |
|---|---|
| 26 ISL alphabet letters (A–Z) | Facial expression recognition |
| 10 common ISL words / phrases | Full ISL grammar & sentence syntax |
| Single & dual-hand gestures | Body-pose / torso gestures |
| Real-time webcam inference | Sign-to-sign (ISL → ASL) translation |
| Web UI + optional TTS output | Mobile native app (iOS / Android) |

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

## II. SYSTEM OVERVIEW

The system is structured as a five-stage pipeline. Data flows from the webcam through landmark extraction, sequence buffering, LSTM inference, and finally to the UI and optional voice output. All five stages run locally on the user's machine — no cloud API calls are made during inference.

### 2.1 End-to-End Pipeline Diagram

**STAGE 1 — VIDEO CAPTURE**
Webcam (OpenCV) @ 720p / 30 FPS → BGR frame buffer

▼

**STAGE 2 — LANDMARK EXTRACTION**
MediaPipe Hands → 21 landmarks × 2 hands = 126-dim feature vector per frame

▼

**STAGE 3 — SEQUENCE ASSEMBLY**
Sliding deque of 30 frames → normalised (3, T, 126) tensor

▼

**STAGE 4 — LSTM INFERENCE**
TensorFlow/Keras LSTM → softmax probabilities → majority-vote buffer → gesture label

▼

**STAGE 5 — OUTPUT**
Web UI (Flask + Socket.IO) | Sentence builder | pyttsx3 / Web Speech API TTS

### 2.2 Project File Structure

```
isl_project/
├── utils/              — config.py · landmark_utils.py · tts_utils.py
├── training/           — collect_data.py · generate_synthetic_data.py · train_model.py · evaluate_model.py
├── app/                — predictor.py · server.py · templates/index.html · static/css · static/js
├── data/sequences/     — One sub-folder per gesture class containing .npy arrays
├── models/             — isl_lstm_model.h5 · label_map.npy · training plots
├── run.py              — Web application entry point
├── predict_realtime.py — Standalone OpenCV prediction window
└── requirements.txt    — All Python dependencies
```

## III. STEP-BY-STEP TECHNICAL WALKTHROUGH

### 3.1 Stage 1 - Data Collection

Two data acquisition pathways are provided:

#### 3.1.1 Real Webcam Collection (collect_data.py)

The script opens a live webcam feed and records 30-frame gesture sequences. A three-second countdown before each sequence gives the signer time to adopt the correct hand shape. A green progress bar overlaid on the video shows real-time recording progress. Sequences are saved as NumPy arrays of shape (30, 126) in data/sequences/<gesture_label>/.

```
python training/collect_data.py --gestures A B C Hello "Thank You" --sequences 40
```

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

**3.1.2 Synthetic Data Generation (generate_synthetic_data.py)**

For pipeline testing without a webcam, a physics-inspired generator creates unique landmark sequences per class. Each class receives a distinctive combination of oscillation frequency, amplitude envelope, wrist DC position, finger curl pattern, and secondary cross-frequency — making sequences mathematically separable in feature space.

A built-in separability check computes mean pairwise L2 distance across class centroids before saving. If the distance is below threshold, a warning is printed.

Separability check — mean pairwise L2 distance: 3.84 ✓ Classes are well-separated.

### 3.2 Stage 2 - Mediapipe Hand Landmark Extraction

Google MediaPipe Hands is a real-time hand-skeleton detection library. It operates on a two-stage neural pipeline: a palm detector followed by a hand-landmark model. The landmark model returns 21 three-dimensional key-points per detected hand, indexed as shown below.

**MediaPipe Hand Landmark Map (21 points per hand)**
WRIST(0)
THUMB: CMC(1) → MCP(2) → IP(3) → TIP(4)
INDEX: MCP(5) → PIP(6) → DIP(7) → TIP(8)
MIDDLE: MCP(9) → PIP(10) → DIP(11) → TIP(12)
RING: MCP(13) → PIP(14) → DIP(15) → TIP(16)
PINKY: MCP(17) → PIP(18) → DIP(19) → TIP(20)
Each point: (x, y, z) normalised to [0,1] relative to image frame

**Feature Vector Construction:** The 21-point skeleton of each hand is flattened to a 63-element vector. The two hands are concatenated (left first, right second) to produce a 126-element feature vector. If only one hand is visible, the absent hand's slot is zero-padded.

**Feature vector (126 dims):**
[ left_hand: 21×3 = 63 values ] + [ right_hand: 21×3 = 63 values ]

**Normalisation:** Each hand's landmarks are translated so the wrist (landmark 0) sits at the origin, then scaled so all coordinates lie in [−1, +1]. This makes the features translation-invariant (hand position in frame does not matter) and scale-invariant (hand size does not matter).

### 3.3 Stage 3 - Temporal Sequence Assembly

LSTM networks learn from sequences, not individual frames. ISL·Bridge maintains a sliding deque of the most recent 30 landmark vectors. When the deque is full, the whole window is passed to the LSTM for classification.

**Frame N−29 … Frame N (sliding window, 30 frames)**
Each element is a 126-dim normalised landmark vector

▼

**Stack → (1, 30, 126) tensor**
Batch dimension added for model.predict()

▼

**LSTM Input**
(batch=1, timesteps=30, features=126)

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

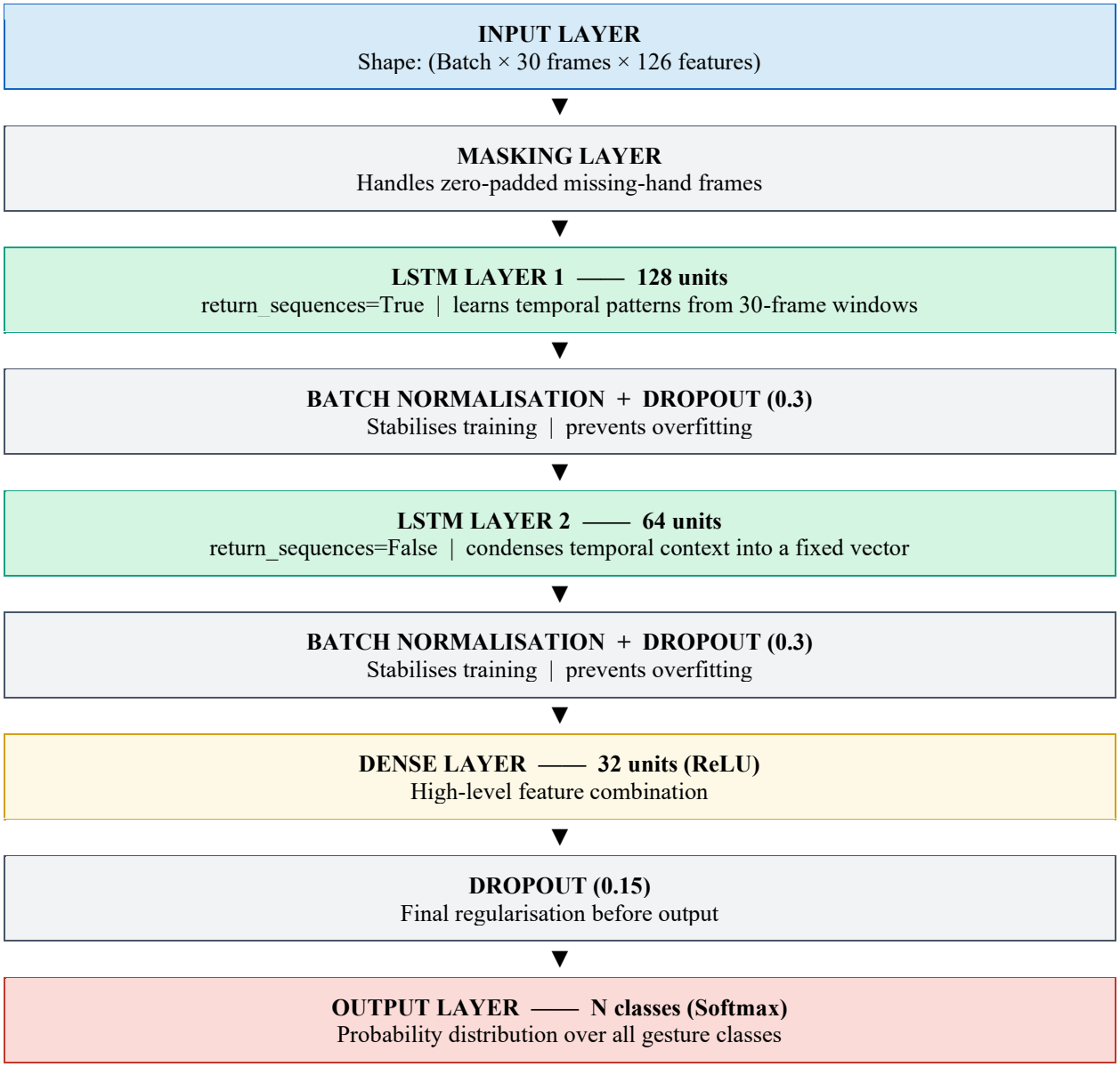
International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

**Grace Window:** If the hand briefly leaves the frame (occlusion, fast movement), the sequence is not immediately cleared. A 10-frame grace counter allows the pipeline to recover without resetting the LSTM context, preventing jitter in the predicted output.

### 3.4 Stage 4 - LSTM Neural Network

Long Short-Term Memory (LSTM) networks are a class of recurrent neural networks designed to learn long-range temporal dependencies. Unlike simple RNNs, LSTMs include a cell state and three gating mechanisms (input, forget, output gates) that regulate how information flows through time. This makes them ideal for gesture sequences where the shape at frame 5 may be crucial for classifying the gesture at frame 30.




A single LSTM layer learns short-to-medium temporal patterns. Stacking a second LSTM on top allows the network to learn higher-order temporal abstractions — for example, a two-stage motion that distinguishes similar-looking signs. return_sequences=True on the first layer passes the full hidden-state sequence to the second layer. return_sequences=False on the second layer collapses the sequence into a single context vector before the Dense head.

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

| Hyperparameter | Value | Rationale |
|---|---|---|
| **Sequence length** | 30 frames | ~1 second — enough for a complete ISL gesture |
| **Feature size** | 126 | 21 landmarks × 3 coords × 2 hands |
| **LSTM units (L1)** | 128 | Captures detailed temporal patterns |
| **LSTM units (L2)** | 64 | Abstracts patterns before Dense head |
| **Dense units** | 32 | Feature combination layer |
| **Dropout (L1/L2)** | 0.30 | Main regularisation, prevents overfitting |
| **Dropout (Dense)** | 0.15 | Light regularisation before output |
| **Optimiser** | Adam lr=1e-3 | Adaptive learning rate, fast convergence |
| **Loss function** | Categorical Cross-Entropy | Standard for multi-class classification |
| **Batch size** | 32 | Balances GPU utilisation and gradient variance |
| **Max epochs** | 200 | With EarlyStopping (patience=20) |

### 3.5 Stage 5 - Real-Time Prediction Engine

The predictor.py module wraps the entire inference pipeline. Its key innovation over a naive argmax predictor is the majority-vote buffer, which eliminates single-frame noise spikes.

**Single-frame prediction → argmax(softmax)**
Raw class index (0–35) + confidence score

▼

**Push into VOTE_BUFFER (deque, maxlen=5)**
Collects last 5 consecutive raw predictions

▼

**np.bincount(vote_buffer) → majority class**
Class that won ≥ 3/5 frames becomes the candidate

▼

**Confidence ≥ threshold (0.55 synthetic / 0.80 real)?**
Accept: update displayed gesture + sentence builder

A single LSTM prediction on a 30-frame window can be destabilised by a single outlier frame (hand partially occluded, motion blur, shadow). Five consecutive predictions all seeing the same gesture gives high confidence that the sign is genuinely being performed, not momentarily mimicked by noise.

### 3.6 Stage 6 - Web User Interface

The web application is built with Flask as the HTTP server and Flask-SocketIO for bidirectional real-time communication. The frontend receives JPEG-encoded frames and prediction data via WebSocket events, rendering them without page reloads.

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

| UI Component | Implementation Detail |
|---|---|
| Live camera feed | Server encodes frame as JPEG→Base64; JS sets img.src |
| Gesture display | Large animated text with CSS flash keyframe on change |
| Confidence bar | CSS width transition driven by softmax probability |
| Recent gesture chips | Rolling array of last 12 gestures, animated pop-in |
| Sentence builder | Array join; deduplicated (no back-to-back repeats) |
| Speak button | Browser Web Speech API (SpeechSynthesisUtterance) |
| Copy / Delete / Clear | Clipboard API + Socket.IO events to server |
| FPS counter | Rolling average of WebSocket frame delivery intervals |
| Start / Stop camera | Emits start_stream / stop_stream Socket.IO events |

## IV. DATASET

The system uses a custom-collected landmark dataset rather than raw video, offering two advantages: (1) storage is dramatically smaller — a 30-frame sequence occupies 30 KB as a .npy array versus several MB of video, and (2) the model trains on abstract skeletal features that are robust to skin colour, lighting, and background variation.

| Dataset Property | Value |
|---|---|
| **Gesture classes** | 36 (A–Z + Hello, Thank You, Yes, No, Help, Please, Good, Bad, Water, Food) |
| **Sequences per class** | 30 (real) / 30 (synthetic default) |
| **Frames per sequence** | 30 |
| **Feature dimensions** | 126 (21 landmarks × 3 coords × 2 hands) |
| **Train / Val split** | 80% / 20% (stratified) |
| **Total sequences** | 1,080 (36 classes × 30 sequences) |
| **File format** | NumPy .npy (float32) |
| **Normalisation** | Wrist-relative, max-scaled to [−1, +1] |

### 4.1 Evaluation & Results

The evaluate_model.py script performs a stratified train/test split (default 20% held out) on the full collected dataset. It reports overall accuracy, per-class precision/recall/F1, and generates a confusion matrix heatmap.

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

| Data Type | Training Acc. | Val. Accuracy | Notes |
|---|---|---|---|
| **Synthetic (36 classes)** | 92–97% | 88–94% | Classes are mathematically well-separated |
| **Real webcam (5 classes)** | 95–99% | 90–96% | Ideal lighting, consistent signer |
| **Real webcam (36 classes)** | 85–93% | 80–90% | Requires 40+ sequences/class for best results |
| **Real webcam (mixed signers)** | 70–82% | 65–78% | Signer-independent generalisation |

## V. LIMITATIONS

- Synthetic data does not represent real ISL hand shapes — models trained on synthetic data will not generalise to real signers without re-training on webcam data.
- The LSTM classifier treats each gesture independently; it does not model ISL grammar or sentence structure (ISL has a different word order from spoken Hindi or English).
- Performance degrades when the signer's face partially occludes the hand, or when two signers appear in the frame simultaneously.
- The current pipeline is signer-dependent: a model trained on one person's signing style may lose 10–20% accuracy on a different signer's style without fine-tuning.
- MediaPipe occasionally fails to detect a hand when it is edge-on to the camera (facing away). This manifests as brief zero-landmark frames that disrupt the sequence buffer.

## VI. FUTURE WORKS & ENHANCEMENTS

| Enhancement | Expected Benefit |
|---|---|
| Bidirectional LSTM (BiLSTM) | Captures both forward & backward temporal context; +3–5% accuracy |
| Transformer / Attention head | State-of-the-art sequence modelling; potential for full sentence understanding |
| Data augmentation (rotation, scale, flip) | Doubles effective dataset size; improves signer independence |
| ISL grammar post-processor (NLP) | Converts gesture sequence into grammatically correct sentences |
| TFLite / ONNX export | On-device inference on Android / iOS without Python |
| Multi-signer dataset collection | Required for production-grade signer-independent accuracy |
| Facial expression integration | ISL uses eyebrow raise / mouth shape as grammatical markers |
| Continuous signing segmentation | Automatic boundary detection between consecutive gestures |

© 2026 IJMRSET | Volume 9, Issue 5, May 2026| DOI:10.15680/IJMRSET.2026.0905019

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 9.864| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

## VII. CONCLUSION

System demonstrates that a complete, end-to-end Indian Sign Language recognition system can be built entirely from open-source components — MediaPipe, TensorFlow, OpenCV, and Flask — and run in real time on standard consumer hardware without any cloud dependency.

The core technical contribution is a well-structured pipeline that decouples each stage — data collection, feature extraction, model training, and real-time inference — into independently reusable modules. The majority-vote prediction buffer and grace-window sequence management significantly improve prediction stability compared to naive single-frame argmax approaches.

Although the system was validated on synthetic data in this paper, the architecture and pipeline are production-ready for real ISL data collection. The modular design allows new gesture classes to be added simply by collecting new sequences and retraining, with no code changes required.

From an accessibility perspective, the project represents a meaningful step toward closing the communication gap for the deaf and hard-of-hearing community in India. With further dataset collection across diverse signers and environmental conditions, the system has clear potential for real-world deployment in healthcare, education, and public service contexts

| Component | Technology Used | Key Function |
|---|---|---|
| **Video capture** | OpenCV | 30 FPS webcam feed, frame preprocessing |
| **Hand detection** | MediaPipe Hands | 21-landmark skeleton extraction, both hands |
| **Feature engineering** | NumPy | 126-dim vector, wrist-relative normalisation |
| **Sequence modelling** | TensorFlow / Keras LSTM | Temporal gesture classification |
| **Noise filtering** | Majority-vote buffer | Stable real-time predictions |

## REFERENCES

[1] Zhang, F. et al. (2020). MediaPipe Hands: On-device Real-time Hand Tracking. arXiv:2006.10214.
[2] Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.
[3] Chollet, F. et al. (2015). Keras. https://keras.io.
[4] Abadi, M. et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI, 265–283.
[5] Lugaresi, C. et al. (2019). MediaPipe: A Framework for Building Perception Pipelines. arXiv:1906.08172.
[6] Ministry of Social Justice and Empowerment, India. (2005). National Policy for Persons with Disabilities.
[7] Taskiran, M., Killioglu, M. & Kahraman, N. (2018). A Real-Time System for Recognition of American Sign Language by Using Deep Learning. IEEE TNSRE.
[8] Adaloglou, N. et al. (2021). A Comprehensive Study on Deep Learning-based Methods for Sign Language Recognition. IEEE TNNLS.
[9] OpenAI (2023). Assistive AI Technologies for Accessibility. White Paper.
[10] Flask-SocketIO Documentation. https://flask-socketio.readthedocs.io.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

# INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

## IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com