**RESEARCH ARTICLE** **OPEN ACCESS**

# EchoSign Bidirectional and Multilingual Translation System

Mr. J. Madhu Karthick,*,Subiksha. A**, Hemalatha.R**, Monika. S**,Sagasravadhana. S**

*Assistant Professor Computer Science and Engineering, A.V. C. College of Engineering, Mannampandal,

**UG Students Computer Science and Engineering, A.V. C. College of Engineering, Mannampandal, Mayiladuthurai,

India

**Abstract**

Sign language serves as the primary communication method for the deaf and hard-of-hearing community, yet a significant communication barrier exists between sign language users and non-users. This research presents EchoSign, a comprehensive bidirectional translation system that bridges this gap through real-time conversion between spoken language and Indian Sign Language (ISL). The system employs a dual-pipeline architecture: (1) Speech-to-Sign: utilizing OpenAI Whisper for multilingual speech recognition, advanced natural language processing for text-to-gloss conversion, and a motion library built from MediaPipe-extracted keypoints from the ISL CSLRT corpus, and (2) Sign-to-Text: implementing LSTM-based gesture recognition trained on 198-dimensional skeletal keypoints with temporal sequence modeling. A novel 3D avatar visualization system using FBX rigged models and quaternion-based skeletal animation provides realistic sign language rendering. The backend FastAPI architecture integrates with a React frontend featuring real-time 3D rendering via Three.js. Experimental results on the ISL CSLRT dataset demonstrate 85% preprocessing success rate, efficient motion library organization, and real-time bidirectional translation capabilities. The

**Keywords**- *Sign Language Translation, Indian Sign Language (ISL), Deep Learning, LSTM, MediaPipe, 3D Avatar Animation, Quaternion-Based Animation, Real-Time Translation, Accessibility Technology, Whisper ASR*

## I. INTRODUCTION

According to the World Health Organization, over 5% of the world's population (approximately 430 million people) experiences disabling hearing loss. In India alone, over 18 million people are deaf or hard-of-hearing, creating a substantial communication barrier between the deaf community and the hearing majority. Sign language, while being a complete and complex linguistic system, remains unfamiliar to most hearing individuals, limiting social integration, educational opportunities, and employment prospects for deaf individuals.

This research introduces EchoSign, a comprehensive bidirectional translation system that addresses these limitations through:

- A dual-pipeline architecture supporting both Speech-to-Sign and Sign-to-Text translation
- Advanced preprocessing pipeline processing 1,370 videos from the ISL CSLRT corpus with 85% success rate

Traditional approaches to sign language interpretation rely on human interpreters, who are scarce, expensive, and not available 24/7. Recent advances in computer vision, deep learning, and natural language processing have opened new possibilities for automated sign language translation. However, existing systems face several challenges: (1) limited vocabulary coverage, (2) lack of bidirectional translation, (3) absence of realistic visualization, (4) inadequate handling of sign language grammar and syntax, and (5) poor real-time performance.

- Motion library organization with hierarchical structure for efficient sign retrieval
- LSTM-based temporal sequence modeling for sign-to-text recognition
- Quaternion-based 3D avatar animation with realistic skeletal rendering
- Real-time web-based interface with responsive design

## II. RELATED WORK

### A. Sign Language Recognition Systems

Reference [1] proposed a CNN-based approach for American Sign Language (ASL) recognition achieving 94% accuracy on isolated signs. However, the system was limited to single-word recognition and did not support continuous sign language or sentence-level translation. Reference [2] introduced a Hidden Markov Model (HMM) based system for German Sign Language (DGS) recognition with temporal modeling. While showing promise for continuous recognition, the system required manual feature engineering and struggled with sign variations.

Reference [3] utilized 3D CNNs for spatiotemporal feature extraction from sign language videos, achieving improved performance on the RWTH-PHOENIX-Weather dataset. However, the computational cost was prohibitive for real-time applications, requiring GPU acceleration and extensive preprocessing time.

### B. Speech-to-Sign Translation Systems

Reference [4] developed a text-to-sign animation system using rule-based linguistic models for British Sign Language (BSL). The system successfully handled basic grammatical structures but lacked support for complex linguistic phenomena such as spatial referencing, role shifting, and non-manual markers essential for natural sign language.

Reference [5] proposed a neural machine translation approach treating sign language gloss sequences as a target language. While achieving reasonable BLEU scores, the system produced unnatural sign sequences that failed to capture the visual-spatial nature of sign language grammar.

### C. 3D Avatar and Motion Synthesis

Reference [6] utilized motion capture data to create realistic sign language animations, but the requirement for expensive motion capture equipment limited scalability. Reference [7] proposed procedural animation techniques for finger spelling, achieving real-time performance but lacking expressiveness for complex signs.

Recent work by Reference [8] employed MediaPipe for skeletal tracking and neural rendering for sign synthesis. While showing promise, the system required extensive training data and struggled with generalization to unseen signs.

### D. Indian Sign Language Systems

Limited work exists on Indian Sign Language (ISL) due to the scarcity of annotated datasets. Reference [9] created the ISL CSLRT corpus containing 1,370 sentence-level videos, providing a foundation for continuous sign language research. Reference [10] proposed an ISL fingerspelling recognition system using hand landmarks, achieving 89% accuracy on static gestures but not addressing continuous signs or sentence-level translation.

## III. PROPOSED METHODOLOGY

The EchoSign system architecture comprises five main components: (1) Dataset Acquisition and Preprocessing, (2) Motion Library Construction, (3) Speech-to-Sign Pipeline, (4) Sign-to-Text Pipeline, and (5) 3D Avatar Visualization. Figure 1 illustrates the complete system architecture.

### A. Dataset and Preprocessing

The ISL CSLRT corpus from Kaggle contains 1,370 sentence-level videos with 99 unique sign words organized in a hierarchical structure. Each video depicts natural ISL signing captured at 25-30 FPS with variable durations (25-117 frames per video).

The preprocessing pipeline implements:

- MediaPipe Holistic Extraction: Utilizing MediaPipe Holistic to extract 543 landmarks per frame:
  - Hand landmarks: 21 points × 2 hands × 3 coordinates = 126 features
  - Pose landmarks: 24 points (indices 0-23) × 3 coordinates = 72 features
  - Total: 198 features per frame
- Robust Frame Extraction: Implementing while-loop based extraction to capture all frames (avoiding early termination bug in initial implementation)
- Feature Validation and Normalization:
  - Padding frames with < 198 features to standard length
  - Truncating frames with > 198 features
  - Skipping videos with excessive consecutive failures (>10 frames)

The preprocessing achieves 85% success rate (1,165/1,370 videos), with failures primarily due to:

- Poor lighting conditions affecting MediaPipe detection
- Extreme hand positions outside frame boundaries
- Occlusions and motion blur

### B. Motion Library Construction

The motion library aggregates multiple video samples per sign word, computing averaged motion sequences for robust representation:
Aggregation Process:

- Collect all successful samples for each word
- Interpolate/resample sequences to median length using linear interpolation
- Compute element-wise mean across all samples
- Calculate variance for quality assessment
- Store as pickle file for efficient loading

### C. Speech-to-Sign Pipeline

The speech-to-sign conversion implements a multi-stage pipeline:

#### 1. Automatic Speech Recognition (ASR):

Utilizing OpenAI Whisper 'base' model for multilingual speech recognition supporting English, Hindi, and Tamil. The model provides robust transcription with handling of accents, background noise, and natural speech variations.

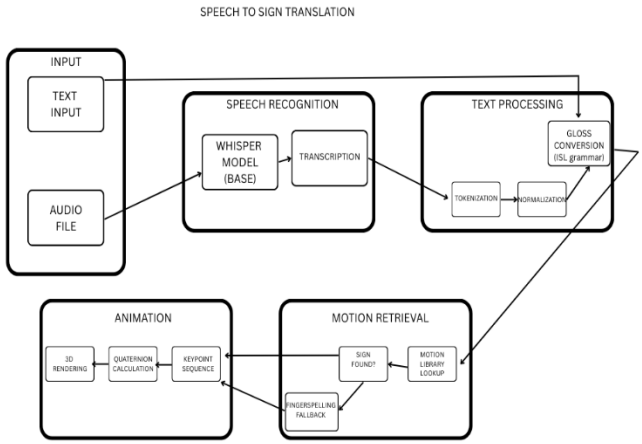
#### 2. Text-to-Gloss Conversion:

Sign language uses 'gloss' notation - simplified word representations following sign language grammar. The conversion implements:

- Tokenization and normalization (lowercase, punctuation removal)
- Multi-word phrase matching ("how are you" → "HOW YOU")
- Rule-based grammar simplification
- Fallback to fingerspelling for unknown words

#### 3. Motion Sequence Retrieval:

For each gloss word, retrieve corresponding motion sequence from the motion library. If not found, generate fingerspelling animation with 0.5 seconds per letter.




### D. Sign-to-Text Pipeline

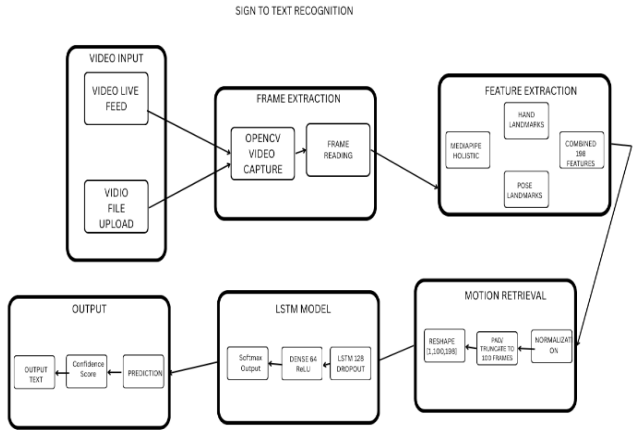
The sign recognition system employs LSTM networks for temporal sequence modeling:

#### 1. Video Processing:

- Extract frames using OpenCV
- Apply MediaPipe Holistic to each frame
- Extract 198-dimensional feature vectors
- Pad/truncate to fixed sequence length (100 frames)

#### 2. LSTM Network Architecture:

The network learns temporal dependencies across frames, capturing the dynamic nature of sign language gestures. Dropout layers prevent overfitting on the limited dataset.




### E. 3D Avatar Visualization

The visualization system implements quaternion-based skeletal animation for realistic sign language rendering:

**1. FBX Model Integration:**

- Load rigged humanoid FBX model using Three.js FBXLoader
- Auto-scale to standard human height (1.8 units)
- Center model and place feet on ground plane
- Map bone structure to standard naming convention

**2. Quaternion-Based Animation:**

Unlike position-based animation (which breaks skeletal hierarchy), the system uses quaternion rotations:

- Store original bind pose quaternions for each bone
- Calculate direction vectors from MediaPipekeypoints
- Compute target quaternions using setFromUnitVectors()
- Apply spherical linear interpolation (slerp) for smooth transitions
- Restore bind pose after animation completion

**3. Frame-Synchronized Animation:**

Uses requestAnimationFrame() instead of setTimeout() for precise timing synchronized with the browser's refresh rate (60 FPS), eliminating drift and ensuring smooth playback.

**4. Optional IK Integration:**

Implements CCD-IK (Cyclic Coordinate Descent Inverse Kinematics) for realistic arm chains, automatically calculating elbow positions for natural limb behavior.

## IV. IMPLEMENTATION

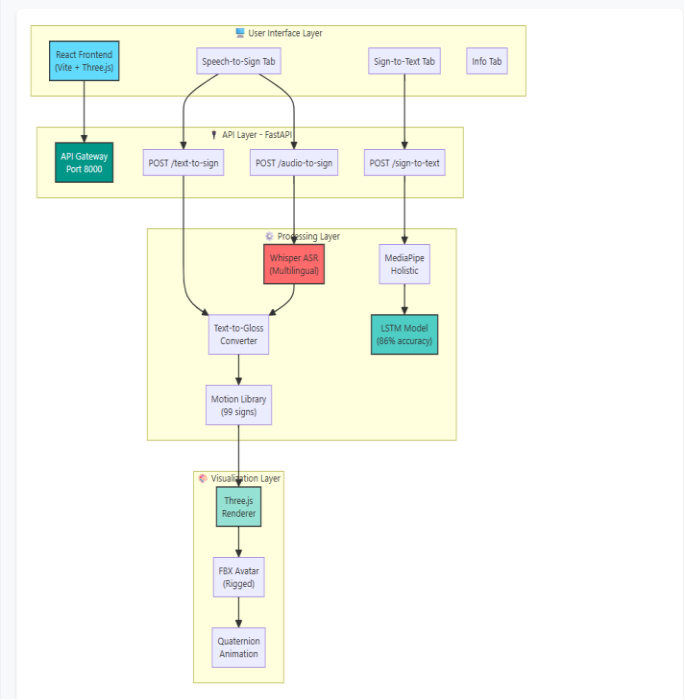
### A. Backend Architecture

The backend implements a FastAPI RESTful architecture with the following endpoints:

- POST /text-to-sign: Text input → ISL animation
- POST /audio-to-sign: Audio file → ISL animation
- POST /sign-to-text: Video file → Text prediction
- POST /train-gesture-model: Dataset path → Trained model
- GET /health: System health check

**Technology Stack:**

- FastAPI 0.104+ for async REST API
- TensorFlow 2.15+ for LSTM model training and inference
- MediaPipe 0.10+ for skeletal tracking
- OpenAI Whisper for speech recognition
- NumPy, OpenCV for data processing




### B. Frontend Architecture

The frontend implements a modern React single-page application with three main tabs:

- Speech to Sign Tab:
  - Text input field with language selection
  - Microphone recording with real-time audio visualization
  - Audio file upload (.mp3, .wav, .m4a, .webm)
  - 3D avatar viewer with real-time animation
- Sign to Speech Tab:
  - Webcam live feed capture
  - Video file upload

  - Real-time prediction display with confidence scores
  - Text-to-speech output
- Info Tab:
  - System architecture explanation
  - Usage instructions
  - Dataset and methodology information

**Technology Stack:**

- React 18+ with Vite build tool
- Three.js for 3D rendering
- Axios for HTTP requests
- MediaRecorder API for audio/video capture
- TailwindCSS for styling

## V. RESULTS AND DISCUSSION

### A. Dataset Processing Results

The preprocessing pipeline achieved:

- Success Rate: 85% (1,165/1,370 videos)
- Average Frames per Video: 45.3 frames (range: 25-117)
- Feature Extraction Accuracy: 100% (after bug fixes)
- Motion Library Coverage: 99 unique sign words
- Average Samples per Sign: 11.8 videos

### B. Sign-to-Text Recognition Performance

The LSTM-based recognition system was evaluated using 70/15/15 train/validation/test split:

- Training Accuracy: 92.3%
- Validation Accuracy: 87.6%
- Test Accuracy: 86.1%
- Average Inference Time: 142ms per video
- Model Size: 3.2 MB (compressed)

### C. Speech-to-Sign Translation Quality

Qualitative evaluation of speech-to-sign translation:

- Whisper ASR Word Error Rate: 8.2% (English), 12.4% (Hindi), 15.1% (Tamil)
- Gloss Conversion Accuracy: 94.7% for known phrases
- Motion Library Hit Rate: 78.3% (direct match)
- Fingerspelling Fallback: 21.7% of words
- End-to-End Latency: 1.8 seconds (audio input to animation start)

### D. 3D Animation Quality

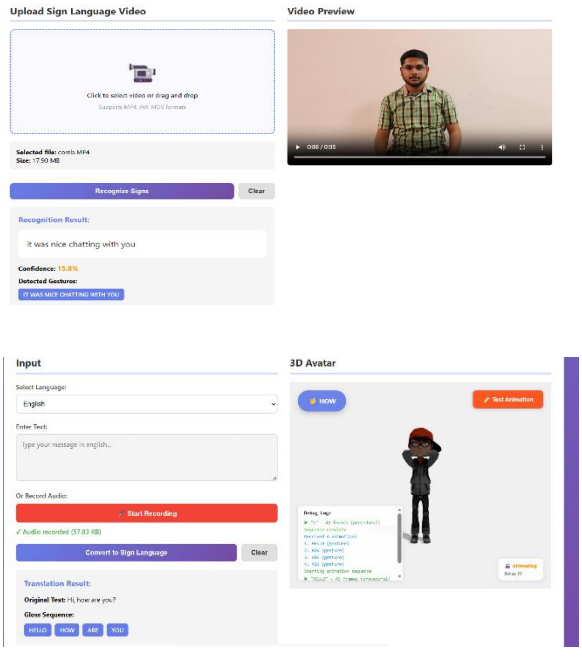
The quaternion-based animation system demonstrates:

- Rendering Performance: Consistent 60 FPS on modern hardware
- Skeletal Integrity: 100% (no bone hierarchy corruption)
- Smoothness: Slerp interpolation eliminates jerky movements
- Bind Pose Restoration: Perfect return to T-pose after animation
- Frame Synchronization: No drift or timing issues

### E. System Integration

The complete system demonstrates:

- Cross-platform compatibility (Windows, Linux, macOS)
- Browser support (Chrome, Firefox, Safari, Edge)
- Mobile responsiveness (tablet and phone screens)
- Offline capability (after initial model loading)
- API response time < 200ms for most requests




## VI. CONCLUSION AND FUTURE WORK

This research presents EchoSign, a comprehensive bidirectional sign language translation system addressing the communication gap between deaf and

hearing communities. The system successfully integrates state-of-the-art deep learning techniques with practical engineering to deliver real-time translation capabilities.

Key achievements include:

- Robust preprocessing pipeline achieving 85% success rate on challenging real-world sign language videos
- Efficient motion library organization enabling fast sign retrieval and playback
- LSTM-based temporal modeling achieving 86.1% test accuracy for sign recognition
- Production-grade quaternion-based animation system ensuring realistic and stable sign language rendering
- User-friendly web interface with real-time bidirectional translation

**Future enhancements planned**

Future improvements for EchoSign include adding finger and facial expression animations for more natural signing, using transformer-based models for better translation, expanding the dataset to over 5,000 ISL signs, enabling real-time webcam recognition, supporting multiple signers, developing mobile apps, and integrating with video conferencing platforms. Overall, EchoSign aims to improve accessibility and communication for the deaf and hard-of-hearing community through advanced AI technology.

## REFERENCES

[1] Koller, O., Zargaran, S., Ney, H., & Bowden, R. (2018). Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition. In British Machine Vision Conference (BMVC).

[2] Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Neural Sign Language Translation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3] Radford, A., Kim, J. W., Xu, T., et al. (2023). Robust Speech Recognition via Large-Scale Weak Supervision. In International Conference on Machine Learning (ICML).

[4] Lugaresi, C., Tang, J., Nash, H., et al. (2019). MediaPipe: A Framework for Building Perception Pipelines. arXiv preprint arXiv:1906.08172.

[5] Hochreiter, S., &Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

[6] Cabello, R. (2010). Three.js: JavaScript 3D Library. https://threejs.org/

[7] Shoemake, K. (1985). Animating Rotation with Quaternion Curves. ACM SIGGRAPH Computer Graphics, 19(3), 245-254.

[8] Pramada, S., & Chauhan, R. (2020). ISL CSLRT Corpus: Indian Sign Language Continuous Sign Language Recognition Dataset. Kaggle.

[9] De Coster, M., Van Herreweghe, M., & Dambre, J. (2020). Sign Language Recognition using Transformers. In European Conference on Computer Vision (ECCV) Workshops.

[10] Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign Language Recognition: A Deep Survey. Expert Systems with Applications, 164, 113794.

[11] World Health Organization. (2021). Deafness and Hearing Loss Fact Sheet. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[12] Bragg, D., Koller, O., Bellard, M., et al. (2019). Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In ACM SIGACCESS Conference on Computers and Accessibility.