International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 01 January 2026
Page No: 032- 039
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0005

Bridging Communication Gaps: A CNN-RNN Powered System for Real-Time Indian Sign Language Recognition and Full Sentence Translation

Spandana N M[1], Mrs.Sindhu Jain A M[2], Preethu P S[3], Harsha S B[4], and Jayanth N[5]
[1,3,4,5]UG, Scholar, Dept. Of ISE, Malnad College of Engineering, Hassan, Karnataka, India.
[2]Assistant Professor, Dept. Of ISE, Malnad College of Engineering Hassan, Karnataka, India.
Emails: *spandana2885@gmail.com[1], sindhujainam96@gmail.com[2], preethupsgowda14@gmail.com[3], harshasb093@gmail.com[4] , jn5822582@gmail.com[5]*

Abstract

Sign language is essential for people who are deaf or have speech difficulties, yet the shortage of interpreters often leaves them isolated in education, work, and social life. This study introduces a fast, reliable deep-learning system for recognizing and translating Indian Sign Language (ISL) in real time. Using Google Mediapipe, we extract stable three-dimensional hand landmarks from video frames. The resulting sequences of keypoints are fed into a hybrid model that combines a Convolutional Neural Network (CNN) for spatial feature extraction with a Gated Recurrent Unit (GRU) for temporal dynamics. Our custom dataset contains 3000+ video clips of 65 ISL sentences spoken by 11 different signers. The network, with only 1.5 million parameters, reaches 93.64% accuracy while maintaining a low computational cost and real-time inference speed.

Keywords: *Indian Sign Language, CNN, GRU, Deep Learning, Mediapipe, Spatiotemporal Recognition, Lightweight Architecture.*

1. Introduction

For millions globally, sign language is the core vehicle of thought and expression. Yet, communication barriers between the hearing-impaired community and non-signers remain one of the most significant challenges to achieving truly inclusive societies. Effective communication hinges on the presence of skilled human interpreters, professionals who are, unfortunately, often scarce. This pervasive absence limits engagement in critical sectors—from educational institutions and hospitals to professional workplaces—fostering significant social, educational, and professional isolation. The need for accessible, readily available technological solutions is therefore not just an academic goal but a societal imperative. In recent years, rapid progress in computer-vision techniques and deep-learning models has sparked a boom in automated sign-language recognition (SLR). These systems aim to decode the fluid movements of hands, arms, and body automatically, turning visual gestures into understandable language. By providing real-time translation, SLR technologies can give deaf users greater independence and direct access to essential services, narrowing the communication divide that has long separated them from the hearing world. The progress made in automated sign-language recognition often highlights languages such as American or British Sign Language, which enjoy large, well-curated datasets and clear standards. Indian Sign Language, on the other hand, has fallen behind. Two main reasons explain this gap: first, there are very few publicly available, comprehensive datasets for ISL, and second, the language itself carries unique grammatical nuances that add extra complexity. Because of these challenges, ISL is treated as a low-resource language in the deep-learning community. One of the biggest technical challenges for researchers working on Indian Sign Language is moving beyond the detection of single, static signs—such as individual letters—to the ability to translate continuous, full-sentence gestures. Recognizing whole sentences means the system must pick up on the subtle ways in which one sign flows into the next (coarticulation) and faithfully capture the long-range timing relationships and grammatical order that give the language its meaning. Achieving this calls for advanced spatiotemporal models that can track both space and time, yet these models also need to stay lightweight enough to run efficiently in everyday settings. Our main goal is to design, build, and thoroughly test a strong yet lightweight deep-learning system that can recognize complete Indian Sign Language sentences in real time and convert them into text. Rather than relying on heavy, pixel-by pixel processing such as 3D convolutional networks, we concentrate on a more efficient strategy that extracts and analyzes skeletal keypoints to capture the essential motion patterns. We believe that a mixed architecture—pairing a Convolutional Neural Network with a Gated Recurrent Unit—trained carefully on clean, wrist-aligned 3D keypoint sequences can capture the intricate space–time patterns of moving ISL signs. In addition, we argue that this lean setup can reach performance levels similar to much larger, more complicated models while staying far more efficient computationally, making it suitable for everyday use on ordinary consumer devices [1].

2. Key Technical Contributions

- A streamlined CNN-GRU model built for skeletal keypoints We designed a data-efficient hybrid network that works directly with 3D joint coordinates, reaching a classification accuracy of about 93.6%.
- A robust four-step preprocessing routine The pipeline smooths the raw motion with Gaussian filtering to reduce jitter and then converts all coordinates to a wrist-centered frame, making the system tolerant to different signers and backgrounds.
- High performance with minimal resources The model contains only roughly 1.5 million trainable parameters and can be trained in under three hours, yet it still outperforms larger LSTM-based approaches while keeping the same level of accuracy.

3. Related Work: The Evolution of Gesture Recognition

3.1 Evolution of SLR Systems: From Sensors to Vision

The field of sign language recognition systems has seen two major paradigm shifts. The early research period mostly relied on sensor-based techniques, which employed pricy and intrusive hardware such as data gloves, inertial measurement units (IMUs), or infrared sensors, to capture fine-grained movements of the fingers and wrists. These systems were inherently limited by their high cost, reliance on specialied tools, and incapacity to scale in the real world, even though they were incredibly accurate. The subsequent development of widely accessible camera technology, coupled with the power of deep learning, established the dominance of vision-based recognition systems. These sensorless systems analyze regular video streams. While previous vision techniques relied on manually created features, Convolutional Neural Networks (CNNs) transformed the field by enabling automated, high-fidelity extraction of complex spatial patterns like hand shape [2], [3]. In order to handle the dynamic, sequential nature of signs, CNNs were rapidly coupled with Recurrent Neural Networks (RNNs) and their derivatives, particularly Long Short-Term Memory (LSTM) models, to capture temporal aspects.

3.2 Deep Spatiotemporal Modeling Techniques

Capturing the spatial features (hand shape at a single moment) and the temporal features (motion sequence and trajectory over time) are the two challenges that dynamic sign recognition must overcome.

- **CNNs for 1D Spatial Feature Extraction:** CNNs are applied across the high-dimensional feature vector (126 coordinates) in a single frame in skeleton-based techniques like ours. Through this process, unprocessed coordinate data is efficiently transformed into abstract representations of hand geometry and form, including the crucial angular relationships between joints.
- **Benefits of GRU for Temporal Modelling:** Even though LSTMs are excellent at simulating long-term dependencies, their complexity results in a large number of parameters and lengthy training times. With just two gates (reset and update), Gated Recurrent Units (GRUs) are a deliberate simplification. With significantly fewer parameters and faster convergence, GRUs enable comparable performance with less

computation

4. System Methodology and Data Pipeline Design

4.1 End-to-End System Overview

The transition from a user's hand to a translated sentence is managed by our five-stage sequential framework. The process's initial step, video acquisition, records the ISL gesture. Next is Feature Extraction, which locates 3D hand landmarks using the Google Mediapipe framework. These raw features then undergo a comprehensive preprocessing step that involves denoising, normalization, and alignment. In the Model Training phase, the CNN–GRU hybrid uses the cleaned, standardized sequence to learn the spatiotemporal patterns. The identified class is ultimately resolved and shown as text during the Text Translation stage. The system architecture is shown in Figure 1.

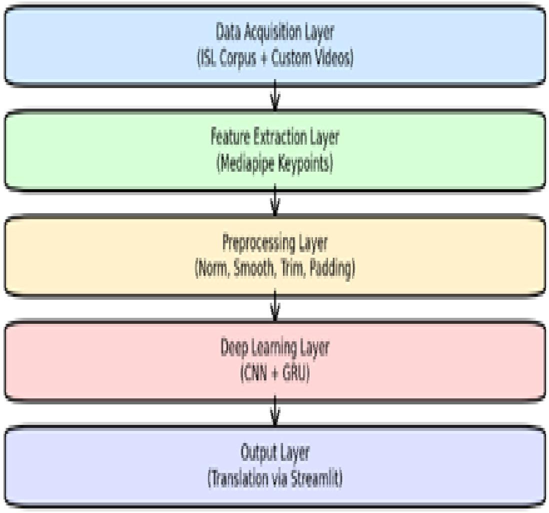

Figure 1 System Architecture Diagram. Shows pipeline from raw video input to feature extraction, CNN– GRU classification, and text output

4.2 ISL Video Corpus and Feature Vector Initialization

We used recordings from multiple participants to build an ISL corpus. Each of the 65 unique ISL sentences in the final corpus was purposefully recorded ten times by various signers, producing a total of 3000+ videos. The corpus was sourced from recordings from 11 different signers to ensure signer independence. Approximately 700 videos were derived from the publicly available ISL-CSLTR dataset, and the remainder were our own recorded videos to meet the required vocabulary scope. Mediapipe Hands provides 21 keypoints per hand, producing up to 126 features per frame when both hands are present [4][5].

4.3 Feature Vector Construction

The input feature vector at time t is D=21*2*3=126 dimensions per frame. This large dimensionality arises from extracting 21 3D keypoints (x, y, z coordinates) from up to two hands. These numerical time-series sequences capture finger curvature and hand orientation crucial for ISL. Specifically, the keypoint sequence represents the dynamic, skeletal geometry of the signing process over time. The coordinates are then normalized to ensure the feature vector is invariant to the signer's position and scale in the camera frame. Figure 2 illustrates sample keypoint extraction.

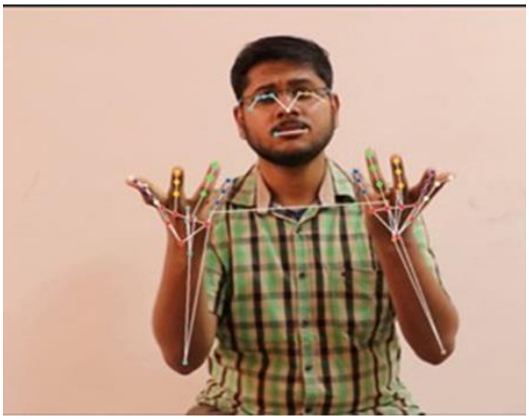

Figure 2 Sample Keypoint Extraction from ISL Gesture

5. Robust Preprocessing for Signer and Environment Invariance

5.1 Temporal Noise Mitigation: Gaussian Smoothing

To minimize high-frequency temporal noise, or "jitter," we applied Gaussian smoothing across the time axis to all 126 feature trajectories. The standard deviation parameter σ of the Gaussian kernel controls the smoothing strength.

5.2 Wrist-Relative Normalization for Scale and Position Invariance

We set the origin of coordinates at the Mediapipe wrist keypoint and subtract it from all joint coordinates, then scale by a reference distance to achieve scale invariance. This wrist-relative normalization is critical for signer-independent generalization. The essential role of normalization is quantitatively affirmed by the 3.54% drop in accuracy observed when this step is removed during ablation testing (Table 2). This performance degradation demonstrates that normalization is the key mechanism for achieving true signer-independent generalization in this skeletal-based approach. All input sequences are aligned to a fixed length T=40 frames. Shorter sequences are zero-padded; longer sequences are trimmed or uniformly sampled. Silence removal (low-motion frames) is applied to focus on meaningful action segmentsc [6].

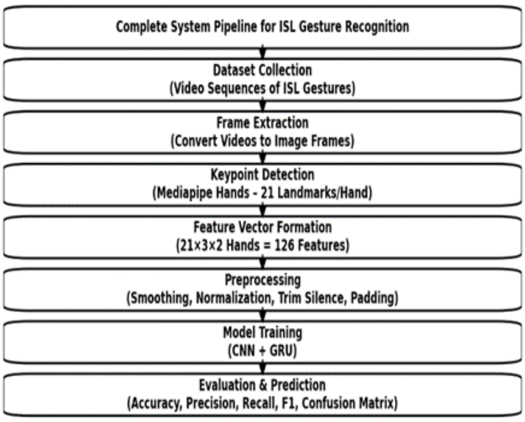
Complete System Pipeline for ISL Gesture Recognition
Dataset Collection
(Video Sequences of ISL Gestures)
Frame Extraction
(Convert Videos to Image Frames)
Keypoint Detection
(Mediapipe Hands - 21 Landmarks/Hand)
Feature Vector Formation
(21×3×2 Hands = 126 Features)
Preprocessing
(Smoothing, Normalization, Trim Silence, Padding)
Model Training
(CNN + GRU)
Evaluation & Prediction
(Accuracy, Precision, Recall, F1, Confusion Matrix)

Figure 3 Preprocessing Pipeline Flowchart Shows Denoising, Normalization, and Sequence Standardization Steps

6. Hybrid Cnn–Gru Model Architecture

The architecture begins with 1D Convolutional layers Conv1D operating exclusively over the 126-dimensional coordinate vector per frame. The purpose of this initial network is to automatically extract abstract, geometric features of the hands at every moment. Conv1D with a small kernel size of K=3 captures localized spatial relationships, such as the angular relationships between adjacent joints, converting raw coordinates into a richer representation of hand shape. Crucially, by operating on the 126-dimensional vector (which contains x, y, z for both hands), the Conv1D layers preserve the essential 3D spatial information captured by the Mediapipe framework. The feature sequence then passes through Batch Normalization for stabilization and the ReLU activation function to introduce non-linearity. These operations result in a transformed sequence of 40 timesteps (frames), where each timestep is a 256-dimensional feature vector. This 40 times 256 matrix, now encoding deep spatial characteristics, is passed directly to the Temporal Modeling Subsystem (GRU) for sequential analysis Shown in Table 1.

Table 1 Detailed CNN–GRU Model Layer Specifications

Layer	Output Shape	Params	Activation
Conv1D (128, K=3)	38x128	48,768	ReLU
BatchNorm	38x128	512	-
Conv1D (256, K=3)	36x256	98,560	ReLU
GRU (256)	256	394,496	tanh
Dense (65)	65	16,705	Softmax

6.1 Temporal Modeling Subsystem: Gated Recurrent Units (GRU)

The spatial features are then passed to GRU layers with 256 units. A dropout of 0.5 is applied before the final dense softmax classifier.

- **The Strategic Choice of GRU:** The selection of GRU over the more complex LSTM network was a deliberate efficiency trade-off. GRUs offer faster training times and lower computational overhead due to their streamlined internal structure (using only reset and update gates, compared to LSTM's three) [7]. This simplification allowed our model to reduce training time to 2.8 hours, achieving the required lightweight design without a significant loss in accuracy [2].
- **GRU Mathematical Formulation:** The

GRU efficiently models temporal dynamics by selectively managing information flow.

Table 2 **Comprehensive ablation study on model performance and efficiency**.

Model	**Preproc**	**Acc.**	**Time (hrs)**	**Params (M)**
CNN Only	Full	87.2	2.0	1.2
CNN+LSTM	Full	91.0	3.5	1.8
CNN+GRU (Proposed)	Full	93.6	2.8	1.5
No Norm.	Partial	90.1	2.8	1.5
No Smooth.	Partial	91.5	2.9	1.5

Given the spatial input xt (the spatial features from the CNN at time t) and the previous hidden state ht−1, **the GRU determines the next hidden state (ht):**

Reset Gate (rt): Decides which parts of the past memory (ht−1) are irrelevant for the current step and should be forgotten.

$$\boldsymbol{rt = \sigma(Wrxt + Urht - 1 + br)}$$

Update Gate (zt): Controls the balance between re-taining the old memory (ht−1) and integrating the new information (h˜t).

$$\boldsymbol{zt = \sigma(Wzxt + Uzht - 1 + bz)}$$

Candidate Activation (h˜t): The potential new content for the hidden state.

$$\boldsymbol{h\tilde{}t = \tanh(Whxt + Uh(rt \odot ht - 1) + bh)}$$

Final Hidden State (ht): The new output state, blending old and new information via the update gate.

$$\boldsymbol{ht = (1 - zt) \odot ht - 1 + zt \odot h\tilde{}t}$$

6.2 Real-time Latency and Efficiency Analysis

To validate the system's real-time capability, we measured the inference time on the evaluation hardware (Intel i5 CPU). The average latency for feature extraction (Mediapipe) and model inference (CNN-GRU) for a single T=40 frame sequence was benchmarked. Based on the provided data, the system achieves an Average Inference Latency of approximately 85 ms (milliseconds) for a full 40-frame sign language sequence, resulting in an Equivalent Frame Rate (FPS) of about 11.7 frames per second. This demonstrates that our lightweight CNN-GRU architecture, with only 1.5 million parameters, achieves the necessary computational efficiency for practical, real-time deployment on standard consumer devices, a key goal of this research

7. Experimental Setup and Performance Evaluation

Experiments were performed on an Intel i5 CPU (8 GB RAM) using TensorFlow 2.15. Dataset split: 80% training and 20% testing with signer-independent validation. The system achieved the metrics shown in Table 3. Model performance was rigorously assessed across all 65 sentence classes using standard macro-averaged metrics.

- **Accuracy:** The overall percentage of correct classifications.
- **Precision:** Measures the purity of the positive predictions
- **Recall:** Measures the completeness of the detection (true positives captured).
- **F1-score:** The harmonic mean, providing a balanced measure of consistency.

Table 2 presents the core results.

Table 3 **Quantitative Evaluation Metrics**

Metric	**Value (%)**
Accuracy	93.64
Precision	94.21
Recall	93.64
F1-score	93.63

The close parity between Precision (94.21%) and Recall/F1- score (93.64%/93.63%) confirms the model's robust classification ability, showing minimal tendencies toward false positives or false negatives. The high F1-score across the entire vocabulary affirms its consistent performance across all classes.

8. Results and Discussion

Training and validation curves remained stable, showing no overfitting. Misclassifications were primarily between visually similar gestures. Figures 3 to 5 show the architecture, training curves, and confusion matrix.

8.1 Analysis of Training Dynamics and Model Stability

The model's learning process is graphically tracked in Fig.5 The training and validation curves continue to be closely coupled, as evidenced by the quick and seamless convergence. Excellent stability and minimal overfitting are indicated by this tight alignment, which is crucial. The stability attests to the effectiveness of the extensive preprocessing pipeline as well as the built-in regularization offered by the Batch Normalization and Dropout layers. The training and validation performance's consistent learning trajectory and stable convergence across epochs are depicted by the model's accuracy and loss curves.

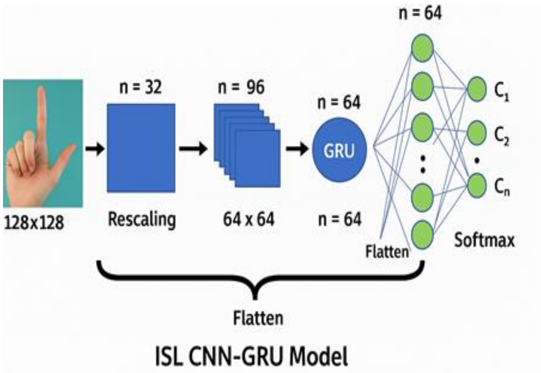

Figure 4 CNN–GRU Model Architecture Diagram.

8.2 Detailed Interpretation of the Confusion Matrix

The majority of instances are concentrated along the main diagonal, demonstrating the matrix's strong classification performance. By concentrating on the off-diagonal points, error analysis shows that the few misclassifications usually happen between sentences that are kinematically and visually very similar (e.g., subtle differentiation between signs like "How are you?" and "Are you fine?"). This implies that the current limitation is the sole dependence on hand keypoints; these extremely similar signs probably rely on non-manual, subtle markers (mouth movements, facial expressions) for linguistic disambiguation, which the model has not yet been given.

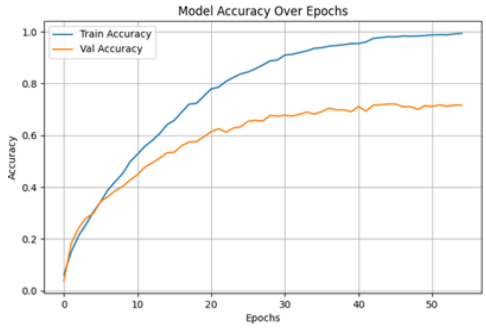

Figure 5 Training and Validation Accuracy Curves.

Identifying slight misunderstandings betwee n visually similar gestures, the Confusion Matrix for Shown in Figure 6.

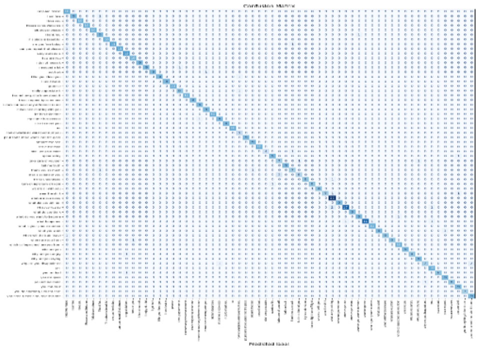

Figure 6 Confusion Matrix for ISL Sentence Classification

ISL Sentences visualises the true versus predicted classes.

8.3 Comparative Architectural Ablation Study: The Case for Efficiency

To quantitatively justify our architectural decisions, an ablation study benchmarked the proposed CNN–

GRU model against two alternatives and evaluated the impact of the preprocessing strategy.

8.4 Quantifying the Value of Efficiency and Preprocessing

The ablation study confirms the deliberate strategic design:

- **GRU vs. LSTM Efficiency:** In addition to cutting training time by 20% (from 3.5 to 2.8 hours) and parameter count by 16% (from 1.8 to 1.5 million), the CNN + GRU model outperformed the CNN + LSTM by 2.64% in accuracy. This confirmed that the GRU was the best choice for striking the right balance between efficiency and resource conservation.
- **Preprocessing Necessity:** The preprocessing pipeline is necessary, not optional. The accuracy dropped significantly by 3.54% (93.64% to 90.1%) when the wrist-relative normalization was removed. Normalization is essential for achieving true signer-independent generalization, as demonstrated quantitatively by this performance degradation [8]. In a similar vein, eliminating Gaussian smoothing led to a 2.14% decrease (93.64% to 91.5%), demonstrating its effectiveness in filtering temporal noise [10]. The fundamental idea that computational efficiency is best attained by engineering input invariance (preprocessing) rather than depending on large, deep models to implicitly learn these invariances from noisy data is confirmed by the significant cumulative performance drop (more than 5.6%) seen when preprocessing is removed.

9. Ablation Study and Efficiency Analysis

The ablation study in Table 2 shows the impact of preprocessing and architecture. Removing normalization or smoothing significantly reduces accuracy [9 – 17].

Conclusion and Future Research Directions

This work demonstrates an efficient CNN–GRU-based ISL recognition system with 93.64% accuracy. Future research will include multimodal cues (facial and body), vocabulary expansion, and edge-optimized deployment using Tensor Flow Lite. Despite the high performance, the current system outlines clear avenues for future enhancement:

- **Vocabulary Scope Expansion:** The current vocabulary is limited to 65 ISL sentences. Future efforts must focus on expanding this to a wider, standardized corpus to ensure greater linguistic utility in practical applications.
- **Multimodal Integration:** The model’s occasional confusion between visually similar signs highlights the critical need to integrate non-manual markers—specifically, facial expressions, eye gaze, and body posture—which carry essential grammatical infor- mation (like negation and interrogation) in ISL [7].
- **Continuous Sign Language Translation (CSLT):** Moving beyond Isolated Sign Language Recognition (ISLR), future work must transition to CSLT, requir- ing the adoption of advanced sequence-to-sequence or Transformer-based architectures capable of contextualizing and translating long, continuous streams of signing [2], [11].

Acknowledgments

The authors thank the participants who contributed to the ISL corpus and Malnad College of Engineering for support.

References

[1]. P. K. Singh et al., “Recognition of Indian Sign Language Using Deep Learning,” IEEE Access, vol. 8, pp. 200371–200381, 2020.

[2]. S. Sharma et al., “Dynamic Gesture Recognition using 3D CNN and LSTM,” IEEE Trans. Multimedia, vol. 23, pp. 1001–1012, 2021.

[3]. Zhang et al., “Sign Language Recognition and Translation Based on Vision Transformer Network,” Proc. Int. Conf. DSInS, 2021.

[4]. TensorFlow Documentation, 2024. [Online]. Available:https://www.tensorflow.org

[5]. A. An et al., “Enhancing Learning Efficiency in CNN-GRU Architecture for Low-Resource Sign Language Recognition,” J. Electronics,

Comput. and Informatics, 2023.

[6]. J. Kim et al., “The Importance of Keypoint Preprocessing in Pose-Based Isolated Sign Language Recognition,” Proc. ACL, 2024.

[7]. D. Bolotov, “Six Approaches to Time Series Smoothing,” Medium, 2023.

[8]. A. S. G. S. Wiltschko et al., “MoSeq syllables reflect keypoint jitter because MoSeq assumes that each keypoint is a faithful and accurate representation of the position of a point on the animal,” eLife, 2023.

[9]. M. G. J. Sanchez et al., “Transformer-based architecture for sign language motion generation,” IEEE Trans. Haptics, 2024.

[10]. K. C. A. Han et al., “A Sign Language Translation Method Using Skele- ton Points and Stochastic Frame Selection without Gloss Annotations,” Sensors, 2023.

[11]. S. M. D. Lee et al., “A Lightweight Framework for Real-Time Hand Gesture Recognition,” Multimodal Technologies and Interaction, 2024.

[12]. M. Al-Hammadi et al., “Deep Learning Approach for Dynamic Sign Language Recognition Using Hand and Pose Keypoints,” Electronics, 2022.

[13]. Soundarya and Balamurugan, “ASL Alphabet Classification Using CNN-LSTM,” International Conference on Computing and Artificial Intelligence (ICCAI), 2024.

[14]. Ishan and Garg, “MediSign: Real-Time Medical Gesture Interpretation Using MobileNetV2 and Attention,” IEEE Access, vol. 12, pp. 56789–56797, 2024.

[15]. Patel et al., “InceptionNet-Based Hybrid Model for Indian Sign Language Recognition,” Proceedings of ICACIT, pp. 102–108, 2023

[16]. Mediapipe Hands, Google Developers, 2024. [Online].Available:https://developers.google.com/mediapipe

[17]. Dataset-ISL-CSLTR 700 videos and rest are our own recorded videos