IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151116

SignAI: Intelligent Bidirectional Indian Sign Language Translation Using Skeleton Keypoints

Anand S. Balapure[1], Vedant Y. Badgujar[1], Netra A. Jayaswal[1], Tejaswini B. Sharma[1], Sheetal P. Patil[2]

Student, Department of Artificial Intelligence and Machine Learning, R. C. Patel Institute of Technology, Shirpur, Maharashtra, India[1]

Assistant Professor, Department of Artificial Intelligence and Machine Learning, R. C. Patel Institute of Technology, Shirpur, Maharashtra, India[2]

Abstract: Communication between Deaf and hearing individuals remains difficult because interpreters and accessible translation tools are scarce. This paper presents SignAI, a bidirectional Indian Sign Language (ISL) translation prototype that combines Sign-to-Text (S2T) and Text-to-Sign (T2S) translation in a single web application. The S2T module classifies isolated signs from MediaPipe Holistic keypoints using a Conv1D–BiGRU–attention network, supported by motion-based segmentation and confidence gating; recognised glosses are turned into English sentences with a rule-based generator or few-shot prompting of FLAN-T5-base. The T2S module maps text to validated glosses and renders the matching sign clips as a skeleton video, with optional PoseTransformer and fingerspelling support. On a custom 60-class ISL adjective dataset of 1,022 clips, the classifier reaches 74.29% top-1 and 88.57% top-5 accuracy on the test set (n = 140; 95% CI: 66.47–80.81%). At a confidence threshold of 0.65, 89.16% of accepted predictions are correct at 59.29% coverage. A data and code audit and its corrective measures are also reported.

Keywords: Sign language recognition, Indian Sign Language, MediaPipe, skeleton keypoints, bidirectional GRU, sign language production, confidence gating.

I. INTRODUCTION

Sign languages are natural languages with their own grammar, vocabulary and spatial structure, not signed versions of spoken languages [1], [2]. Deaf and hard-of-hearing people often face communication barriers without an interpreter, and automatic sign language recognition (SLR) and production (SLP) are complementary solutions. Most published systems address only one direction [1], [3], [4], and few integrated bidirectional systems describe their failures openly.

This paper presents SignAI, a bidirectional ISL prototype delivered through a web interface. The Sign-to-Text direction recognises signs from live webcam video using 2D skeleton keypoints from MediaPipe [5] and a Conv1D–BiGRU–attention classifier. The Text-to-Sign direction maps English text to a gloss sequence and renders a skeleton video, mainly from stored keypoint clips. The contributions are intentionally modest and based on project evidence:

- **System integration:** a bidirectional prototype with live-webcam recognition, a motion-based segmentation state machine, confidence gating, a gloss-to-sentence generator and a retrieval-based text-to-sign pipeline.
- **Evaluation and calibration:** test-split metrics with confidence intervals, and a threshold sweep quantifying the accuracy–coverage trade-off of gated isolated-sign recognition.
- **Failure analysis:** an audit of data leakage, landmark normalisation, transition contamination, missing background class and class imbalance, with the status of each fix.

No claims are made of architectural novelty, state-of-the-art accuracy or continuous translation; limitations are summarised in Section VIII.

II. RELATED WORK

Isolated recognition. Word-level SLR classifies a pre-segmented clip into one of a fixed vocabulary of signs. On the WLASL benchmark [3], an I3D RGB model reached 65.89% top-1 accuracy and a skeleton-based Pose-TGCN 55.43% on the 100-word subset, with accuracy dropping as vocabulary grows. Skeleton-aware multi-modal fusion [6] and hand-model-aware self-supervised pre-training [7] further improve recognition, while AUTSL [8] and INCLUDE [9] support Turkish and Indian sign language research. Like [6], our method uses only 2D keypoints, reducing dependence on appearance and background at the cost of fine visual detail.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151116

Continuous translation. Continuous systems operate on unsegmented sentences. Camgöz et al. introduced neural sign language translation on PHOENIX-2014T [2] and a Transformer that learns recognition and translation jointly [1]. Our system instead classifies segmented isolated signs and generates an English sentence from recognised glosses, which is not end-to-end translation.

Sign language production. Progressive Transformers [4] were an early end-to-end approach to pose generation, but regression-to-the-mean yields under-articulated signs. Retrieval and stitching approaches concatenate real recordings and avoid this, but have a closed vocabulary and cannot model co-articulation. Human evaluation by signers is needed to judge intelligibility [4]. Our T2S direction is mainly retrieval-based, with an optional learned PoseTransformer built on the Transformer [10].

Research gap. Skeleton-based systems are typically evaluated offline on pre-segmented clips, address a single direction and seldom document failure causes such as split leakage; this work is therefore an engineering and evaluation case study.

III. SYSTEM OVERVIEW

SignAI is a Python application with an HTTP server that serves a browser interface with two panels. In the S2T panel the browser streams webcam frames to the server, which extracts keypoints, runs the classifier through the segmentation state machine and returns suggestions, accepted glosses, sentences and a skeleton overlay. In the T2S panel the user types text and receives a rendered video. It uses PyTorch 2.1.0, MediaPipe 0.10.9, OpenCV 4.9.0.80 and HuggingFace Transformers 4.38.0 [11], [12]. Fig. 1 shows both workflows. The system implements isolated sign classification (S2T), word-level sign retrieval with video concatenation (T2S) and partial sentence composition from recognised glosses.



Fig. 1. Overview of the bidirectional system: (a) Sign-to-Text workflow (top) and (b) Text-to-Sign workflow (bottom).

IV. SIGN-TO-TEXT METHOD

A. *Input Representation*

Frames are processed with MediaPipe Holistic (model complexity 1, landmark smoothing enabled) [5]. Per frame, 33 pose, 21 left-hand and 21 right-hand landmarks are kept, i.e. 75 joints with only x and y coordinates, giving a 150-dimensional feature vector. Face landmarks and depth are not used, so non-manual markers such as facial expression and mouth patterns are not captured.

B. *Preprocessing*

Each clip is trimmed of inactive frames using a hand-motion energy threshold of 0.008. Missing landmarks are filled across gaps of up to two frames. Each frame is normalised by centring on the shoulder midpoint and dividing by shoulder width; lower-body joints are zeroed. The sequence is resampled to 30 frames by nearest-neighbour indexing, giving a 30 × 150 tensor. An earlier normalisation bug found in the audit is not present in the final code.

C. *Classifier*

The classifier (checkpoint *conv_gru_v2*) is a Conv1D–BiGRU–attention network (Fig. 2, Table I). The input is linearly projected to 128 features and passed through two temporal convolution blocks with a skip connection, then a two-layer bidirectional GRU [13] with layer normalisation and an additive attention temporal pooling layer [14], producing a 256-

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151116

feature clip representation. A three-layer head outputs logits for 60 classes (59 signs plus NO_SIGN). The model is about 3.0 MB and uses standard building blocks; no architectural novelty is claimed.

D. *Real-Time Inference Engine*

The live stream is segmented into candidate signs by a three-state machine. In *IDLE* it waits for sustained hand motion above a threshold; in *COLLECTING* it gathers frames until a pause or velocity dip signals a boundary; in *PAUSE* it verifies the boundary and either classifies the collected sequence or returns to COLLECTING if motion resumes. False predictions are limited by: marking outputs below the confidence threshold as uncertain; a NO_SIGN class that absorbs background and transitional motion; a 1.2 s refractory period for repeating the same sign; and live suggestions smoothed by an exponential moving average ($\alpha = 0.35$) that must persist for three consecutive predictions.

Gloss-to-sentence: accepted glosses are converted into English either by a rule-based generator covering about a dozen patterns (pronoun + adjective, noun + adjective, simple SVO, questions and compound sentences) or by prompting google/flan-t5-base [15] with eight few-shot examples. This is a demonstration feature not yet evaluated with BLEU or human assessment.

TABLE I CLASSIFIER CONFIGURATION (CONV_GRU_V2)

Component	Configuration	Output
Projection	Linear 150 → 128	(B,30,128)
Conv1D	2 × [Conv1d(128, k=3), BatchNorm, ReLU], dropout 0.25, residual	(B,128,30)
BiGRU	2 layers, 128 hidden/dir., dropout 0.5, LayerNorm	(B,30,256)
Attention	Linear 256→128, tanh, Linear 128→1, softmax	(B,256)
Head	256→256 → 128 → 60 (LN, ReLU, dropout)	(B,60)

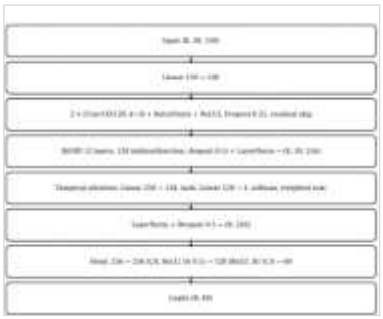


Fig. 2. Layer structure of the S2T classifier (conv_gru_v2).

V. TEXT-TO-SIGN METHOD

The T2S pipeline (Fig. 1b) has four stages. (1) *Text-to-gloss:* the input is normalised and glossed by a fine-tuned mT5-small model [16] trained on about 355 sentence–gloss pairs, by local vocabulary matching that drops grammar-specific words such as articles and copulas, or by exact phrase lookup in a sentence–gloss CSV file. (2) *Validation:* glosses without a sign resource are excluded. (3) *Pose-source resolution:* sources are used in priority order—validated source-video clips, the PoseTransformer, or preprocessed keypoint clips rendered as a skeleton. (4) *Rendering:* clips are concatenated with short pauses and encoded to MP4. For out-of-vocabulary words a fingerspelling fallback is available; otherwise the default strict mode rejects unsupported glosses.

The PoseTransformer is a standard encoder–decoder Transformer [10] (hidden size 256, 4 heads, 4 encoder and 4 decoder layers) that greedily generates up to 160 pose frames of 150 dimensions. The T2S target vocabulary covers 62 ISL adjective descriptors, backed by 1,340 training and 347 validation keypoint files. The pipeline keeps English word order without function words and does not model ISL grammar, so it is word-level sign retrieval and concatenation.

VI. DATASET AND EXPERIMENTAL SETUP

A. *Custom ISL Adjective Dataset*

The classifier is trained on a custom ISL dataset created for this project, containing 59 sign classes, mostly adjectives or descriptors (e.g. BIG_LARGE, COLD, HAPPY, SLOW, TALL, WEAK), plus a NO_SIGN class, giving 60 classes. The processed data comprise 1,022 clips stored as 30 × 150 float32 arrays, split into 742 training, 140 validation and 140 test clips (72.6%, 13.7%, 13.7%), obtained from camera recordings and skeleton-video renderings. Clips per class range from 4 (NICE, THIN) to 42 (LONG, SHORT, TALL, WIDE).

B. *Data Quality Audit*

An audit identified five issues in earlier models: (i) landmark normalisation placed missing joints at phantom positions (fixed); (ii) raw videos and their skeleton renderings appeared across different splits, with 47, 39 and 7 shared videos between train–validation, train–test and validation–test respectively; (iii) arm raise and drop transitions at clip edges

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151116

(inactive-frame trimming added); (iv) no background class (NO_SIGN added); and (v) class imbalance (addressed with a weighted sampler). Splits were not re-verified after the latest run, so residual optimistic bias cannot be excluded.

C. *Training and Evaluation Protocol*

Training used a single laptop GPU (NVIDIA GeForce RTX 3050, 4 GB) with mixed precision for 80 epochs, using AdamW [18] (learning rate 5e-4 with cosine decay to 2.5e-5, weight decay 1e-4), batch size 32 and a class-weighted sampler. Augmentation used random scaling, rotation, Gaussian noise and temporal shifts, with mixup [17] ($\alpha = 0.2$) on 70% of batches. Recognition is reported as top-1, top-5 and macro accuracy; since the deployed system gates low-confidence outputs, gated top-1 accuracy and coverage (fraction of predictions accepted) are also given, with 95% confidence intervals. The operating threshold of 0.65 comes from the project threshold report (validation split).

VII. RESULTS AND DISCUSSION

A. *Recognition Accuracy*

Table II gives the re-evaluation of the saved production model. On the test split it achieves 74.29% top-1 accuracy (95% CI [66.47, 80.81]), 88.57% top-5 accuracy and 70.56% macro accuracy. Training-log accuracies are not used as evidence of generalisation, since the saved model re-evaluated lower on the validation split (71.43%).

TABLE II RE-EVALUATION OF THE PRODUCTION MODEL (60 CLASSES, 140 CLIPS PER SPLIT)

Metric	Validation	Test
Top-1 accuracy (95% CI)	71.43% [63.45, 78.26]	74.29% [66.47, 80.81]
Top-5 accuracy	87.86%	88.57%
Macro accuracy (all classes)	67.22%	70.56%
Gated top-1 accuracy, $\theta = 0.65$ (95% CI)	93.06% [84.75, 97.00]	89.16% [80.66, 94.19]
Gated coverage	51.43%	59.29%
Mean confidence, correct / incorrect	0.742 / 0.442	0.747 / 0.478

B. *Confidence Gating*

Table III shows the threshold sweep on the validation split. Raising the threshold increases the accuracy of accepted predictions but reduces coverage steeply: at $\theta = 0.65$, 51.4% of predictions are accepted with 93.1% accuracy; at $\theta = 0.75$ accuracy is 96.5% at 40.7% coverage; and at $\theta = 0.90$ all 26 accepted predictions are correct but only 18.6% are accepted. The test-split gated result (89.16% at 59.29% coverage) differs from the validation point, so calibration does not transfer exactly. Correct predictions carry higher mean confidence (0.747) than incorrect ones (0.478), supporting confidence-based rejection.

TABLE III THRESHOLD SWEEP (VALIDATION SPLIT, AS INFERRED FROM THE THRESHOLD REPORT)

Threshold	0.30	0.40	0.50	0.60	0.65	0.75	0.80	0.90
Accepted	88.6%	78.6%	66.4%	54.3%	51.4%	40.7%	35.7%	18.6%
Accepted accuracy	75.8%	80.0%	86.0%	92.1%	93.1%	96.5%	98.0%	100%
N accepted	124	110	93	76	72	57	50	26

C. *Error Analysis*

Confusions occur mainly between signs with similar trajectories: SHORT → SMALL_LITTLE (3 test clips, both with downward flat palms), LONG → SLOW, WIDE → BIG_LARGE (both with outstretched arms) and TALL → WARM. Because handshape and facial cues are not modelled, such pairs are hard to separate; some confident errors also occur (e.g. ALIVE → CHEAP, 0.870).

VIII. LIMITATIONS, CONCLUSION AND FUTURE WORK

SignAI integrates an isolated-sign classifier based on skeleton keypoints, a real-time segmentation and confidence-gating mechanism, and a retrieval-based text-to-sign pipeline in one web application. On a custom 60-class adjective dataset the classifier achieves 74.29% top-1 test accuracy, and gating at 0.65 yields 89.16% accuracy on 59.29% of predictions. The main contribution lies in the integration and failure analysis rather than headline accuracy. Key limitations are the small dataset with undocumented signers, possible residual split leakage, no per-class F1, no quantitative T2S evaluation and

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151116

no Deaf-community validation. Future work includes splitting by recording and signer, per-class metrics and cross-validation, LSTM and Transformer baselines, testing on WLASL [3] and INCLUDE [9], formal evaluation of the T2S component (vocabulary coverage, retrieval accuracy, back-translation and assessment by fluent signers), adding face and depth data, vocabulary expansion and a user study with Deaf participants.

REFERENCES

[1]. N. C. Camgöz, O. Koller, S. Hadfield and R. Bowden, "Sign language transformers: joint end-to-end sign language recognition and translation," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10023–10033.

[2]. N. C. Camgöz, S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural sign language translation," in *Proc. IEEE/CVF CVPR*, 2018, pp. 7784–7793.

[3]. D. Li, C. Rodriguez, X. Yu and H. Li, "Word-level deep sign language recognition from video: a new large-scale dataset and methods comparison," in *Proc. IEEE/CVF WACV*, 2020, pp. 1459–1469.

[4]. B. Saunders, N. C. Camgöz and R. Bowden, "Progressive transformers for end-to-end sign language production," in *Computer Vision – ECCV 2020*, LNCS vol. 12356, Springer, 2020, pp. 687–705.

[5]. C. Lugaresi et al., "MediaPipe: a framework for building perception pipelines," *arXiv preprint* arXiv:1906.08172, 2019.

[6]. S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li and Y. Fu, "Skeleton aware multi-modal sign language recognition," in *Proc. IEEE/CVF CVPR Workshops*, 2021, pp. 3413–3423.

[7]. H. Hu, W. Zhao, W. Zhou and H. Li, "SignBERT+: hand-model-aware self-supervised pre-training for sign language understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 11221–11239, 2023.

[8]. O. M. Sincan and H. Y. Keles, "AUTSL: a large scale multi-modal Turkish sign language dataset and baseline methods," *IEEE Access*, vol. 8, pp. 181340–181355, 2020.

[9]. A. Sridhar, R. G. Ganesan, P. Kumar and M. Khapra, "INCLUDE: a large scale dataset for Indian sign language recognition," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2020, pp. 1366–1375.

[10]. A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11]. A. Paszke et al., "PyTorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12]. T. Wolf et al., "Transformers: state-of-the-art natural language processing," in *Proc. EMNLP: System Demonstrations*, 2020, pp. 38–45.

[13]. K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.

[14]. D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[15]. H. W. Chung et al., "Scaling instruction-finetuned language models," *J. Mach. Learn. Res.*, vol. 25, no. 70, pp. 1–53, 2024.

[16]. L. Xue et al., "mT5: a massively multilingual pre-trained text-to-text transformer," in *Proc. NAACL-HLT*, 2021, pp. 483–498.

[17]. H. Zhang, M. Cissé, Y. N. Dauphin and D. Lopez-Paz, "mixup: beyond empirical risk minimization," in *Proc. ICLR*, 2018.

[18]. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

 This work is licensed under a Creative Commons Attribution 4.0 International License