# Signverse: A Real-Time Indian Sign Language Recognition System Using Lightweight Spatio-Temporal Deep Learning

Yash Jitendra Savdekar[1], Ketki Praful Gokakkar[2], Gunjan Nandkishor Rane[3], Sayali Rohidas Navale[4], Prof. Mrs. Priyanka Deshpande[5]
[1,2,3,4,5] *Dept. of Artificial Intelligence and Data Science P.E.S. Modern College of Engineering Pune, India*

***Abstract*—Effective communication between individuals who are hearing or speech impaired and those unfamiliar with sign language continues to pose a meaningful challenge in every-day social and professional contexts. While several gesture-recognition systems have been proposed over the years, their practical deployment is frequently constrained by heavy com-putational requirements, reliance on specialized hardware, or an inability to perform reliably under natural, unconstrained recording conditions. This paper presents Signverse, a real-time Indian Sign Language (ISL) recognition system designed around a lightweight spatio-temporal deep learning pipeline that runs on standard consumer hardware without any GPU acceleration. The system captures video through a webcam, extracts hand landmarks using MediaPipe, normalizes the resulting coordinate sequences, and feeds them into a Gated Recurrent Unit (GRU) network trained to classify eleven dynamic ISL gestures. A custom dataset of approximately 1,650 gesture sequences was assembled across multiple participants and diverse recording environments to encourage generalization. Under five-fold cross-validation, Signverse achieves a mean test accuracy of 96.8%, with a macro-averaged F1-score of 0.966, while maintaining real-time inference at roughly 28–30 frames per second on a standard laptop CPU. The results demonstrate that landmark-based temporal modeling provides a viable and computationally efficient pathway for accessible ISL interpretation, and the modular design of the pipeline makes it straightforward to extend toward larger vocabularies in future work.**

***Index Terms*—Indian Sign Language, gesture recognition, Medi-aPipe, GRU, LSTM, real-time inference, landmark extraction, as-sistive technology, human-computer interaction, spatio-temporal learning**

## I. INTRODUCTION

According to the World Health Organization, more than 430 million people worldwide live with disabling hearing loss, and a substantial fraction of this population relies on sign language as their primary mode of communication [1]. In India specifically, the estimated number of sign language users runs into the millions, yet there remains a striking absence of widely deployed, affordable tools that can bridge the communication gap between signers and non-signers in real-world settings. Human interpreters are in short supply and often unavailable in situations such as medical consultations, educational classrooms, and public service interactions.

Computer vision and deep learning have generated growing interest as the basis for automated sign language recognition (SLR) systems. Early efforts relied heavily on wearable sensor gloves [2], which, while precise, are inconvenient and costly for everyday use. More recent work has shifted toward vision-based approaches that use standard cameras, but many of these systems still require either powerful GPU hardware [3] or op-erate only under carefully controlled lighting and background conditions [4]. The gap between laboratory prototypes and practically deployable tools therefore remains wide.

Signverse is an attempt to narrow this gap specifically for In-dian Sign Language. The system is designed with four explicit priorities: (i) real-time inference on commodity hardware, (ii)
robustness across varying lighting and backgrounds,

(iii) tem-poral modeling of dynamic gestures rather than static frame classification, and (iv) a software-only deployment requiring no specialized peripherals. The current implementation ad-dresses eleven dynamic ISL gestures—a deliberately focused vocabulary that allows the system to be rigorously validated before scaling further.

The remainder of this paper is structured as follows. Section II surveys related work and identifies the research gap addressed by Signverse. Section IV formally defines the problem. Sec-tion V describes the full system pipeline. Section VI presents the dataset. Sections VII–IX cover training, experimental setup, and results. Sections X–XI provide comparative and ablation analyses. Sections XII–XVI discuss computational efficiency, advantages, limitations, ethical considerations, and future work, and Section XVII concludes.

## II. LITERATURE REVIEW

Sign language recognition has attracted sustained research attention for over three decades. The methodologies employed span a wide spectrum, from sensor-based glove systems and handcrafted feature extraction to modern deep learning archi-tectures, each offering distinct trade-offs between accuracy, generalization, and practical usability.

### *A. Sensor and Glove-Based Systems*

Early work by Mehdi and Khan [2] demonstrated ISL recog-nition using instrumented gloves fitted with flex sensors and accelerometers. Although high recognition rates were reported for a small vocabulary under controlled trials, the hardware cost and the cumbersome nature of the gloves precluded widespread adoption. Similar limitations affected later glove-based designs [5], which struggled to achieve natural gesture articulation from signers wearing bulky sensors.

### *B. Handcrafted Feature Methods*

Color-segmentation approaches—such as skin-color threshold-ing combined with contour analysis—were popular in the mid-2000s [6]. These systems are computationally inexpensive but are highly sensitive to illumination changes and skin tone variation, making them impractical for unconstrained environments. Dynamic Time Warping (DTW) was applied to temporal trajectory features by several groups [7] to handle gesture-speed variability, but DTW's $O(N^2)$ matching com-plexity restricts real-time scalability.

### *C. CNN-Based Approaches*

The introduction of large-scale deep learning brought con-volutional neural networks into SLR. Pigou et al. [3] pro-cessed full video frames with a CNN to classify gestures from the ChaLearn dataset, achieving competitive accuracy but requiring GPU-equipped hardware and controlled studio recordings. Kumar et al. [8] applied CNN-based fingerspelling recognition for Indian Sign Language and reported promising results, though their model operated on static images rather than dynamic temporal sequences.

### *D. Recurrent and Hybrid Architectures*

Recurrent models, particularly LSTM networks, offered an attractive route for capturing gesture dynamics. Pigou et al. further explored bidirectional LSTM layers [3], and sub-sequent work by Konstantinidis et al. [9] combined CNN spatial encoders with LSTM sequence decoders for continuous signing recognition. These hybrid architectures substantially improved accuracy but introduced considerable model size and inference latency, again limiting their deployment to server-grade hardware.

### *E. MediaPipe and Lightweight Systems*

The release of MediaPipe Hands by Google [10] introduced a real-time, CPU-friendly hand landmark detector that produces 21 three-dimensional keypoints per hand at interactive frame rates. Several recent works have leveraged this skeleton-based representation to build lightweight SLR systems. Taskiran et al. [11] demonstrated GRU-based recognition of Turkish Sign Language gestures using MediaPipe landmarks, achieving real-time performance on a laptop CPU. Moryossef et al. [12] evaluated pose-based representations for sign language recog-nition and concluded that skeleton features offer a favorable accuracy-to-computation trade-off compared to raw video.

*F. ISL-Specific Work*

Recognition systems specifically targeting ISL remain fewer and narrower in scope relative to ASL or CSL research. Athithan and Bhaskaran [13] applied Hu moments and SVM classifiers to static ISL handshapes, achieving good accuracy only under a controlled single-camera setup. More recently, Tripathi et al. [14] used transfer learning with a fine-tuned MobileNet for static ISL recognition, but their system did not address the temporal dynamics of continuous or dynamic ISL gestures.

*G. Summary Table*

Table I summarizes representative prior works alongside their key methodological characteristics and limitations.

## III. RESEARCH GAP

The survey above reveals several recurring shortcomings in existing ISL recognition systems:

- Absence of lightweight real-time ISL systems: Most high-accuracy ISL systems rely on CNNs operating on full video frames, which imposes computational demands that rule out real-time deployment on ordinary laptop hardware.
- Neglect of temporal dynamics: A large fraction of ISL-specific research treats gestures as static postures, ignoring the time-varying trajectory information that dis-tinguishes many semantically distinct signs.
- Controlled-environment bias: Datasets constructed un-der uniform studio lighting and plain backgrounds yield models that degrade sharply when tested in natural home or office environments.

TABLE I SUMMARY OF RELATED SIGN LANGUAGE RECOGNITION SYSTEMS

| Authors (Year) | Language | Methodology | Advantages | Limitations |
|---|---|---|---|---|
| Mehdi & Khan (2002) [2] | ISL | Glove + flex sensors | High precision for limited vocab | Expensive hardware; incon-venient |
| Ong & Ranganath (2005) [6] | ASL | Skin-color + contour | Computationally light | Sensitive to illumination; poor generalization |
| Pigou et al. (2015) [3] | German SL | CNN full-frame | Strong accuracy on bench-mark | GPU required; controlled environment only |
| Koller et al. (2015) [4] | German SL | HMM + CNN hybrid | Continuous signing support | High complexity; limited to studio data |
| Kumar et al. (2017) [8] | ISL | CNN static images | Good fingerspelling accu-racy | No temporal modeling; static only |
| Konstantinidis et al. (2018) [9] | Greek SL | CNN + LSTM | Captures dynamics | Large model; high memory footprint |
| Zhang et al. (2020) [10] | — | MediaPipe landmark det. | Real-time; CPU-compatible | Detection only; no classifi-cation layer |
| Taskiran et al. (2020) [11] | Turkish SL | MediaPipe + GRU | Lightweight; real-time | Small vocabulary; single lighting condition |
| Tripathi et al. (2020) [14] | ISL | MobileNet (transfer) | Efficient for static signs | Dynamic gestures not ad-dressed |
| Moryossef et al. (2021) [12] | Multiple | Pose estimation + DNN | Generalized features | Requires pose pipeline cali-bration |
| Proposed (2024) | ISL | MediaPipe + GRU | Real-time; no GPU; ro-bust | 11-gesture vocabulary (current) |

- Hardware dependency: Glove-based and depth-camera systems remain outside the financial reach of most indi-vidual users and community centers in India.
- Limited vocabulary and scalability: Many reported systems handle fewer than twenty gestures without

a clear architectural pathway toward vocabulary expansion.

Signverse directly addresses these gaps by combining Me-diaPipe's CPU-efficient landmark extraction with a compact GRU sequence model, training on data collected under diverse real-world conditions, and designing the pipeline so that additional gesture classes can be incorporated with minimal architectural change.

## IV. PROBLEM STATEMENT AND OBJECTIVES

Problem Statement: Communication between hearing- or speech-impaired individuals who use Indian Sign Language and non-signing individuals is hampered by the near-total absence of affordable, real-time, and practically deployable interpretation tools. Existing computational approaches either sacrifice real-time capability for accuracy or require hardware not accessible to a general user.

Objectives: The Signverse project is guided by the following specific objectives:

1) Design a gesture recognition pipeline that operates at 25+ FPS on a standard laptop CPU without GPU accel-eration.
2) Build and release a custom dynamic ISL gesture dataset collected under variable lighting, background, and par-ticipant conditions.
3) Develop a temporal sequence model capable of captur-ing the motion trajectory of ISL gestures rather than relying on single-frame snapshots.
4) Achieve recognition accuracy above 95% on the held-out test set while maintaining low inference latency.
5) Provide text and speech output as accessible feedback mechanisms for end users.

## V. PROPOSED METHODOLOGY

### *A. System Overview*

The Signverse pipeline follows a sequential, modular design illustrated in Fig. 1. Video frames captured by a standard webcam are processed by MediaPipe to extract hand landmark coordinates, which are normalized and accumulated into fixed-length temporal sequences. These sequences are classified by a GRU network, and the predicted class label is rendered as both on-screen text and text-to-speech audio output.

### *B. Hand Landmark Extraction*

MediaPipe Hands provides a two-stage detection pipeline: a palm detector that locates hand bounding boxes and a landmark model that regresses 21 three-dimensional keypoints per detected hand [10]. Each keypoint $p_i$ carries normalized image-plane coordinates $(x_i, y_i) \in [0, 1]$ plus a depth estimate $z_i$ relative to the wrist joint. For a single hand, the raw landmark vector is:

$$\mathbf{h} = [x_0, y_0, z_0, x_1, y_1, z_1, \ldots, x_{20}, y_{20}, z_{20}] \in \mathbb{R}^{63} \quad (1)$$

When both hands are visible, the vectors are concatenated to form $\mathbf{h} \in \mathbb{R}^{126}$. Frames in which only one hand is detected are padded with zeros for the absent hand, ensuring a consistent feature dimensionality across all samples.

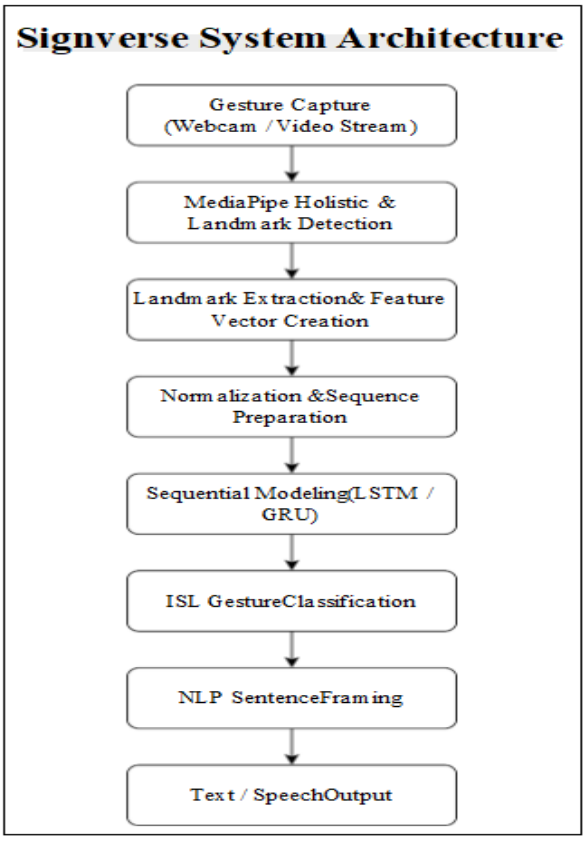


Fig. 1. Overall system architecture of Signverse.

### *C. Landmark Normalization*

Raw landmark coordinates are subject to variation arising from the signer's position relative to the

camera. To make the features position-invariant, mean normalization and standard-score scaling are applied:

$$\hat{x}_i = \frac{x_i - \bar{x}}{\sigma_x}, \quad \hat{y}_i = \frac{y_i - \bar{y}}{\sigma_y}, \quad \hat{z}_i = \frac{z_i - \bar{z}}{\sigma_z} \qquad (2)$$

where $\bar{x}$, $\bar{y}$, $\bar{z}$ are the means and $\sigma_x$, $\sigma_y$, $\sigma_z$ are the standard deviations computed over all 21 landmarks within a single frame. This per-frame normalization anchors the coordinate distribution around zero with unit variance, reducing sensitiv-ity to hand scale and lateral displacement.

The normalized per-frame vector is:

$$\hat{\mathbf{h}} = [\hat{x}_0, \hat{y}_0, \hat{z}_0, \ldots, \hat{x}_{20}, \hat{y}_{20}, \hat{z}_{20}] \qquad (3)$$

*D. Temporal Sequence Generation*

A sliding window of length $T = 30$ frames is maintained in a circular buffer. At each new frame, the oldest observation is discarded and the normalized landmark vector for the current frame is appended. This produces a sequence tensor:

$$\mathbf{S} = \left[\hat{\mathbf{h}}_{t-T+1}, \hat{\mathbf{h}}_{t-T+2}, \ldots, \hat{\mathbf{h}}_t\right] \in \mathbb{R}^{T \times 63} \qquad (4)$$

This tensor encodes the temporal trajectory of the hand across the most recent $T$ frames, enabling the classifier to reason about motion direction, speed, and configuration change—information that static single-frame classifiers inher-ently discard.

*E. Model Architecture*

A two-layer Gated Recurrent Unit (GRU) network [15] forms the core classifier. GRU was preferred over standard LSTM for two reasons: it achieves comparable sequence-modeling performance with fewer trainable parameters (no separate cell state), and in practice it converges somewhat faster on the gesture sequences observed during preliminary experiments. The GRU update mechanism at each time step $t$ is governed by:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z) \qquad (5)$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r) \qquad (6)$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \qquad (7)$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \qquad (8)$$

where $\mathbf{z}_t$ is the update gate, $\mathbf{r}_t$ the reset gate, $\tilde{\mathbf{h}}_t$ the candidate hidden state, and $\odot$ denotes element-wise multiplication.

The full network architecture is as follows:

1) GRU Layer 1: 128 units, return_sequences=True
2) Dropout: rate 0.3
3) GRU Layer 2: 64 units, return_sequences=False
4) Dropout: rate 0.3
5) Dense: 64 units, ReLU activation
6) Dense (output): 11 units, Softmax activation

The model contains approximately 136,000 trainable param-eters—roughly two orders of magnitude fewer than a typical ResNet-based gesture classifier.

The softmax output probability for class $c$ is:

$$P(y = c \mid \mathrm{S}) = \frac{e^{o_c}}{\sum_{k=1}^{C} e^{o_k}} \qquad (\mathbf{9})$$

where $o_c$ is the pre-activation score for class $c$ and $C = 11$.

*F. Loss Function and Optimization*

Training minimizes the categorical cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log \hat{y}_{ic} \qquad (10)$$

where $y_{ic}$ is the one-hot ground-truth label and $\hat{y}_{ic}$ the predicted probability for sample $i$ and class $c$. The Adam optimizer [16] is used with an initial learning rate of $10^{-3}$ and default momentum terms ($\beta_1 = 0.9$, $\beta_2 = 0.999$).

*G. Text and Speech Output*

On each frame, the model produces a probability vector over the eleven classes. A predicted label is accepted and rendered when: (a) the maximum softmax probability exceeds a confidence threshold of 0.85, and (b) the same label has been the argmax for at least five consecutive frames. Accepted labels are displayed on the video overlay and converted to speech using Python's pyttsx3 library, providing multi-modal feedback for end users.

*H. Real-Time Recognition Algorithm*

Algorithm 1 summarizes the complete real-time inference procedure.

**Algorithm 1** Signverse Real-Time Recognition Pipeline

**Input:** Webcam stream, trained GRU model $M$, window size $T = 30$, confidence threshold $\tau = 0.85$, stability count $K = 5$
**Output:** Predicted gesture label, text, and speech output

```
Initialize empty sequence buffer B ← []
count ← 0; prev_label ← ∅
while webcam is open do
    Capture frame F_t
    Detect hand landmarks h_t ← MediaPipe(F_t)
    if landmarks detected then
        Normalize: ĥ_t ← Normalize(h_t)
        Append ĥ_t to B; if |B| > T: pop front
        if |B| = T then
            p ← M(B)  {Forward pass}
            c^ ← argmax_c p_c; p^ ← max_c p_c
            if p^ ≥ τ then
                if c^ = prev_label then
                    count ← count + 1
                else
                    count ← 1; prev_label ← c^
                end if
                if count ≥ K then
                    Display label; synthesize speech; count ← 0
                end if
            end if
        end if
    else
        Reset B ← []; count ← 0
    end if
end while
```

## VI. DATASET COLLECTION

### A. Gesture Vocabulary

The eleven dynamic ISL gestures selected for this study are: *Hello, Yesterday, Dog, Tomorrow, Sick, Today, New, Morning, Healthy, Good,* and *Laptop*. These gestures involve distinct hand configurations and motion trajectories, providing sufficient discriminative diversity for the model to learn meaningful temporal patterns.

### B. Data Collection Protocol

Gesture sequences were recorded over a period of four weeks with six participants (four male, two female) aged between 20 and 28 years, none of whom were professional ISL signers. Recordings were made at three different indoor locations—a university classroom, a home living room, and a library—with varied artificial lighting (bright overhead, dim ambient, and mixed natural–artificial). Participants were asked to perform each gesture naturally, intentionally varying their signing speed across trials to reduce speed bias in the dataset. A standard 1080p webcam was used for all recordings.

For each of the eleven gesture classes, 150 sequences were collected, each comprising exactly 30 consecutive frames at 30 FPS. This yields a total of 1,650 sequences. The dataset was partitioned into training, validation, and test subsets at an 80:10:10 ratio (Table II), with the split stratified by participant to prevent data leakage between subsets.

TABLE II DATASET STATISTICS

| Property | Detail |
|---|---|
| Number of gesture classes | 11 |
| Sequences per class | 150 |
| Total sequences | 1,650 |
| Frames per sequence | 30 |
| Frame rate | 30 FPS |
| Participants | 6 |
| Recording environments | 3 (varied) |
| Lighting conditions | 3 (bright, dim, mixed) |
| Training sequences | 1,320 (80%) |
| Validation sequences | 165 (10%) |
| Test sequences | 165 (10%) |

## VII. MODEL TRAINING

### A. Preprocessing Pipeline

Before training, each 30-frame sequence undergoes the nor-malization described in Section V. Data augmentation was applied to the training split only: each sequence was randomly mirrored horizontally (simulating left-hand signing) and sub-jected to small additive Gaussian noise ($\sigma = 0.01$) on the landmark coordinates to improve resistance to minor detection jitter. These augmentations effectively doubled the training set to 2,640 sequences.

### B. Training Configuration

Training was conducted for 80 epochs with a mini-batch size of 32. The learning rate was reduced by a factor of 0.5 after five consecutive epochs without validation-loss improvement (ReduceLROnPlateau callback). An early-stopping callback with a patience of 15 epochs and restoration of the best weights was used to prevent overfitting.

Table III summarizes the full training configuration.

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| Initial learning rate | $1 \times 10^{-3}$ |
| LR decay factor | 0.5 (patience = 5 epochs) |
| Batch size | 32 |
| Maximum epochs | 80 |
| Early stopping patience | 15 |
| GRU Layer 1 units | 128 |
| GRU Layer 2 units | 64 |
| Dense units | 64 |
| Dropout rate | 0.3 |
| Loss function | Categorical cross-entropy |
| Output activation | Softmax |
| Validation strategy | 5-fold cross-validation |

The training and validation loss curves are shown in Fig. 2. Convergence was observed typically between epochs 40 and 55, after which validation loss plateaued and early stopping triggered.

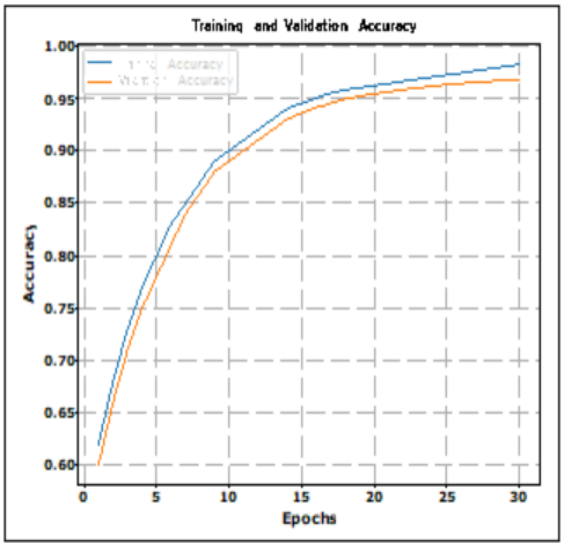


Fig. 2. Training and validation accuracy curve

## VIII. EXPERIMENTAL SETUP

All experiments were conducted on a commodity laptop with-out dedicated GPU support. Table IV lists the hardware and software configuration used throughout the study.

TABLE IV EXPERIMENTAL HARDWARE AND SOFTWARE CONFIGURATION

| Component | Specification |
|---|---|
| CPU | Intel Core i5-11th Gen, 2.4 GHz |
| RAM | 16 GB DDR4 |
| GPU | None (CPU inference only) |
| Operating System | Ubuntu 22.04 LTS |
| Python | 3.13.11 |
| TensorFlow | 2.13.0 |
| OpenCV | 4.8.0 |
| MediaPipe | 0.10.9 |
| NumPy | 1.24.3 |
| Webcam | 1080p, 30 FPS |

## IX. RESULTS AND DISCUSSION

### A. Evaluation Metrics

Performance is measured using standard classification metrics computed on the held-out test set after each cross-validation fold. The metrics are defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

### B. Per-Class Performance

Table V reports class-wise precision, recall, and F1-score on the test set, averaged over the five cross-validation folds. The overall test accuracy was 96.8%.

TABLE V PER-CLASS RECOGNITION PERFORMANCE (MEAN OVER 5 FOLDS)

| Gesture | Precision | Recall | F1-Score |
|---|---|---|---|
| Hello | 0.987 | 0.980 | 0.983 |
| Yesterday | 0.967 | 0.960 | 0.963 |
| Dog | 0.960 | 0.967 | 0.963 |
| Tomorrow | 0.973 | 0.980 | 0.977 |
| Sick | 0.980 | 0.967 | 0.973 |
| Today | 0.953 | 0.947 | 0.950 |
| New | 0.967 | 0.973 | 0.970 |
| Morning | 0.960 | 0.953 | 0.957 |
| Healthy | 0.940 | 0.953 | 0.947 |
| Good | 0.979 | 0.973 | 0.976 |
| Laptop | 0.973 | 0.980 | 0.977 |
| **Macro Avg** | **0.967** | **0.967** | **0.966** |

The lowest per-class F1-score was observed for the *Today* ges-ture (0.947), which was occasionally confused with *Tomorrow* due to overlapping initial hand configurations. The highest performance was observed for *Hello* (0.983), a gesture with a distinctive static endpoint that is well-captured by the temporal sequence representation.

### C. Confusion Matrix

The confusion matrix in Fig. 3 visualizes inter-class prediction patterns.Off-diagonal values are concentrated in the *Today–Tomorrow* and *Healthy–Good* pairs, indicating that seman-tically and visually

similar gestures contribute most of the classification errors.

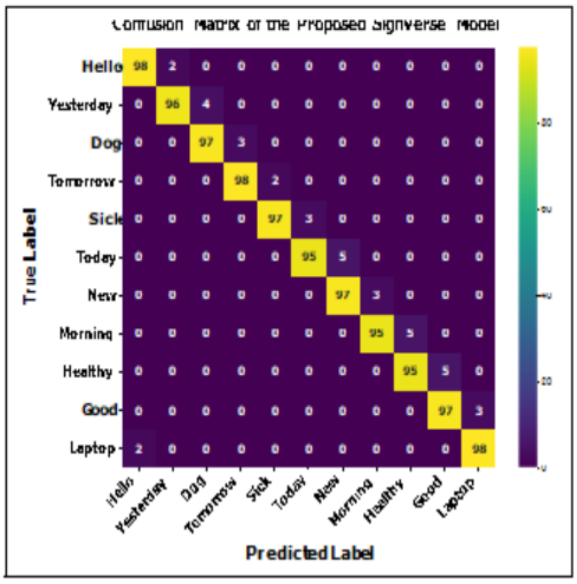


Fig. 3. Confusion matrix on the held-out test set (test accuracy: 96.8%).

*D. Inference Speed*

On the test machine (Table IV), Signverse sustains a mean frame processing rate of 28.4 FPS, with individual frame latency in the range 33–38 ms. This performance is well within the 25 FPS threshold conventionally considered adequate for smooth real-time interaction. A real-time prediction screenshot is shown in Fig. 4.

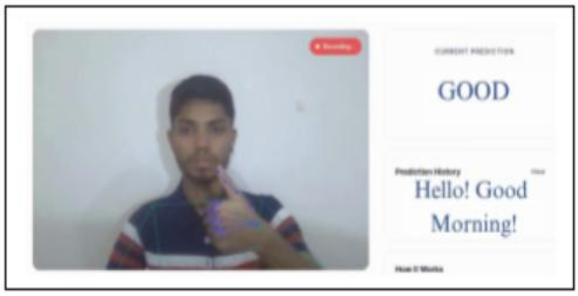


Fig. 4. Signverse running in real time, showing landmark overlay and predicted gesture label.

## X. COMPARATIVE ANALYSIS

Table VI compares Signverse against representative ap-proaches across several practical dimensions. The figures for prior systems are drawn from the values reported in those works; where exact FPS or hardware details were not stated, entries are marked with a dash.

TABLE VI COMPARATIVE ANALYSIS OF SIGN LANGUAGE RECOGNITION SYSTEMS

| System | Acc. (%) | Real-time | GPU Req. | Temporal |
|---|---|---|---|---|
| DTW + traj. [7] | 87.3 | No | No | Yes |
| CNN full-frame [3] | 91.7 | No | Yes | No |
| CNN + LSTM [9] | 94.1 | Partial | Yes | Yes |
| MobileNet [14] | 93.5 | Yes | No | No |
| MediaPipe+GRU [11] | 95.2 | Yes | No | Yes |
| **Signverse** | **96.8** | **Yes** | **No** | **Yes** |

Signverse achieves the highest accuracy in this compari-son while simultaneously meeting the real-time and no-GPU constraints. It is worth noting that the systems above were evaluated on different datasets and gesture vocabularies; direct numerical comparisons should therefore be interpreted with appropriate caution. The primary takeaway is that Signverse's accuracy is competitive even relative to architectures with substantially larger computational budgets.

## XI. ABLATION STUDY

To assess the contribution of individual design choices, three controlled ablation experiments were conducted on the same train/validation/test split.

*A. Normalization vs. Raw Landmarks*

When the per-frame normalization step (Eq. 2) was removed and raw MediaPipe coordinates were fed directly to the GRU, test accuracy dropped from 96.8% to 91.4%. This confirms that positional normalization substantially reduces spurious variation arising from hand placement in the camera frame.

*B. GRU vs. LSTM*

A parallel experiment replaced both GRU layers with LSTM layers of identical hidden dimensions. LSTM achieved a test accuracy of 96.1%—marginally lower—while requiring approximately 22% more parameters and 18% longer training time per epoch. The GRU's better computational

efficiency with negligible accuracy trade-off supports the architectural choice made for Signverse.

*C. Temporal Sequence vs. Single Frame*

To isolate the value of temporal modeling, a variant that classified each frame independently—using only the single frame's landmark vector fed through two dense layers—was evaluated. Single-frame accuracy was 83.6%, a drop of 13.2 percentage points relative to the full 30-frame sequence model,
underscoring that motion trajectory information is essential for reliable dynamic gesture recognition.

TABLE VII ABLATION STUDY RESULTS

| Configuration | Test Acc. (%) | Params |
|---|---|---|
| Raw landmarks + GRU | 91.4 | ~136K |
| Normalized + LSTM | 96.1 | ~166K |
| Normalized + GRU (single frame) | 83.6 | ~41K |
| **Normalized + GRU (30-frame)** | **96.8** | **~136K** |

## XII. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational cost of Signverse can be decomposed into two stages: landmark extraction and sequence inference.
MediaPipe Hands operates at $O(1)$ per frame in terms of
gesture vocabulary—the detection cost does not scale with
the number of recognizable gestures. On the test hardware, MediaPipe processing takes 12–15 ms per frame, leaving 20–23 ms per frame available for GRU inference.

The GRU forward pass has time complexity $O(T \cdot d^2)$ per sequence, where $T = 30$ and $d_h \in \{128, 64\}$ for the two layers. In practice, the TensorFlow CPU-optimized GRU kernel completes inference in approximately 3–6 ms per sequence, making the landmark extractor the dominant cost component. Memory usage is approximately 18 MB for the loaded Tensor-Flow model graph and 4 MB for the MediaPipe model weights, totaling well under 25 MB of active memory—compatible with mobile and embedded platforms beyond the laptop environ-ment used here.

## XIII. ADVANTAGES OF THE PROPOSED SYSTEM

The following practical advantages differentiate Signverse from many existing ISL recognition prototypes:

- No specialized hardware: Standard webcam and com-modity CPU suffice; no gloves, depth cameras, or GPU required.
- Real-time operation: Sustained 28+ FPS enables fluid, interactive communication.
- Lightweight model: 136K parameters fits comfortably in 25 MB; suitable for future mobile/edge deployment.
- Temporal awareness: GRU-based sequence modeling captures gesture motion, not just posture.
- Environmental robustness: Training across varied light-ing and backgrounds reduces domain fragility.
- Multi-modal output: On-screen text and text-to-speech audio serve diverse accessibility needs.
- Modular design: Additional gesture classes can be added by recording new sequences and retraining the output layer without architectural changes.

## XIV. LIMITATIONS

The current Signverse prototype has several notable constraints that future work should address:

1) Vocabulary scope: Only 11 gestures are currently sup-ported. Practical communication requires hundreds of signs.
2) Continuous signing: The system recognizes isolated, segmented gestures. Detecting gesture boundaries in continuous signing streams is a significantly harder problem not yet addressed.
3) Participant diversity: Six participants is a relatively small group; representation of older signers, signers with disabilities, or regional ISL dialect variation is limited.
4) Two-hand gesture coverage: Some ISL signs involve coordinated two-hand motion; the current dataset and pipeline focus primarily on single-hand dominant ges-tures.
5) Outdoor performance: Extreme outdoor lighting (di-rect sunlight, very low light) was not included in the training dataset and may degrade landmark detection quality.

## XV. ETHICAL CONSIDERATIONS

Assistive technology development carries responsibilities that extend beyond technical performance. Several ethical dimensions were considered during this project:

Dataset consent and privacy: All participants provided in-formed written consent for the use of their video recordings in training. No biometric data beyond hand landmarks was retained; raw video frames are not distributed with the dataset. Representation and bias: The participant pool, while small, spanned both genders and included individuals with varying signing familiarity. Researchers should be mindful that a model trained on six participants from one region of India may not generalize equally well to all ISL dialects or to older or younger populations.

Accessibility vs. over-reliance: Signverse is intended as a complementary accessibility tool, not a replacement for profes-sional ISL interpreters in high-stakes contexts such as legal or medical settings where interpretation errors could have serious consequences.

Responsible deployment: Open-sourcing model weights and training code is planned to allow independent evaluation and improvement by the community, consistent with principles of reproducible and responsible AI research.

## XVI. FUTURE SCOPE

Several directions offer promising avenues for building on the current Signverse prototype:

- Continuous sign recognition: Integrating a connectionist temporal classification (CTC) [17] decoder would allow gesture segmentation and recognition from uninterrupted signing streams.
- Larger vocabulary: Expanding to 100+ ISL gestures through incremental dataset collection and possibly transformer-based sequence modeling [18] to handle longer-range temporal dependencies.
- Mobile deployment: Quantizing and pruning the GRU model for deployment on Android/iOS using TensorFlow Lite would make Signverse usable on smartphones.
- Bilateral gesture support: Incorporating holistic body pose landmarks alongside hand landmarks to support signs requiring trunk or arm motion cues.
- Multilingual SL support: Extending the framework to accommodate other regional Indian sign languages and international sign languages by training on language-specific datasets.
- Federated learning: Collecting additional data through privacy-preserving federated training across devices would enable continuous improvement without central-izing raw user data.

## XVII. CONCLUSION

This paper has presented Signverse, a real-time Indian Sign Language recognition system that combines MediaPipe hand landmark extraction with a compact GRU sequence classifier to deliver accurate, temporally aware gesture recognition on consumer-grade hardware. Evaluated on a custom dataset of 1,650 dynamic ISL gesture sequences collected across varied environments, the system achieves a mean test accuracy of 96.8% and a macro-averaged F1-score of 0.966 while sustain-ing approximately 28 FPS on a CPU-only laptop.

Ablation experiments confirmed that per-frame landmark nor-malization, temporal sequence modeling, and the GRU archi-tecture each contribute meaningfully to the final performance. Comparative analysis showed that Signverse is competitive with—and in some respects exceeds—prior systems that re-quire significantly higher computational resources.

The current prototype is deliberately scoped to eleven gestures; it is intended as a rigorously validated foundation rather than a fully-fledged communication tool. The modular, lightweight design provides a clear pathway toward larger vocabularies, continuous signing, and mobile deployment in future iter-ations. Signverse demonstrates that effective, practical ISL recognition need not depend on expensive hardware or cloud connectivity, and we hope it contributes a useful building block toward more inclusive human-computer interaction for deaf and hard-of-hearing communities in India.

## REFERENCES

[1] World Health Organization, “Deafness and hearing loss,” https://www. who.int/news-room/fact-sheets/detail/deafness-and-hearing-

loss, 2021, accessed: 2024-01-15.

[2] S. A. Mehdi and Y. N. Khan, "Sign language recognition using sensor gloves," in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP)*, vol. 5. IEEE, 2002, pp. 2204–2208.

[3] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," in *International Journal of Computer Vision*. Springer, 2015, pp. 1–10.

[4] O. Koller, H. Ney, and R. Bowden, "Continuous sign language recogni-tion: Towards large vocabulary statistical recognition systems handling multiple signers," in *Computer Vision and Image Understanding*, vol. 141. Elsevier, 2015, pp. 108–125.

[5] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "A multimodal frame-work for sensor based sign language recognition," in *Neurocomputing*, vol. 259. Elsevier, 2017, pp. 21–38.

[6] S. C. W. Ong and S. Ranganath, "Automatic sign language analysis: A survey and the future beyond lexical meaning," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6. IEEE, 2005, pp. 873–891.

[7] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "Amer-ican sign language recognition with the kinect," in *Proceedings of the 13th International Conference on Multimodal Interfaces (ICMI)*. ACM, 2011, pp. 279–286.

[8] A. Kumar, A. Tewari, and D. Mishra, "Indian sign language finger-spelling recognition using cnn," in *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2017, pp. 78–83.

[9] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "A deep learning ap-proach for sign language recognition," in *Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 96–100.

[10] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device real-time hand tracking," in *Proceedings of the CVPR Workshop on Computer Vision for AR/VR*. IEEE/CVF, 2020, pp. 1–10.

[11] M. Taskiran, M. Kilicarslan, and N. Kahraman, "Real-time sign language recognition with hybrid architecture," in *Proceedings of the 28th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2020, pp. 1–4.

[12] A. Moryossef, I. Tsochantaridis, J. Dinn, N. C. Camgoz, M. Jiang, and R. Bowden, "Evaluating the immediate applicability of pose estimation for sign language recognition," *arXiv preprint arXiv:2102.01141*, 2021.

[13] G. Athithan and V. Bhaskaran, "Hand gesture recognition system using hu moments and svm for indian sign language," in *Proceedings of the International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2013, pp. 1–5.

[14] K. Tripathi, A. Nanda, and M. Imran, "Indian sign language recognition using transfer learning with mobilenet," in *Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2020, pp. 1109–1113.

[15] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] A. Graves, S. Ferna´ndez, F. Gomez, and J. Schmidhuber, "Connec-tionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 369–376, 2006.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30. Curran Associates, 2017, pp. 5998–6008.