JOURNALS | Journal of Biomedical Systems and Engineering
Volume 1 Issue 1 (September 2026)
J. Biomed. Syst. Eng., 2026, 1(1), 26802
DOI : https://doi.org/10.64189/bse.26803

Research Article

OPEN ACCESS

Real-Time ISL to Text Interpretation and Parkinson's Tremor Correction with Augmented Reality Support

Farhana Siddiqui, Iqra Ansari, Nusaybah Kazi* and Jamil Khatri

Department of Computer Engineering, Mohammed Haji Saboo Siddik College of Engineering, Mumbai, Maharashtra, 400008, India

*Corresponding Author
Nusaybah Kazi
nusaybah.231215.co@mhssce.ac.in

Received: 28 July 2026
Revised: 11 September 2026
Accepted: 17 September 2026
Published Online: 22 September 2026

Citation
F. Siddiqui, I. Ansari, N. Kazi, J. Khatri, Real-time ISL to text interpretation and Parkinson's tremor correction with augmented reality support, *Journal of Biomedical Systems and Engineering,* 2026, 1(1), 26803, https://doi.org/10.64189/bse.26803.

Abstract
Communication between Indian Sign Language (ISL) users and people who do not understand sign language can be difficult, whereas involuntary hand movements can make gesture-based interactions more challenging. This work presents a real-time assistive system that combines ISL alphabet recognition, landmark-based temporal stabilization, and augmented-reality (AR)-style visual feedback. A webcam provides the input stream, and MediaPipe Hands obtains 21 three-dimensional hand landmarks from each detected hand. The landmark representation is converted into 128 × 128 skeleton-based samples and classified using a convolutional neural network implemented with TensorFlow. The project dataset contains 500 images for each of the 26 ISL alphabet classes, yielding 13,000 images in total. For the reproducible quantitative evaluation reported here, a stratified 80:20 train test split with random_state = 42 was used, yielding 10,400 training and 2,600 held-out test images. The CNN achieved 99.27% accuracy, with a macro precision, macro recall, and macro F1 score equal to 99.27%. Live processing was benchmarked over 100 webcam frames at 108.40 ms per frame, corresponding to approximately 9.23 FPS. MediaPipe landmark processing accounted for 38.30 ms per frame, and CNN inference accounted for 69.73 ms per frame. The runtime implementation uses a confidence threshold of 0.85 and a temporal stability rule requiring 20 consecutive identical predictions before a letter is appended. Shaky-motion testing was qualitative, and AR-style output was demonstrated in the working prototype. The system is presented as an assistive recognition prototype and not as a medical treatment or diagnostic system.

Keywords: Indian Sign Language (ISL); Tremor support; Augmented reality (AR); Real-time gesture recognition; Computer vision; Assistive technology; Hand landmarks.

 https://doi.org/10.64189/bse.26803

1. Introduction

Indian sign language (ISL) is a structured visual language used for communication by sign-language users in India.[1-3]. Automatic recognition can help reduce communication barriers by converting visible signs into text and, where needed, speech.[1,3,4] Vision-based recognition is attractive because a conventional camera can capture hand configurations without the use of wearable sensors. Recent work has investigated skeleton and landmark representations because they preserve hand structure while reducing the dependence on raw RGB appearance, background clutter, illumination, and other irrelevant visual variations.[5-8] This representation is therefore appropriate for a lightweight real-time pipeline in which the classifier should focus on hand geometry rather than the complete camera image. Real-time recognition remains challenging because hand position, signing speed, occlusion, background conditions, and differences between users can affect recognition performance.[1,5,9,10] Involuntary hand movement can introduce short-term fluctuations into the observed landmark sequence. Computer vision studies have demonstrated that camera-derived motion information can be used to characterize hand movement and tremor.[11,12] In the present work, tremor-related movement is treated as a recognition-support concern rather than as a clinical measurement or treatment. The implemented runtime uses temporal stabilization based on 20 consecutive identical predictions; a moving-average filter can be implemented in the future for detecting tremor movements, as it aligns with the lightweight, real-time and low-latency requirements of the system. The proposed system integrates hand-landmark extraction, skeleton-based CNN classification, temporal stabilization, and AR-style visual feedback in one live workflow. The implementation uses a conventional webcam and a lightweight local processing pipeline so that the system can be demonstrated on a standard laptop.

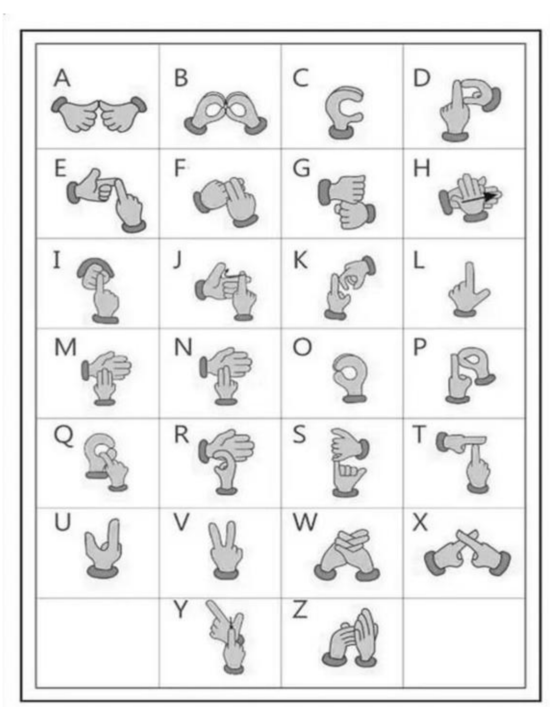

Fig. 1: Indian Sign Language alphabet chart

1.1 Main contributions

- A 26-class ISL alphabet recognition pipeline using 21-point MediaPipe hand landmarks and 128 × 128 skeleton representations. [4,13]
- A CNN-based classifier trained on 13,000 project images and evaluated using a reproducible 80:20 stratified split with random_state = 42.
- Quantitative reporting of accuracy, macro precision, macro recall, macro F1 score, end-to-end latency, FPS, and processing-stage latency.
- A runtime temporal-stability mechanism requiring 20 consecutive identical predictions before a letter is appended, reducing short-lived prediction fluctuations during live interaction.
- Integration of recognition output with the project's Unity/AR-style interface and visual runtime feedback.

The ISL alphabet used as the basis for the 26-class recognition task in the current implementation is presented in Fig. 1.

2. Literature review

Research on sign language recognition has moved from handcrafted image-processing methods to machine learning and deep learning approaches. Earlier ISL work has demonstrated the use of computer vision and MediaPipe-based landmark extraction for recognizing hand configurations.[4] More recent work has examined real-time ISL translation using combinations of RGB information, pose estimates, convolutional networks, and temporal models. For example, Geetha et al. proposed SignFlow, which combines CNN features with transformer-based temporal modeling for continuous ISL recognition and specifically addresses real-time processing across different frame rates.[5] Skeleton and landmark representations are also widely investigated because they describe hand or body structure without the need for a classifier to process every pixel in the original image. Deng et al. proposed a skeleton-based multifeature learning method and reported that skeleton information can be used for sign language recognition while reducing the dependence on raw RGB appearance.[6] This finding supports the use of hand-landmark representations in systems where computational efficiency and robustness to background variation are important. Recent reviews have shown that sign language interpretation systems commonly combine computer vision, machine learning, deep learning, and text or speech output. Najib's review highlights the continuing need to connect research prototypes with practical interpretation functions, including real-time processing and text-to-speech support.[14] A 2025 survey by Violet and Leena Sri further discussed vision-based, sensor-based, and data-driven sign language recognition approaches and the importance of dataset diversity and generalizability.[9] A broader review by Tao et al. similarly surveyed traditional and deep learning approaches, datasets, and challenges in sign language recognition, situating vision-based recognition within the wider research landscape.[15] Tremor analysis is a related but distinct research area. Computer-vision methods have been investigated for estimating and analyzing hand tremors without physical contact.[11,12] These studies motivate the use of landmark trajectories and temporal processing when hand motion involves unwanted movement. The present work does not attempt to diagnose Parkinson's disease; instead, the evaluated implementation uses a confidence threshold and a 20-consecutive-identical-prediction temporal-stability rule to reduce short-lived recognition fluctuations. A moving-average filter was discussed in the earlier project description but is not present in the evaluated implementation. AR has also been explored as a way of providing visual feedback during sign language interactions.[16] The gap addressed by the present project is therefore not the invention of any individual component. Rather, the contribution is the integration of ISL alphabet recognition, landmark-based preprocessing, temporal stabilization, and AR-style output in a single lightweight workflow intended for live camera interaction. The current work is an implementation-focused prototype, and its quantitative performance still requires more detailed benchmarking.

2.1 Summary of related work

To position the proposed framework within the current research landscape, a comparative overview of foundational literature is provided in Table 1.

3. Methodology

3.1 System design and architecture

We designed the system as a sequence of modules connected through the live camera stream. The user first accesses the application through a login interface. The interface includes user name-password authentication and face-login support using OpenCV. After successful access, the user is directed to the camera interface. The login path used before the camera interface is opened is shown in Fig. 2. The camera-side workflow used to obtain live ISL gesture input is shown in Fig. 3.

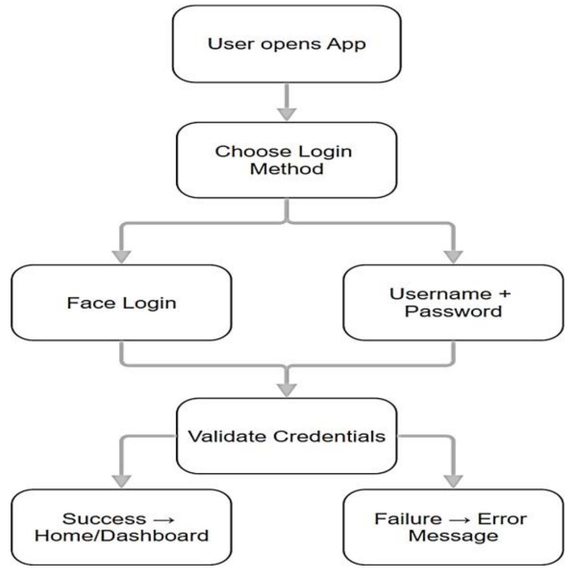

Fig. 2: Login and authentication workflow

The webcam stream is processed with MediaPipe Hands. MediaPipe detects the hand and estimates 21 three-dimensional landmarks for each detected hand. The system uses these landmarks rather than the complete RGB image as the main representation for recognition. This reduces the dependence on background appearance, clothing, illumination, and skin-tone variation. The extracted landmark information is converted into a skeleton-based representation and resized to 128 × 128 pixels. The dataset is organized into 26 class folders corresponding to the ISL alphabets A–Z.

Table 1. Methodological summary and relevance of closely related literature

Study	Main approach	Focus	Relevance to this work
Velmathi & Goyal[4]	MediaPipe-based recognition	ISL gesture recognition	Supports landmark-based ISL recognition.
Singhal et al. [1]	Machine-learning-based ISL detection	Real-time translation	Shows the continuing importance of real-time ISL systems.
Friedrich et al. [11]	Computer-vision tremor analysis	Video-based tremor measurement	Motivates vision-based handling of tremor movement.
Bungay et al. [12]	Contactless hand tremor measurement	Hand tremor analysis	Supports camera-based tremor assessment concepts.
Nunnari et al. [16]	AR sign-language interaction	3D/AR feedback	Supports AR as an interaction and feedback mechanism.
Geetha et al. [5]	CNN + Transformer + pose/RGB	Continuous ISL recognition	Recent real-time ISL benchmark and methodology.
Deng et al. [6]	Skeleton-based multifeature learning	Sign language recognition	Supports use of skeleton/landmark information.
Najib [14]	Research review	ML/AI sign interpretation	Highlights integration and practical interpretation requirements.
Violet & Leena Sri [9]	Comprehensive survey	Recent SLR methods	Provides current research context and challenges.

https://doi.org/10.64189/bse.26803

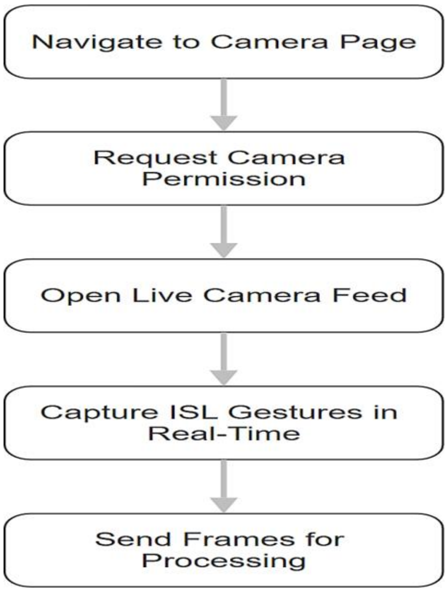

Fig. 3: Camera interface for capturing ISL gestures

Examples of the skeleton representation used to preserve the spatial configuration of the hand while reducing irrelevant background information are shown in Fig. 4.

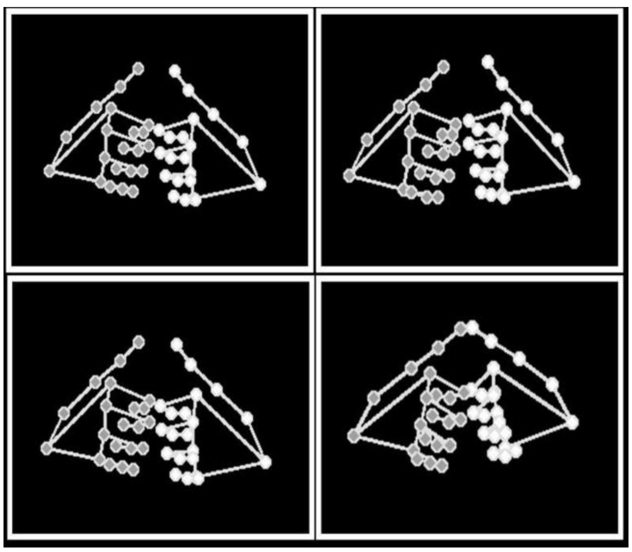
Fig. 4: Skeleton-based gesture representation

The project dataset contains 500 images for each alphabet class. Therefore, the complete dataset contains 26 × 500 = 13,000 images. For the quantitative evaluation reported in this revision, a stratified 80:20 train test split with random_state = 42 was used, corresponding to 10,400 training images and 2,600 held-out test images. No separate validation set was created; the held-out test set was supplied as validation data during model training.

3.1.1 Dataset and split

A comprehensive breakdown of the dataset architecture, total volume, and explicit distribution subsets is detailed in Table 2.

3.2 Experimental environment

Hardware: 12th Gen Intel Core i7-1265U @ 1.80 GHz (10 cores/12 logical processors), 16 GB RAM, Intel UHD integrated graphics with no dedicated GPU, and a built-in laptop webcam at 640 × 480. Software: Windows 11, Google Chrome, Python 3.11.9, TensorFlow 2.15.0, MediaPipe 0.10.9, OpenCV 4.8.0.74, NumPy 1.26.4, and scikit learn 1.3.2.

3.3 Operational workflow

The operational workflow begins with the camera input and ends with the visual output. First, the webcam captures the user's hand. MediaPipe Hands detects hand landmarks and provides a 21-point representation. The landmark representation is then prepared as the model input. The CNN produces a class prediction, and the runtime accepts a prediction only when its confidence is at least 0.85. Temporal stabilization then requires the same predicted letter to be observed for 20 consecutive detections before that letter is appended to the output sequence. The recognized result is displayed through the project's AR-style interface. This stabilization rule is a decision-level temporal mechanism; the evaluated implementation at present does not contain a moving-average landmark filter. The complete processing sequence from camera input to recognition, temporal stabilization, AR-style output, and optional text-to-speech output is summarized in Fig. 5.

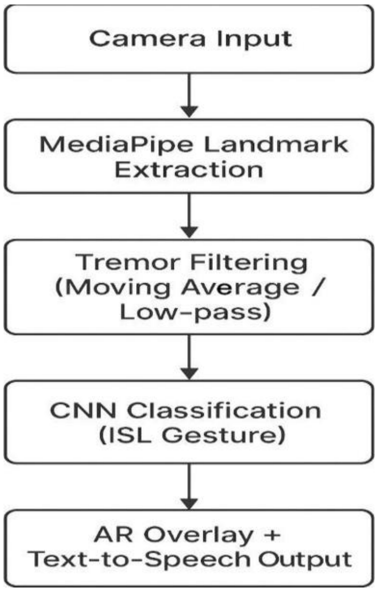

Fig. 5: End-to-end workflow showing gesture capture, temporal stabilization, recognition, and AR-style feedback

Table 2: Distribution matrix and structural breakdown of the ISL alphabet dataset

Item	Value	Details
Number of classes	26	ISL alphabet A–Z
Images per class	500	Classwise image samples
Total images	13,000	26 × 500
Training split	80%	10,400 images
Testing split	20%	2,600 held-out images
Validation split	None	No separate validation set; held-out test set used as validation_data during training

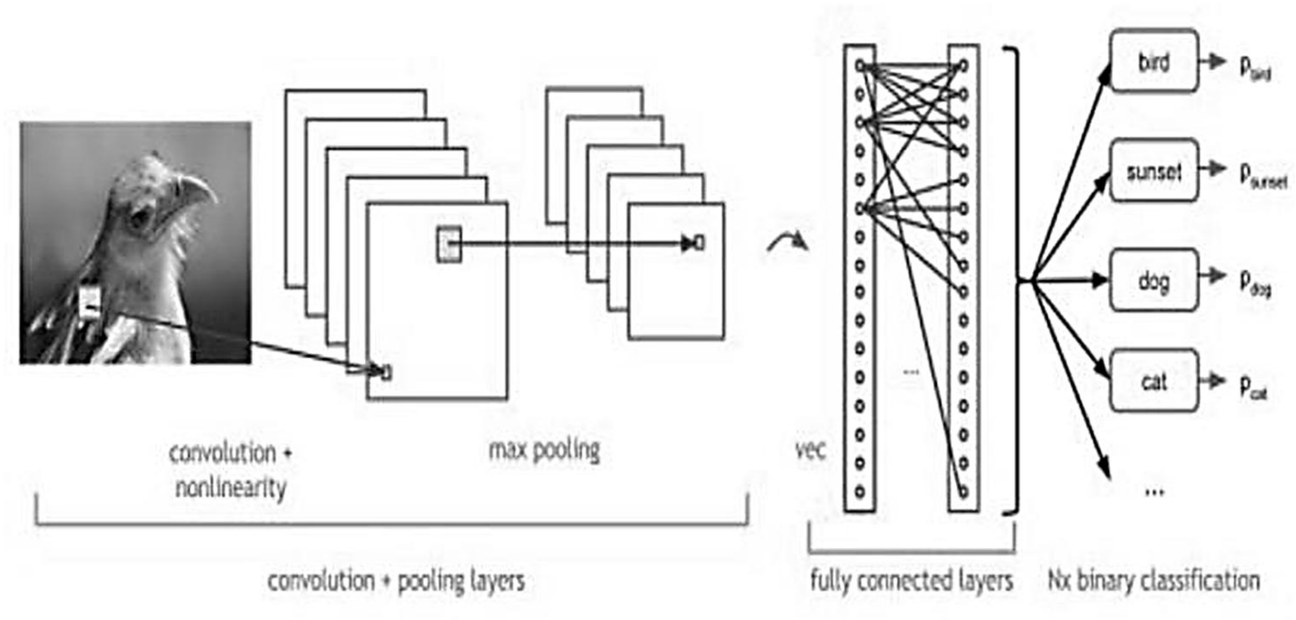

Fig. 7: Convolutional neural network concept used for classification

3.4 ISL recognition, tremor correction, and AR integration

The recognition component uses the spatial relationship between the detected hand landmarks. The prepared skeleton representation is supplied to a lightweight CNN implemented with TensorFlow 2.15.0. The model is intended to recognize the 26 alphabet classes represented in the project dataset. The use of landmark information allows the system to focus on hand geometry rather than processing the full camera frame. Tremor-related movement is handled at the recognition decision stage rather than through a separate moving-average filter. The evaluated implementation uses a confidence threshold of 0.85 and requires 20 consecutive identical predictions before a letter is appended. This rule is intended to suppress short-lived prediction fluctuations during live interactions. Future work can evaluate adaptive low-pass filters such as the Kalman filter or 1€ filter.[17] After classification, the recognized output is placed over the live camera view using our project's AR-style interface. The AR-style overlay in the working prototype is described using JavaScript and the Canvas API for displaying recognized output over the live camera view. The working prototype includes image recognition, shaky-movement/tremor handling, and AR support. The cross-platform mobile extension shown in Fig. 6 is treated as an extension of the design rather than as evidence of a separately benchmarked Android or iOS deployment. The planned route for extending the web-based interface to the Android and iOS platforms while retaining the same overall processing workflow is shown in Fig. 6. The CNN-based classification concept after the hand representation has been prepared is shown in Fig. 7.

4. Results and discussion

The results combine reproducible quantitative evaluation with live prototype observations. The quantitative evaluation uses the 13,000-image project dataset with a stratified 80:20 train–test split and random_state = 42. The held-out test set contains 2,600 images. Live performance was measured separately over 100 webcam frames on the stated laptop hardware.

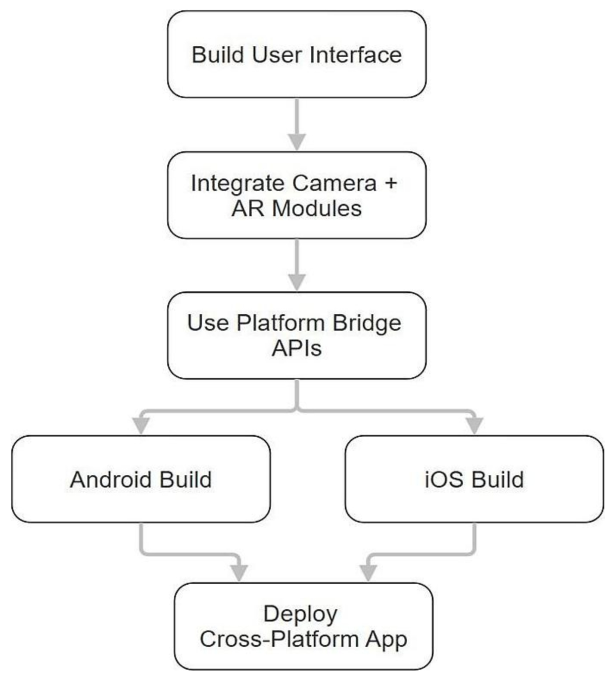

Fig. 6: Cross-platform support and future mobile extension

4.1 Classification performance

The final CNN evaluation yielded a 99.27% overall accuracy (Table 3). The macro precision, macro recall, and macro F1 score were 99.27%. Because all 26 classes contribute equally to the held-out set, the macro measures provide a balanced summary of class-level performance.

Table 3: Core classification metrics and validation capacity of the evaluated CNN

Performance metric	Result
Overall Accuracy	99.27%
Macro Precision	99.27%
Macro Recall	99.27%
Macro F1-score	99.27%
Held-out test set	2,600 images (20%)

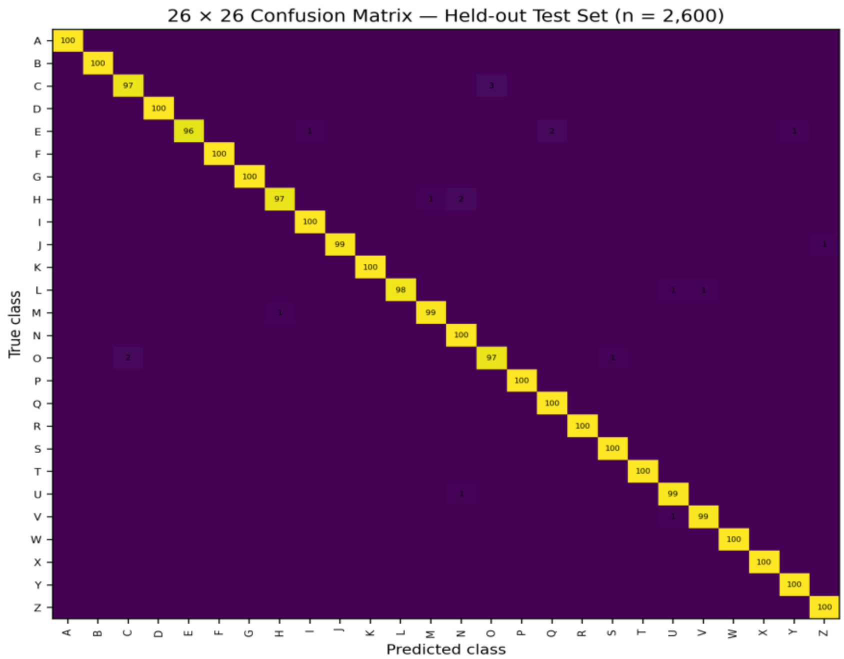

Fig. 8: 26 × 26 confusion matrix for the held-out 2,600-image test set

4.2 Confusion matrix and classwise performance

The complete 26 × 26 confusion matrix is shown in Fig. 8. The diagonal contains 2,581 correct predictions out of 2,600 held-out test images, matching the reported 99.27% accuracy. The 19 errors are concentrated in visually similar configurations: C→O (3), E→I (1), E→Q (2), E→Y (1), H→M (1), H→N (2), J→Z (1), L→U (1), L→V (1), M→H (1), O→C (2), O→S (1), U→N (1), and V→U (1).

Fig. 9: Runtime interface before sign detection

4.3 Real-Time performance

Real-time performance was benchmarked over 100 webcam frames (Table 4). The average end-to-end processing time was 108.40 ms/frame, corresponding to approximately 9.23 FPS. MediaPipe landmark processing accounted for 38.30 ms/frame, and CNN inference accounted for 69.73 ms/frame. These are processing-stage timings, not a filtering-versus-classification comparison. The benchmark was performed locally and excludes network transfer, JPEG encoding, Unity rendering, and text-to-speech latency.

Fig. 10: Runtime interface state before live sign recognition

4.4 Shaky/Tremor-Motion evaluation

Shaky-motion testing was performed qualitatively using the live webcam interface. The following screenshots provide direct visual evidence from the working prototype, including interface states before detection (Fig. 9, Fig. 10), skeleton extraction (Fig. 11), Unity simulator output during recognition (Fig. 12, Fig. 13, Fig. 14), and runtime/API prediction logs (Fig. 15).

Table 4: Computational latency and real-time processing benchmarks of the framework

Metric	Result	Measurement basis
Benchmark frames	100	Live webcam frames
End-to-end latency	108.40 ms/frame	Average total local processing time
Frame rate	9.23 FPS	Derived from average frame time
MediaPipe processing	38.30 ms/frame	Hand/landmark processing
CNN inference	69.73 ms/frame	Classification inference
Temporal stabilization	20 consecutive identical predictions	Decision rule; not separately timed

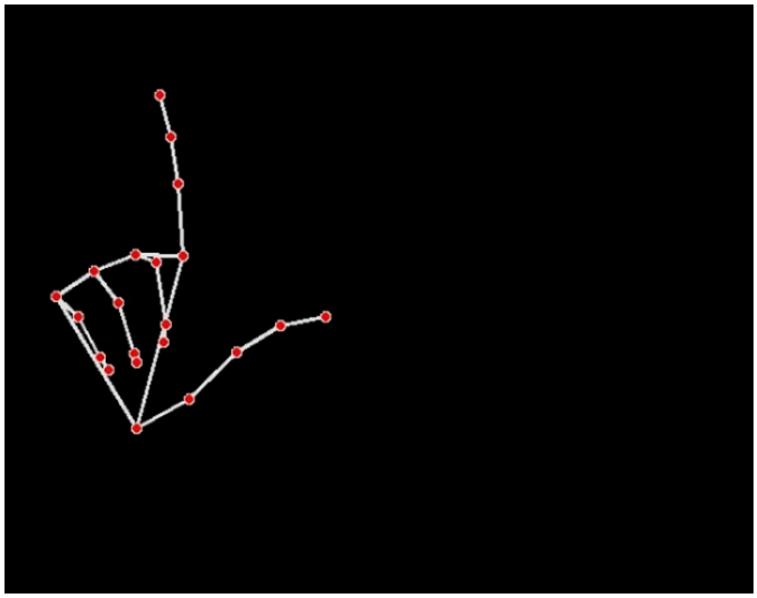

Fig. 11: Extracted hand skeleton representation during live processing

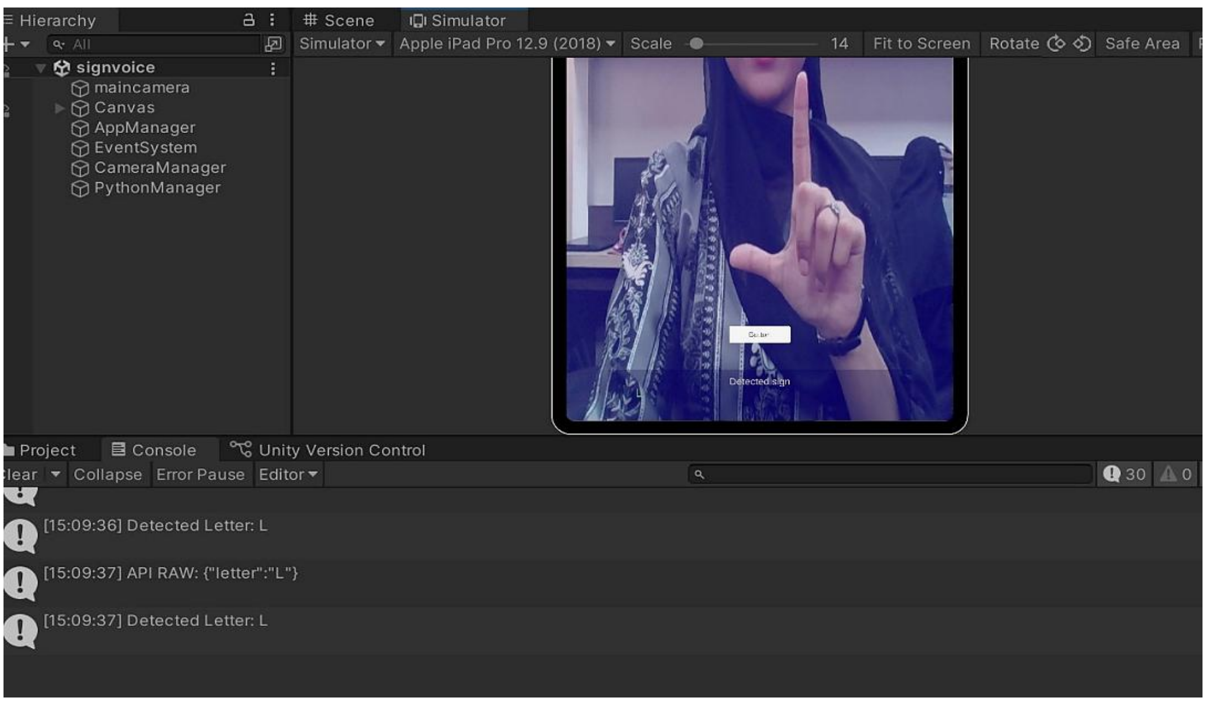

Fig. 12: Unity simulator showing live ISL gesture recognition and detected output

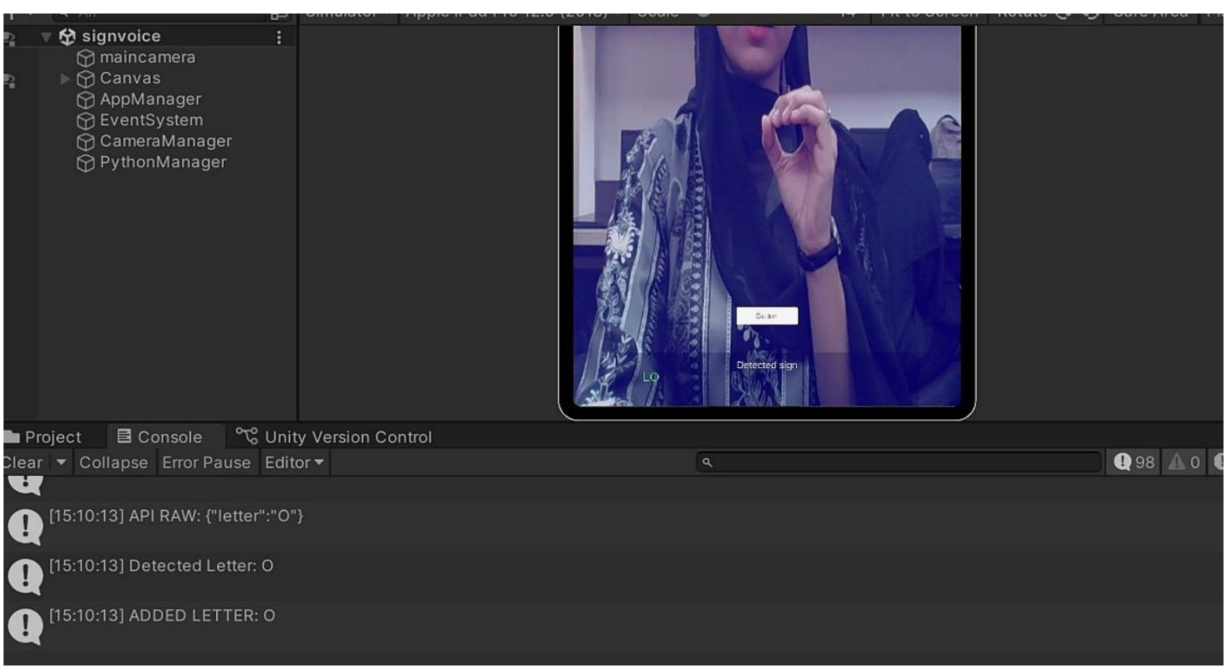

Fig. 13: Unity simulator showing recognition of another ISL hand configuration and console output

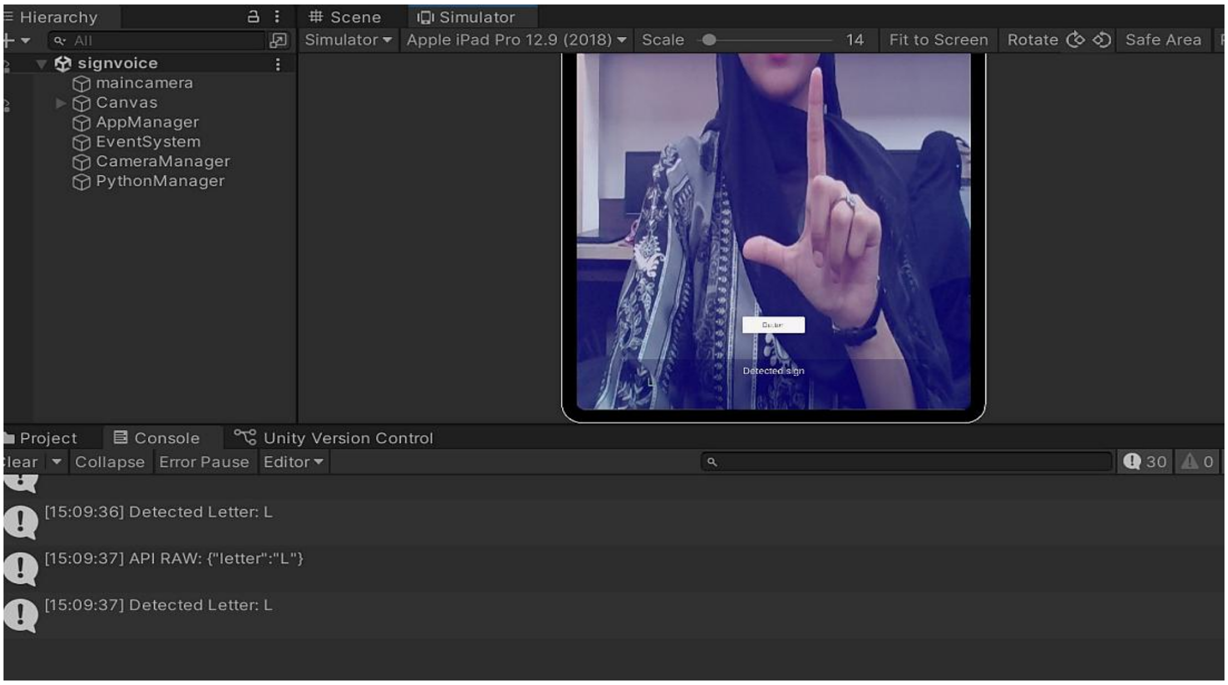

Fig. 14: Unity simulator showing live sign recognition and the corresponding console output

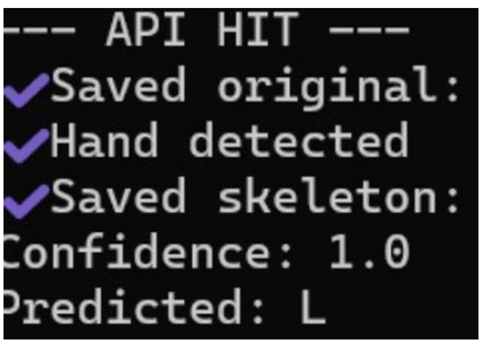

Fig. 15: Runtime API log showing successful hand detection, skeleton generation, confidence, and predicted letters

The screenshots support the functional integration of camera capture, landmark extraction, prediction, API communication, and AR-style display. They do not establish clinical tremor correction or a quantitative filtering benefit. The 20-consecutive-identical-prediction rule is a decision-level temporal stabilization mechanism and may introduce delay when a user changes signs rapidly.

4.5 Discussion

The quantitative results indicate strong classification performance on the held-out test set, with only 19 misclassified images. The confusion matrix shows that most classes were recognized without error and that the remaining errors are localized to a small set of visually similar configurations. The live benchmark shows that CNN inference is the larger component of the measured processing time (69.73 ms/frame), while MediaPipe landmark processing accounts for 38.30 ms/frame. The temporal stabilization mechanism is lightweight and easy to deploy, but it has several limitations. Requiring 20 consecutive identical predictions can suppress short-lived fluctuations but can also delay the acceptance of a legitimate sign change. It is not equivalent to smoothing landmark coordinates and does not quantify tremor amplitude. Future work should compare this approach with adaptive filters such as the Kalman filter and 1€ filter,[17] using controlled tremor protocols and reporting recognition accuracy, latency, and responsiveness. The results should also be interpreted in the context of the experimental setup: a conventional laptop webcam, integrated graphics, and local software execution. The benchmark is therefore a prototype-level measurement rather than a universal hardware-independent real-time guarantee.

5. Conclusion

This paper presents an integrated assistive system for Indian Sign Language recognition with temporal stabilization and AR-style feedback. The implementation uses MediaPipe hand landmarks, a skeleton-based representation, a CNN classifier implemented with TensorFlow 2.15.0, and live visual output. The project dataset contains 13,000 images covering 26 ISL alphabet classes. Quantitative evaluation using a stratified 80:20 split with random_state = 42 produced 2,600 held-out test images and 99.27% accuracy, precision, recall, and F1 score. Live benchmarking over 100 webcam frames produced 108.40 ms/frame, or approximately 9.23 FPS, with 38.30 ms/frame attributed to MediaPipe processing and 69.73 ms/frame to CNN inference. The evaluated runtime does not contain the moving-average filter. Instead, it uses a 0.85 confidence threshold and a 20-consecutive-identical-prediction temporal-stability rule. Shaky-motion evaluation was qualitative, so no unsupported numerical tremor-reduction claim was made. Future work should include controlled tremor experiments, adaptive filtering such as Kalman or 1€ filtering, broader datasets, and measured deployment performance. The system is intended as an assistive recognition prototype and not as a medical treatment or diagnostic system.

Acknowledgments

We would like to express our sincere gratitude to the ICT

Academy for providing us with the opportunity to work on the project titled "Real-Time ISL to Text Interpretation and Parkinson's Tremor Correction with AR support" as a part of our academic curriculum. We are deeply indebted to our respected guide, Er. Farhana Siddiqui, whose constant support, valuable suggestions, and insightful feedback played a crucial role in the completion of this project. We also thank the faculty members and staff of the Department of Computer Engineering, M.H. Saboo Siddik College of Engineering, for providing the necessary resources and academic environment.

CRediT Author Contribution Statement

Farhana Siddiqui: Conceptualization, Formal analysis, Methodology, Project administration, Supervision. **Iqra Ansari:** Formal analysis, Software, Visualization. **Nusaybah Kazi:** Data Curation, Investigation, Software, Validation. **Jamil Khatri:** Formal analysis, Software, Methodology, Writing- Original draft. All authors have read and approved the final version of the manuscript for publication and agree to be accountable for all aspects of the work, ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement

The source code datasets generated and/or analyzed during the current study that support the findings are available from the corresponding author upon reasonable request.

Conflict of Interest

There is no conflict of interest.

Artificial Intelligence (AI) Use Disclosure

The authors declare that artificial intelligence (AI)-assisted tools were used only for language refinement, grammar improvement, and manuscript structuring purposes during the preparation of this work. All technical content, experimental implementation, results, and interpretations were independently developed and verified by the authors.

Supporting Information

Not applicable.

References

[1] G. Velmathi, K. Goyal, Indian sign language recognition using MediaPipe Holistic, arXiv:2304.10256, 2023, doi: 10.48550/arXiv.2304.10256.

[2] R. Singhal, J. Gupta, A. Sharma, A. Gupta, N. Sharma, Indian sign language detection for real-time translation using machine learning, arXiv:2507.20414, 2025, doi: 10.1109/RAIT65068.2025.11089142.

[3] M. U. Friedrich, A. J. Roenn, C. Palmisano, J. Alty, S. Paschen, G. Deuschl, C. W. Ip, J. Volkmann, M. Muthuraman, R. Peach, M. M. Reich, Validation and application of computer vision algorithms for video-based tremor analysis, *npj Digital medicine*, 2024, **7**, 165, doi: 10.1038/s41746-024-01153-1.

[4] J. Bungay, O. Emokpae, S. D. Relton, J. Alty, S. Williams, H. Fang, D. C. Wong, Contactless hand tremor amplitude measurement using smartphones: Development and pilot evaluation, arXiv: 2304.14937v1, 2023, https://arxiv.org/html/2304.14937v1, Accessed 20 July 2026.

[5] F. Nunnari, E. Avramidis, V. Yadav, A. Pagani, Y. Hamidullah, S. Mollanorozy, C. España-Bonet, E. Woop, P. Gebhard, Towards incorporating 3d space-awareness into an augmented reality sign language interpreter. In2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), IEEE, 2023, 1-5.

[6] M. Geetha, N. Aloysius, D. A. Somasundaran, A. Raghunath, P. Nedungadi, Toward real-time recognition of continuous Indian Sign Language: A multimodal approach using RGB and pose, *IEEE Access*, 2025, **13**, 60270–60283, doi: 10.1109/ACCESS.2025.3554618.

[7] Z. Deng, Y. Leng, J. Hu, Z. Lin, X. Li, Q. Gao, SML: A Skeleton-based multi-feature learning method for sign language recognition, *Knowledge-Based Systems*. 2024, **301**, 112288, doi: 10.1016/j.knosys.2024.112288.

[8] F. M. Najib, Sign language interpretation using machine learning and artificial intelligence, *Neural Computing and Applications*, 2025, **37**, 841-857, doi: 10.1007/s00521-024-10395-9.

[9] I. M. Violet, R. L. Sri, A comprehensive survey on recent advances and challenges in sign language recognition systems, *Discover Artificial Intelligence*, 2025, **5**, 419, doi: 10.1007/s44163-025-00629-7.

[10] T. Tao, Y. Zhao, T. Liu, J. Zhu, Sign language recognition: A comprehensive review of traditional and deep learning approaches, datasets, and challenges, *IEEE Access*, 2024, **12**, 75034-75060, doi: 10.1109/ACCESS.2024.3398806.

[11] H. K. Al-Mohair, O. Al-Qershi, Video-based sign language recognition in the deep learning era: representations, temporal modeling, alignment, benchmarks, and open challenges, *The Visual Computer*, 42, **457**, 2026, doi: 10.1007/s00371-026-04665-x.

[12] T. Sembayev, D. Akbarov, Recent advancements in skeleton-based sign language recognition, *Bulletin of Abai KazNPU. Series of Physical and Mathematical Sciences*, 2025, **92**,

184–199, doi: 10.51889/2959-5894.2025.92.4.018.

[13] C. Arya, A. Gusain, K. Kumar, M. Diwakar, I. Gupta, N. K. Pandey, A Lightweight Solution for Real-Time
Indian Sign Language Recognition, In 2024 International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit), 2024, IEEE, doi: 10.1109/GlobalAISummit62156.2024.10947786.

[14] R. Damdoo, P. Kumar, An integrative survey on Indian sign language recognition and translation, *IET Image Processing*, 2025, 19, e70000, doi: 10.1049/ipr2.70000.

[15] Y. Min, Y. Yang, P. Jiao, Z. Nan, X. Chen, A closer look at skeleton-based continuous sign language recognition, In 2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), IEEE, 2025, 4968-4974.

[16] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. L. Chang, M. G. Yong, J. Lee, W. T. Chang, Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172, 2019, doi: 10.48550/arXiv.1906.08172.

[17] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. L. Chang, M. G. Yong, J. Lee, W. T. Chang, Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172, 2019, doi: 10.48550/arXiv.1906.08172.