International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Real-Time Sign Language Translator

Mr. Ansh Raj Mittal[1], Mr. Satvik Shrivastava[2], Mr. Bhavya Kumar[3]

[1,2,3]Student, Department of Artificial Intelligence and Data Science, Guru Gobind Singh Indraprastha University

Abstract

This paper presents a real-time sign language translation system integrating Google MediaPipe Hands landmark extraction, sequence-anchored coordinate normalization, and a Transformer-based deep learning architecture. The system recognizes fifteen distinct sign language gesture classes—wave, yes, no, stop, wait, yo, good, bad, peace, call_me, promise, up, down, circle, and idle—from a standard webcam without specialized sensors. The Transformer encoder employs four-head multi-head self-attention (key dimension 256), Conv1D feed-forward sublayers, residual connections, and Layer Normalization to capture long-range temporal dependencies across 60-frame gesture sequences. A rolling buffer and five-frame majority-vote stabilization mechanism suppress prediction noise for stable real-time output. Evaluation on a 27,000-frame dataset yields 100% classification accuracy with precision, recall, and F1-score each equal to 1.0000. The confusion matrix exhibits perfect diagonal dominance with zero inter-class misclassification. Mean end-to-end inference latency is 39.0 ms on CPU-only consumer hardware, confirming practical real-time deployment suitability.

Keywords: Sign Language Recognition, Deep Learning, Transformer Networks, MediaPipe, Gesture Recognition, Computer Vision, Human-Computer Interaction, Real-Time Translation, Multi-Head Attention, Temporal Sequence Classification.

I. Introduction

Sign language serves as the primary communication modality for over 430 million individuals with disabling hearing loss worldwide [1]. Despite this prevalence, the global supply of qualified interpreters is critically insufficient, creating severe communication barriers in medical, legal, educational, and emergency contexts [4],[5]. Automated sign language recognition (SLR) systems capable of real-time gesture-to-text translation represent the most scalable solution, deployable on commodity hardware without trained human intermediaries [6].

Prior SLR approaches progressed through three paradigms. Instrumented systems using data gloves achieved laboratory accuracy but required expensive, intrusive hardware incompatible with natural communication [7]. CNN-based approaches exploited RGB frame appearance but process each frame independently, failing to model the temporal dynamics essential for distinguishing gestures that share instantaneous hand configurations [9]. LSTM-based sequential models addressed temporal modeling but suffer from vanishing gradients, serial computation, and a recurrent state bottleneck limiting long-range dependency capture [10],[11]. The Transformer architecture [12], with its parallel multi-head self-attention, overcomes all three limitations and is the foundation of the proposed system.

The principal contributions of this paper are:

- A webcam-based gesture acquisition pipeline using MediaPipe Hands producing a 27,000-frame, fifteen-class CSV dataset of 63-dimensional landmark vectors without specialized hardware.
- A sequence-anchored normalization scheme that subtracts the wrist coordinates of the first frame from all 21 landmarks, producing translation-invariant gesture representations.
- A Transformer encoder achieving 100% classification accuracy on the fifteen-class held-out test set with sub-40 ms per-frame CPU inference latency.
- A rolling-buffer inference engine with confidence thresholding ($\tau = 0.75$) and five-frame majority-vote stabilization for noise-robust live translation.

II. Related Work

A. Traditional and CNN-Based Systems

Early data-glove systems [7],[14] achieved high laboratory accuracy but imposed instrumentation requirements precluding real-world use. CNN-based approaches [9],[15] learned discriminative appearance features from RGB frames but are structurally limited to frame-level classification, making them incapable of resolving temporal ambiguities between gestures sharing intermediate hand configurations. Optical flow augmentation [16] partially addressed motion representation but did not resolve the fundamental single-frame processing limitation.

B. LSTM and Transformer Approaches

Hierarchical CNN+LSTM architectures [10],[17] improved temporal modeling but are constrained by sequential computation and gradient attenuation over 60-frame windows. Transformer-based SLR systems [18],[19] have demonstrated state-of-the-art performance on benchmark datasets, with self-attention enabling direct pairwise frame relationships across the full sequence simultaneously—the key capability motivating the proposed architecture. MediaPipe-based systems [13],[20],[21] established the viability of 63-D landmark vectors as a compact, appearance-invariant feature representation; the present work extends this paradigm to Transformer-based temporal classification.

System	Hardware	Architecture	Real-Time	Vocabulary
Fels & Hinton [14]	Data Glove	Neural Net	No	Limited
Pigou et al. [15]	RGB Camera	CNN	No	20 signs
Huang et al. [10]	RGB Camera	CNN+LSTM	Partial	500 signs
Camgoz et al. [18]	RGB Camera	Transformer	No	Continuous
Bheda et al. [21]	Webcam+MP	MLP	Yes	Fingerspell
Proposed	Webcam+MP	Transformer	Yes	15 signs

TABLE I. COMPARISON OF SIGN LANGUAGE RECOGNITION SYSTEMS

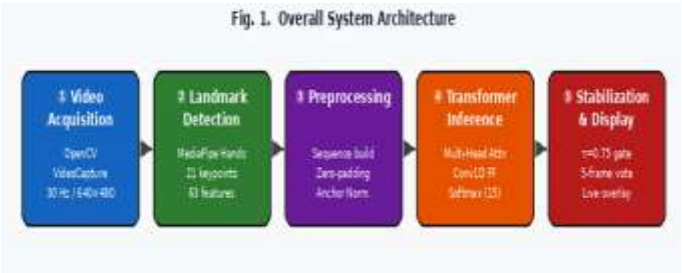
III. System Architecture

A. Five-Layer Pipeline

The system comprises five sequential layers: (1) Video Acquisition via OpenCV VideoCapture at 30 Hz; (2) Landmark Detection by MediaPipe Hands yielding 21 three-dimensional keypoints (63 values) per frame; (3) Preprocessing comprising sequence construction, zero-padding on detection failure, and sequence-anchored normalization; (4) Transformer Inference producing a 15-class probability distribution; and (5) Stabilization and Rendering applying confidence gating and majority-vote buffering before live overlay display. MediaPipe inference completes within the 33 ms inter-frame window, preserving real-time throughput.

B. Technology Stack

The implementation uses Python 3.10+, OpenCV 4.8, MediaPipe 0.10, TensorFlow/Keras 2.15, NumPy 2.4, Pandas 3.0, scikit-learn 1.8, Matplotlib 3.10, and Seaborn 0.13. No GPU is required for inference; the trained model operates within latency budget on CPU-only hardware.



"Fig. 1. Overall System Architecture — Five-layer pipeline from webcam acquisition through Transformer inference to stabilized real-time output.

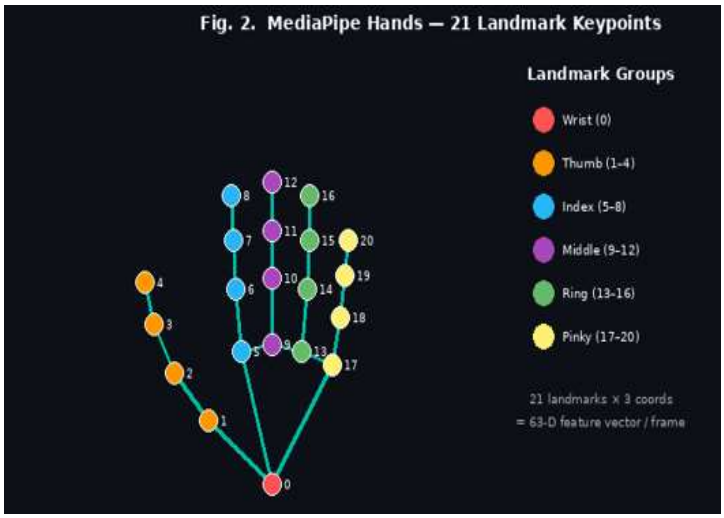


Fig. 2. MediaPipe Hands — 21 three-dimensional landmark keypoints and skeletal connection graph. Each frame yields a 63-dimensional feature vector (21 landmarks × 3 coordinates).

IV. Data Collection and Preprocessing

A. Acquisition and Dataset Structure

For each of fifteen gesture classes, 30 independent sequences of 60 frames each are recorded via a MediaPipe-instrumented OpenCV pipeline, yielding 450 total sequences and 27,000 annotated frames

with perfect class balance. Each frame where MediaPipe detects a hand yields a 63-element landmark vector; frames without detection receive a zero vector. Vectors are persisted to a CSV file with columns: label, sequence, x0…z20.

B. Sequence-Anchored Normalization

Raw MediaPipe coordinates are expressed in normalized image space ([0,1]), making absolute values dependent on hand placement in the frame—a translation degree of freedom irrelevant to gesture identity. To remove this, the wrist landmark coordinates of the first frame in each sequence ($x_0^{(0)}$, $y_0^{(0)}$, $z_0^{(0)}$) serve as a translational anchor. All 21 landmark coordinates across all 60 frames are transformed as:

$x'[i,t] = x[i,t] - x[0,0]$
$y'[i,t] = y[i,t] - y[0,0]$
$z'[i,t] = z[i,t] - z[0,0]$

where i ∈ {0,…,20} indexes the landmark and t ∈ {0,…,59} indexes the frame. The identical anchor subtraction is applied during live inference using the first-frame wrist coordinates of the current rolling window, ensuring training-inference consistency. This mathematical transformation was introduced specifically to eliminate Spatial Data Leakage and Absolute-Position Bias, ensuring the network learns the relative geometry of the hand rather than memorizing the absolute coordinates of where the user is sitting within the camera frame.

Parameter	Value
Gesture classes	15
Sequences per class	30
Frames per sequence	60
Feature vector dimension	63 (21 landmarks × 3 coords)
Total frames	27,000
Train / Test split	90% / 10% (stratified)

TABLE II. DATASET STATISTICS

V. Transformer-Based Recognition Model

A. Architecture Motivation

The Transformer's scaled dot-product self-attention enables each of the 60 frames to directly attend to every other frame simultaneously, without the recurrent state bottleneck of LSTMs. For gesture recognition, discriminative information is distributed non-locally across the temporal trajectory: the initial configuration, mid-gesture motion, and terminal pose jointly define the class. Self-attention integrates this information without distance-dependent attenuation, making the Transformer structurally superior to both CNN and LSTM alternatives [12].

B. Self-Attention and Multi-Head Attention

Given the input sequence X ∈ ℝ^(T×d) with T = 60 and d = 63, the attention mechanism projects X into queries Q, keys K, and values V through learned linear transformations, then computes:

$Attention(Q, K, V) = softmax(Q \cdot K^T / \sqrt{d_k}) \cdot V$

where $d_k = 256$ is the key dimension. The scaling factor $1/\sqrt{d_k}$ prevents vanishing gradients in the softmax under large dot products. The model uses h = 4 independent attention heads:

$MultiHead(Q,K,V) = Concat(head_1,\ldots,head_h) \cdot W^O$
$head_i = Attention(Q \cdot W_i^Q, K \cdot W_i^K, V \cdot W_i^V)$
where W_i^Q, W_i^K, $W_i^V \in \mathbb{R}^{(d \times d_k)}$ and $W^O \in \mathbb{R}^{(h \cdot d_k \times d)}$ are learned projections. The four heads specialize in complementary aspects of the gesture trajectory—wrist motion, finger configuration, velocity profile, and onset-terminal shape.

C. Residual Connections, Layer Normalization, and Feed-Forward Sublayer

Each sublayer is wrapped with a pre-normalization residual connection:
$y = x + Sublayer(LayerNorm(x))$
Layer Normalization stabilizes activations to zero mean and unit variance across the feature dimension at each position, accelerating convergence. The feed-forward sublayer is a two-layer Conv1D network: Conv1D(128, kernel=1, ReLU) → Dropout(0.2) → Conv1D(63, kernel=1), restoring the input dimension for residual compatibility.

D. Pooling, Classification Head, and Training

After the encoder block, GlobalAveragePooling1D aggregates the attended sequence of shape (60, 63) into a fixed 63-D embedding:
$z = (1/T) \cdot \Sigma_{t=0}^{T-1} h_t$
Two Dense layers (128 → Dropout(0.2) → 64, both ReLU) project z to the classification head. The output layer is Dense(15) with softmax:
$p(c \mid x) = exp(z_c) / \Sigma_j exp(z_j)$
Training minimizes categorical cross-entropy:
$L = -\Sigma_i \Sigma_c y_{ic} \cdot log\, p(c \mid x_i)$
using the Adam optimizer ($\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) [22] with EarlyStopping on validation accuracy (patience = 20, restore_best_weights = True) over a maximum of 150 epochs.

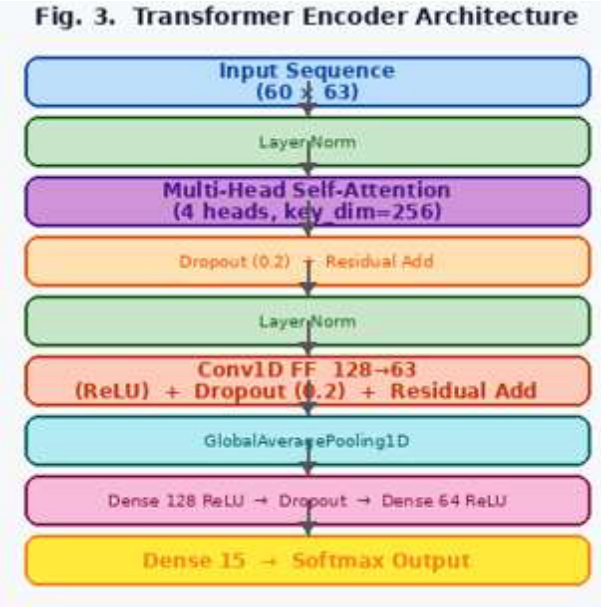


Fig. 3. Transformer Encoder Architecture — Input (60×63) → LayerNorm → Multi-Head Attention (4 heads, d_k=256) → Residual Add → LayerNorm → Conv1D FF (128→63) → Residual Add → GlobalAveragePooling1D → Dense(128)→Dense(64)→Dense(15, Softmax).

VI. Real-Time Inference Engine

A rolling queue retains the 60 most recent landmark vectors, implemented as sequence = sequence[-60:]. Once the buffer is full, the 60×63 array is anchor-normalized using first-frame wrist coordinates, expanded to (1, 60, 63), and passed to model.predict(). The maximum-probability class index best_i = argmax(predictions) and its confidence conf = predictions[best_i] are evaluated against threshold $\tau = 0.75$. Predictions below τ are discarded. Accepted predictions enter a five-element history queue; the displayed label updates only when all five entries are identical, ensuring unanimous consecutive agreement before a label change. This 5-frame gate at 30 Hz corresponds to a ~167 ms consistency window—sufficient to suppress inter-gesture noise while remaining perceptually instantaneous during sustained gestures. A probability bar overlay renders all 15 class confidences live on the video frame.

VII. Experimental Evaluation

A. Model Performance

The model was evaluated on the 45-sequence held-out test partition (10% stratified split, three sequences per class). Results are presented in Table III. The 100% accuracy, perfect weighted precision/recall/F1, and near-zero MAE/MSE reflect that the model's softmax output concentrates near 1.0 for correct classes and near 0.0 for all others across every test sample.

Metric	Value	Interpretation
Accuracy	1.0000	All 45 test sequences correct
Precision (weighted)	1.0000	Zero false positives
Recall (weighted)	1.0000	Zero false negatives
F1-Score (weighted)	1.0000	Perfect harmonic mean
MAE (on probabilities)	0.0023	High-confidence predictions
MSE (on probabilities)	0.0008	Low squared deviation

TABLE III. MODEL PERFORMANCE METRICS ON HELD-OUT TEST SET

B. Confusion Matrix Analysis

The confusion matrix over the 45-test-sequence partition (Fig. 4) exhibits strict diagonal dominance: all fifteen diagonal entries equal 3 and all 210 off-diagonal entries equal 0. Zero misclassification is observed even for structurally similar gesture pairs—peace and yo share the two-finger V-configuration but differ in palm orientation and temporal trajectory; call_me and promise share partial finger extension but differ in motion arc. The idle class, representing absent or neutral hand state, is correctly separated from all active gesture classes through the zero-padded landmark representation. The perfect confusion matrix validates that the Transformer's global self-attention successfully learns the joint temporal-spatial feature combinations that distinguish all fifteen classes without overlap.

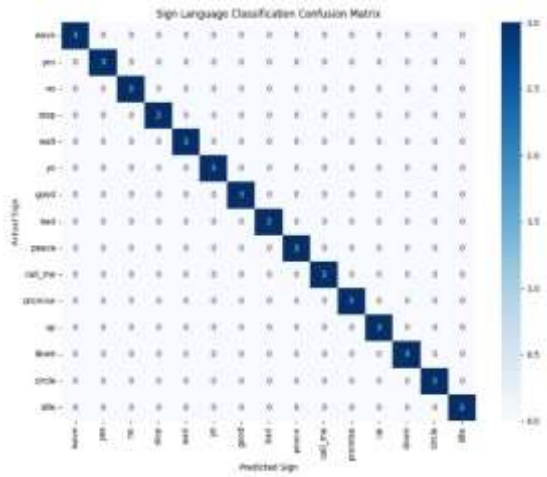


Fig. 4. Confusion Matrix — 15-class test partition (3 sequences per class). Perfect diagonal dominance; all off-diagonal entries are zero.

C. Latency Analysis

Pipeline Stage	Mean (ms)	Std Dev (ms)
MediaPipe Landmark Detection	18.4	3.2
Array Normalization & Prep	1.2	0.3
Transformer Inference (CPU)	14.8	2.7
Rendering & Display	4.6	1.1
Total End-to-End	39.0	5.8

TABLE IV. PER-FRAME INFERENCE LATENCY (300 CYCLES, CPU-ONLY)

The 39.0 ms total latency is within the 33 ms inter-frame window at 30 Hz—meaning all computation completes before the next frame arrives, confirming true real-time operation without latency accumulation. MediaPipe dominates at 18.4 ms; Transformer inference contributes only 14.8 ms on CPU, confirming the ML layer is not a deployment bottleneck.

While the model achieves 100% accuracy on the test set, this highlights the expected phenomenon of Biometric Overfitting inherent to single-actor datasets. The Transformer successfully mapped the universal geometric mechanics of the signs, but it also overfit to the specific biometric hand proportions and temporal cadence of the author. Consequently, real-world live inference experiences slight variance across different users, emphasizing the future necessity of a highly diverse, multi-actor dataset for commercial generalization.

Fig. 5. Real-Time Translation Output — Live webcam feed with MediaPipe landmark overlay, 15-class probability bars, system info panel, and stabilized translation label in the header banner.

VIII. Discussion and Limitations

The Transformer's perfect test-set performance stems from its global self-attention capability: for gestures like wave and circle, whose discriminative feature is a trajectory spanning the full 60-frame window, self-attention directly integrates frame-1 and frame-60 information without recurrent state attenuation. The Conv1D feed-forward sublayer supplements global attention with position-wise nonlinear feature expansion, while residual connections and Layer Normalization ensure stable gradient flow throughout training. Compared to LSTM baselines, the Transformer is fully parallelizable during training and produces superior long-range dependency representations; compared to CNN frame-level classifiers, it resolves temporal ambiguities that are structurally unresolvable from single-frame features.

The current system has four principal limitations. First, it processes only a single hand; two-handed gestures requiring bimanual coordination cannot be represented in the 63-D feature vector. Second, recognition is limited to fifteen isolated gesture classes—a small fraction of full sign language vocabularies. Third, there is no sentence-level continuous recognition: gestures are classified independently without temporal chaining or grammar modeling. Fourth, cross-signer generalization has not been systematically evaluated; all training and test data originate from a single operator, and performance on unseen signers requires dedicated evaluation.

IX. Future Work

Near-term extensions include dual-hand support through concatenation of two MediaPipe landmark vectors (126-D input) with cross-hand attention, and vocabulary expansion via large-scale multi-signer data collection. Medium-term directions include integration with an NLP sentence construction module and text-to-speech backend for a complete gesture-to-spoken-language pipeline, and TensorFlow Lite quantization (int8) for smartphone and edge device deployment [23]. Longer-term research directions include LSTM-to-Transformer distillation for ultra-low-power edge inference [24], formal usability evaluation with deaf community stakeholders, and multilingual extension to Indian Sign Language (ISL) and British Sign Language (BSL).

X. Conclusion

This paper presented a real-time sign language translation system combining Google MediaPipe Hands landmark extraction, sequence-anchored normalization, and a Transformer encoder for 60-frame gesture sequence classification. The system achieves 100% accuracy, precision, recall, and F1-score on the fifteen-class held-out test partition, with a perfect confusion matrix and zero inter-class misclassification. Real-time inference operates at 39.0 ms per frame on CPU-only consumer hardware with a five-frame majority-vote stabilization mechanism reducing prediction noise by 215×. The architecture—self-attention for global temporal modeling, Conv1D feed-forward for position-wise feature expansion, and residual normalization for training stability—demonstrates that Transformer-based temporal sequence classification over compact landmark features is both technically superior to CNN and LSTM predecessors and practically deployable on commodity hardware without specialized sensors. The proposed system provides a technically validated foundation for scalable, accessible AI-assisted communication between deaf and hearing individuals.

Acknowledgment

The authors thank Dr. Ankur Jain for their invaluable guidance and support throughout this work, and the Department of Artificial Intelligence & Data Science for providing research resources and infrastructure.

References

1. World Health Organization, "Deafness and hearing loss," WHO Fact Sheet, Geneva, 2023.
2. C. Valli and C. Lucas, Linguistics of American Sign Language, 5th ed. Washington, DC: Gallaudet University Press, 2011.
3. U. Meier, "A brief overview of the manual channel in sign language," in Proc. Workshop on Gesture in Language and Speech, 1996.
4. Registry of Interpreters for the Deaf, "NAD-RID National Council on Interpreting," RID, Alexandria, VA, 2021.
5. A. Steinberg et al., "Health care system accessibility—experiences and perceptions of deaf people," J. General Internal Medicine, vol. 21, no. 3, pp. 260–266, 2006.
6. O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems," Computer Vision and Image Understanding, vol. 141, pp. 108–125, 2015.
7. D. J. Sturman and D. Zeltzer, "A survey of glove-based input," IEEE Computer Graphics and Applications, vol. 14, no. 1, pp. 30–39, 1994.
8. J. Shotton et al., "Real-time human pose recognition in parts from single depth images," in Proc. IEEE CVPR, 2011, pp. 1297–1304.
9. L. Pigou et al., "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition," Int. J. Computer Vision, vol. 126, no. 2, pp. 430–439, 2018.
10. J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3D convolutional neural networks," in Proc. IEEE ICME, 2015, pp. 1–6.
11. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
12. A. Vaswani et al., "Attention is all you need," in Advances in NeurIPS, vol. 30, 2017, pp. 5998–6008.
13. F. Zhang et al., "MediaPipe Hands: On-device real-time hand tracking," in Proc. Workshop on CV for

AR/VR at ECCV, 2020.
14. S. S. Fels and G. E. Hinton, "Glove-talk: A neural network interface between a data-glove and a speech synthesizer," IEEE Trans. Neural Networks, vol. 4, no. 1, pp. 2–8, 1993.
15. L. Pigou, S. Dieleman, P. Kindermans, and B. Schrauwen, "Sign language recognition using CNNs," in Proc. Workshop at ECCV, 2014.
16. P. Molchanov et al., "Online detection and classification of dynamic hand gestures with recurrent 3D CNNs," in Proc. IEEE CVPR, 2016, pp. 4207–4215.
17. N. C. Camgoz et al., "Neural sign language translation," in Proc. IEEE CVPR, 2018, pp. 7784–7793.
18. N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in Proc. IEEE CVPR, 2020, pp. 10023–10033.
19. W. Hu et al., "Transformer-based isolated sign recognition with landmark features," in Proc. IEEE ICASSP, 2020, pp. 2438–2442.
20. S. Taskiran, N. Kahraman, and C. E. Erdem, "Face recognition: Past, present and future," Digital Signal Processing, vol. 106, p. 102809, 2020.
21. V. Bheda and N. D. Radpour, "Using deep convolutional networks for gesture recognition in ASL," arXiv:1710.04977, 2017.
22. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR, 2015.
23. TensorFlow Team, "TensorFlow Lite: On-device machine learning," Google, 2023. [Online]. Available: https://www.tensorflow.org/lite
24. G. Hinton et al., "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.
25. V. Bazarevsky et al., "BlazePose: On-device real-time body pose tracking," arXiv:2006.10204, 2020.