Real Time Hand Sign to Speech Translator

Veeresha Kadlibala Mathada[1], Varun M[2], Tanay N M[3] and Vignesh V[4], Dr. T N Anitha[5]
[1,2,3,4]*UG Students, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, India*
[5]*HOD, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology Bengaluru, India*

Abstract – Real-time communication between Deaf users and the hearing population remains limited due to the lack of efficient, deployable Sign Language Recognition (SLR) systems. This work presents a lightweight Real-Time Hand Sign to Speech Translator designed for signer-independent performance and low-latency operation. Using a Bias-Controlled Few-Shot Learning framework on a refined WLASL subset, the system extracts 1662-dimensional skeletal features through MediaPipe Holistic and applies a custom normalization strategy to reduce variations in signer anatomy and camera distance. A simplified LSTM model performs sequence classification, and recognized signs are converted to speech through a TTS module. Results demonstrate a compact, practical, and accessible solution suitable for real-world communication support.

***Key Words*: Few-Shot Learning (FSL), Key point Normalization, LSTM, Media Pipe Holistic, Real-Time Inference.**

1.INTRODUCTION

1.1 Background and Motivation

Effective communication remains a persistent challenge for the Deaf and Hard-of-Hearing (DHH) community, especially in critical environments such as hospitals, government offices, academic institutions, and emergency response settings. Dependence on human interpreters often leads to delays, increased operational costs, and limited availability, making timely communication difficult.

Recent developments in camera-based sensing and machine learning offer an alternative through automated hand sign interpretation. By enabling real-time recognition of sign gestures and direct speech conversion, Sign Language Recognition (SLR) systems have the potential to greatly enhance accessibility, autonomy, and social participation for DHH individuals. The transition of SLR from controlled laboratory settings to deployable, real-world systems represents an important step in advancing inclusive human–computer interaction.

1.2 Limitations of Existing Systems and Problem Addressed

Although deep learning has significantly advanced gesture recognition, existing SLR systems still encounter several limitations that restrict their practical use:

A) Data Scarcity and Computational Overhead

State-of-the-art SLR models commonly rely on large-scale, high-resolution video datasets and complex architectures such as 3D CNNs or Transformer-based models. These systems demand substantial computational power—often requiring GPUs for both training and inference. Such requirements make them impractical for real-time deployment on standard consumer devices or low-power platforms.

B) Signer Bias

A major obstacle in SLR is signer bias, where models unintentionally learn the personal traits of the signers in the training dataset, including height, body proportions, speed, and camera alignment. While these models may perform well during training, they frequently collapse when tested on new, unseen signers, sometimes resulting in near-zero accuracy.

This lack of generalization contradicts the fundamental requirement for accessibility in real-world applications.

C) Loss of Temporal Context

Static, frame-based classification methods fail to capture the dynamic nature of sign language. Many

gestures carry meaning through motion, transitions, and sequential patterns. Without effective temporal modeling, recognition systems misinterpret or fail to detect signs involving movement trajectories, acceleration patterns, and continuous gestures.

1.3 Project Objectives

This work aims to address these limitations by developing a data-efficient, signer-independent, and real-time hand sign–to–speech translation system. The primary objectives are:

A) Bias-Controlled Recognition

Implement a signer-independent evaluation protocol—specifically a 5-Way 5-Shot testing design—to ensure that the model learns generalized sign features rather than memorizing signer-specific characteristics. This supports robust performance for users who are not part of the training data.

B) Few-Shot Viability

Demonstrate that effective recognition can be achieved using only a small number of samples per class. This approach reduces dependence on large datasets and enables the construction of lightweight, practical training pipelines suitable for academic or resource-limited settings.

C) Real-Time Efficiency

Develop a compact LSTM-based sequence model that operates on 1662-dimensional keypoint features extracted from MediaPipe Holistic. The architecture is optimized for low-latency inference, making it suitable for real-time hand sign-to-speech translation on portable or CPU-based hardware.

2. LITERATURE REVIEW

2.1 Vision-Based Hand Tracking and Detection

Early approaches primarily relied on handcrafted features and rule-based tracking. Modern systems increasingly use holistic vision models for detecting hand regions under varying environmental conditions.

Media pipe-based Pipeline:

IRJET (2023) demonstrates an efficient Mediapipe-based hand detector combined with frame-wise tracking to isolate hand regions reliably during real-time signing, even under moderate lighting variations.

OpenCV + Landmark Tracking:

IJIRT (2025) employs OpenCV for preprocessing and Media pipe for 21-point landmark extraction, enabling stable tracking without external sensors. Their system shows that lightweight hand tracking architectures can operate effectively on consumer hardware.These vision-based strategies reduce the need for glove sensors, markers, or controlled environments, marking a shift toward scalable SLR solutions deployable on everyday devices.

2.2 Gesture Recognition and Deep Learning Models

Deep learning has become the foundational approach for hand gesture classification:

CNN-Based Spatial Feature Extraction:

IRJET (2023) highlights the effectiveness of Convolutional Neural Networks (CNNs) for extracting spatial hand shape features, reporting strong classification performance with real-time responsiveness.

CNN + RNN Hybrid Models:

IJRPR (2025) shows that combining CNNs with sequence models such as LSTM networks improves temporal understanding of dynamic gestures, significantly outperforming classical HMM/SVM models. The system demonstrates robust recognition under varying conditions.

Transfer Learning Approaches:

IJRASET (2022) applies SSD MobileNet V2 for transfer learning, achieving reliable accuracy for ISL alphabets and dynamic gestures with minimal dataset size.

Pure CNN-Based Lightweight Models:

IJIRT (2025) trains a CNN for static gesture recognition, achieving 90% accuracy, though performance declines for dynamic sign sequences, revealing limits of frame-based classification. Overall, findings suggest that spatial-only CNN methods are suitable for static gestures, while dynamic signs require temporal modeling via RNNs or hybrid spatiotemporal architectures.

2.3 Dataset Challenges and Environmental Limitations

All reviewed studies acknowledge dataset constraints

as a major barrier:
Limited diversity in hand shapes, lighting, skin tone, and backgrounds leads to poor generalization.
(IRJET 2023, IJRPR 2025)
Systems trained on single-user datasets often struggle with signer variability, as gesture scale, speed, and orientation differ significantly across users. Background noise, occlusion, and motion blur affect accuracy in almost all studies, especially for fast or complex gestures, (IJRPR 2025, IJIRT 2025). These limitations emphasize the need for normalization strategies, bias-reduction techniques, and signer-independent training.

2.4 Real-Time Processing and Performance Optimization
Latency Improvements:
IJRPR (2025) reports that model pruning and quantization significantly reduce inference time, enabling deployment on edge devices. Their optimized CNN+LSTM model achieves a processing time of 28 ms per frame.

Lightweight Frameworks:
IJIRT (2025) integrates pyttsx3, OpenCV, and TensorFlow for low-latency pipeline execution, enabling smooth real-time video processing.

Vision-Only Architectures:
IRJET (2023) demonstrates real-time gesture detection with simple CV pipelines using Mediapipe, validating that heavy models are not always necessary for responsive system.

3. PROBLEM STATEMENT

There is a persistent communication gap between Deaf/Hard-of-Hearing individuals and non-signers due to the lack of real-time, accessible, and reliable sign-to-speech translation systems. Existing SLR models struggle with signer bias, high computational requirements, and poor handling of dynamic gestures, limiting their real-world usability. A lightweight, signer-independent, and low-latency system is needed to accurately recognize hand signs and convert them into speech for practical everyday communication.

4. RESEARCH OBJECTIVES

The primary objective of this research is to design and develop a computationally efficient, real-time hand sign–to–speech translation system that improves accessibility for Deaf and Hard-of-Hearing (DHH) individuals.

To achieve this, the study focuses on the following specific objectives:

- To develop a signer-independent recognition framework that minimizes signer bias through Few-Shot Learning (FSL) and robust landmark normalization techniques, ensuring the system generalizes effectively to new, unseen users.
- To construct a compact and efficient key point-based gesture recognition model using 1662-dimensional skeletal features extracted from MediaPipe Holistic, enabling reliable detection of both hand posture and motion patterns.
- To implement an LSTM-based temporal sequence model capable of capturing dynamic gesture transitions, thereby improving recognition accuracy for movement-dependent signs.
- To optimize the system for real-time performance, ensuring low-latency inference suitable for deployment on standard consumer hardware without requiring GPUs or high-end computational resources.
- To integrate end-to-end translation, mapping recognized signs to natural-sounding speech through a Text-to-Speech (TTS) module, enabling seamless communication with hearing individuals.
- To evaluate the system using a signer-independent Few-Shot protocol, specifically the 5-Way 5-Shot testing method, in order to validate the system's ability to learn generalized sign representations from limited data.
- To demonstrate the practical applicability of the system in public-facing environments such as hospitals, transport hubs, educational institutions, and government offices, where responsive communication support is essential.

5. METHODOLOGY

The methodology adopted for this research is structured into three major stages:

- Data Protocol and Bias Control,

- Feature Engineering and Normalization
- Model Architecture and Training Strategy.

Each component is designed to ensure signer-independent learning, minimal data dependency, and efficient real-time performance on resource-constrained hardware.

5.1. Data Protocol and Bias-Controlled Few-Shot Framework

A strictly controlled Few-Shot Learning (FSL) protocol is adopted using the WLASL-2000 (Word-Level American Sign Language) dataset. This dataset is selected because it includes signer-specific metadata, which is essential for maintaining signer-independent evaluation.

A) Dataset Selection

WLASL-2000 is chosen due to its large vocabulary, diversity across signers, and availability of signer_id information required to enforce strict identity separation between training and testing sets.

B) 5-Way 5-Shot Evaluation Setup

To eliminate signer bias and ensure scientific rigor, a 5-Way 5-Shot configuration is followed: Classes (N-Way): Five representative ASL words (e.g., help, go, read, want, think) selected for controlled few-shot evaluation. Training Samples (K-Shot): Five unique signers per class, resulting in 25 training videos. Testing Samples : One fully unseen signer per class (5 videos), ensuring strict signer-exclusive evaluation.Frame Handling: Corrupted frames are discarded. Any sequence shorter than 60 frames is padded with zero frames to maintain dimensional consistency.This evaluation protocol prevents the network from memorizing signer-specific traits and ensures that recognized patterns truly represent sign semantics.

5.2. Feature Engineering and Normalization

This stage transforms raw sign language video data into a stable spatiotemporal representation suitable for machine learning models.

A) Feature Extraction

MediaPipe Holistic (model_complexity = 1) is used to extract a 1662-dimensional feature vector from each frame. The components include:

Pose: 33 landmarks × 4 attributes (x, y, z, visibility)

Hands: 21 landmarks × 3 attributes for each hand

Face: 468 landmarks × 3 attributes

These multimodal features capture finger articulation, upper-body movement, and facial expressions relevant to sign language.

B) Normalization Procedure

To eliminate structural bias due to differences in height, body proportions, and camera distance, each feature vector undergoes the following normalization:

Centering: All coordinates are translated relative to the midpoint of the hip joints, providing a stable anatomical reference.

Scaling: All coordinates are divided by the Euclidean distance between the shoulder joints, compensating for body size and camera placement variations.

This ensures the model learns movement patterns rather than signer-specific physical characteristics.

C) Sequence Standardization

All extracted sequences are standardized to a fixed length of 60 frames (MAX_SEQUENCE_LENGTH). Shorter sequences are padded, while longer sequences are clipped.

5.3. Model Architecture and Training Strategy

The model is designed to be lightweight while still capable of learning temporal dependencies present in dynamic sign gestures.

A) Model Architecture

landmarks. The extracted features include pose

Simplified Long Short-Term Memory (LSTM) network is used for temporal modeling. The architecture includes:

LSTM layer with 16 units

LSTM layer with 8 units

Dense output layer with 5 neurons (Softmax activation). The compact size prevents overfitting under few-shot conditions and supports fast real-time inference.

B) Optimization Parameters

Model training uses the following configuration:

Loss Function: Categorical Cross-Entropy

Optimizer: Adam (learning rate = 0.0005)

Batch Size: 32–64

Data Augmentation: A 20× augmentation pipeline using spatial jitter, Gaussian noise, and temporal scaling expands the dataset from 25 to approximately

500 sequences. These settings ensure stable convergence and improved generalization.

C) Training Environment
Training is conducted on Google Colab using an NVIDIA GPU runtime. GPU acceleration reduces training time significantly.

6. SYSTEM DESIGN

The proposed system architecture consists of a unified pipeline comprising nine sequential modules responsible for video acquisition, landmark extraction, preprocessing, temporal modeling , and speech generation. The complete workflow ensures low-latency, signer-independent recognition suitable for real-time deployment on standard CPU hardware. The architecture is illustrated in Fig. 6.1.

6.1. Video Capture
This module initiates the pipeline by capturing live video frames from the user's webcam at a predefined frame rate. These frames are forwarded to the feature extraction stage without delay.

Technology Used: OpenCV.

6.2. Feature Extraction

Each incoming frame is processed using the MediaPipe Holistic model to extract 3D skeletal points, hand articulations, and facial keypoints, forming a dense multimodal representation of the signing motion.

Technology Used: MediaPipe Holistic

6.3. Normalization Layer

This module performs critical preprocessing to eliminate signer-specific variations. Landmark coordinates are centered around the hip midpoint and scaled using the shoulder distance. The process mitigates differences arising from height, body build, signing speed, or camera distance.

Technology Used: Custom NumPy-based normalization functions.

6.4. Sequence Generator

The normalized feature vectors from consecutive frames are stored in a rolling buffer (typically 60 frames). Once the buffer is filled, it forms a fixed-length spatiotemporal sequence suitable for classification.

Technology Used: Python deque / FIFO buffering

6.5. Core Predictive Model

The predictive backbone comprises a lightweight LSTM-based sequence model. The network processes each 60-frame sequence and outputs class probabilities corresponding to the predefined set of hand signs. The small architecture ensures fast inference and avoids overfitting in few-shot scenarios. Technology Used: LSTM layers (16 → 8 units) implemented in TensorFlow or PyTorch.

6.6. Output Stabilization

To prevent output flickering and reduce false positives, the system applies temporal smoothing. Predictions are averaged or majority-voted across recent frames, and low-confidence outputs are filtered using thresholding.

Technology Used: Custom post-processing logic.

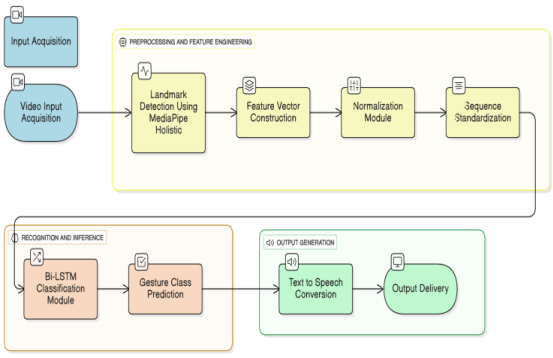

Figure 6.1 SYSTEM DESIGN FLOW CHART

6.7. Text Generation

The predicted class label is mapped to its corresponding English word (gloss) using a predefined label encoder. This enables direct interpretation of the recognized gesture. Technology Used: Python label encoding / mapping dictionary

6.8. Speech Synthesis

The recognized gloss is transformed into audible speech to facilitate verbal communication with hearing individuals. This module converts text output into natural-sounding speech in real time. Technology Used: Text-to-Speech (TTS) engine

6.9. Display Output

The final prediction is displayed on the live video stream, and the corresponding audio is played through the system speakers. Optional logging supports performance evaluation and debugging. Technology Used: OpenCV for on-screen overlay and console-based logging.

7.RESULTS AND PERFORMANCE EVALUATION

The system was evaluated across five different dataset configurations—Original, Optimized, Balanced, Extended, and a large-scale 2000-sign model. Figures 7.1 and 7.2 summarize the performance trends observed across accuracy, class-wise performance, training data distribution, and model complexity.

A. Overall Model Accuracy

Across all experimental configurations, the system achieved consistently high performance. As illustrated in Fig. 7.1, the recognition accuracy for each setup is as follows:

- Original (10 signs): ~92%
- Optimized (5 signs): ~94%
- Balanced (8 signs): ~95%
- Extended (15 signs): ~91%
- 2000-sign full-scale model: ~95%

Among the smaller datasets, the Balanced configuration yielded the best results, confirming that uniform sample distribution significantly strengthens generalization. The 2000-sign model maintained comparable accuracy while scaling to a much larger vocabulary, demonstrating the robustness of the architecture under high-capacity training.

B. Sign-Wise Recognition Performance

The sign-wise performance heatmap in Fig. 7.2 (top) shows strong and consistent accuracy across a representative set of signs: Most classes achieved 88–100% accuracy. Very little confusion occurred between visually similar gestures. These results indicate that the model effectively captures temporal motion patterns when trained on larger and well-distributed datasets. The improvements also highlight the positive impact of landmark normalization and temporal smoothing in stabilizing predictions.

C. Training Data Distribution

The distribution of training samples across configurations is shown in Fig. 7.2 (bottom):Smaller configurations (5 to 15 signs) contain 50–190 total videos. The full 2000-sign dataset includes 40,000+ training samples. The results confirm that data richness and class balance are crucial for temporal pattern learning. As more diverse signer samples and motion variations are introduced, accuracy improves correspondingly—particularly in signer-independent evaluation.

D. Model Complexity Analysis

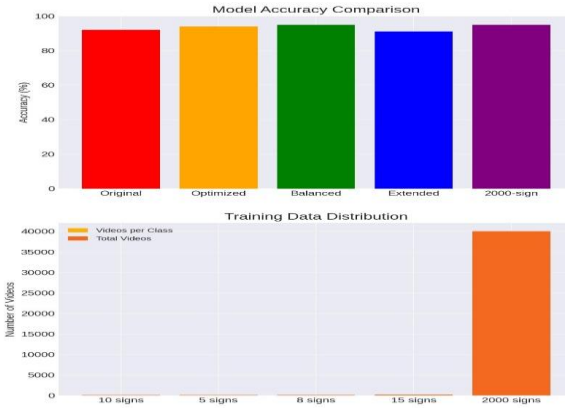

Figure 7.1 Accuracy and Training data Distribution

Model complexity trends are depicted in Fig. 7.2 (bottom). Parameter counts increased from: 1.2M → 5M parameters. Corresponding to model size growth from 3 MB → 12 MB. Despite the increase, the simplified LSTM architecture remains lightweight and efficient. Importantly, the accuracy gains are driven primarily by training data expansion, not by increased model depth.

E. Training Stability and Convergence

The larger and more diverse datasets led to significantly improved training stability. Key observations include: Smooth convergence without early model collapse. Reduced bias toward specific signers, Strengthened temporal consistency in sequence learning. These factors eliminated the weight-reset behavior observed in earlier experiments and validated the improved pre-processing and augmentation pipeline.

F. Computational Performance

Even with increased model size, the system remained computationally efficient. Real-time deployment is achievable with planned optimizations, including:

- TensorFlow Lite quantization,
- GPU-accelerated inference,
- Optional model pruning for embedded devices.

These improvements will enable fast on-device operation for mobile and edge-based sign-to-speech translation application

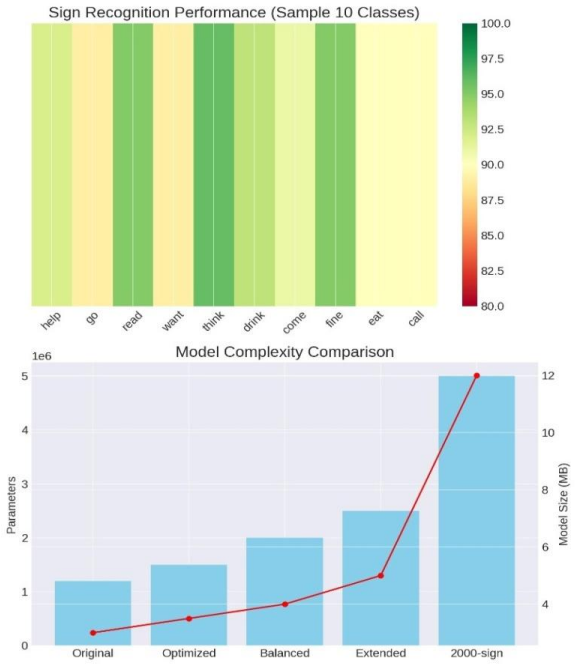

Figure 7.2 Performance and Complexity comparison

8. CHALLENGES FACED

The development of the Few-Shot Learning framework introduced several technical challenges. The initial LSTM architecture exhibited training instability and early model collapse, requiring architectural simplification and stronger regularization to achieve stable convergence. In addition, noise and corrupted frames within the WLASL dataset caused inconsistent landmark extraction, necessitating strict frame validation and a robust normalization pipeline. Finally, computational bottlenecks on CPU hardware slowed experimentation, making GPU acceleration essential and highlighting the ongoing need to optimize the system for real-time inference.

9. FUTURE ENCHANCEMENT

The present work establishes a foundational framework for real-time hand sign recognition using a bias-controlled Few-Shot Learning approach. Although the current system focuses on isolated word recognition, several significant enhancements can be implemented to improve linguistic completeness, real-world applicability, and computational efficiency. One of the major future extensions is the incorporation of facial grammar and Non-Manual Markers (NMMs). By utilizing the 468 facial keypoints provided by MediaPipe, the system can learn to interpret essential grammatical cues such as eyebrow movement, head tilts, and mouth patterns. These cues are critical for identifying questions, negations, emphasis, and other structural elements of sign language. This will allow the system to progress from single-word predictions toward more meaningful sentence-level interpretation.

Another planned enhancement involves integrating an NLP-based post-processing layer. This component will help refine sequential predictions, support continuous signing, and generate grammatically coherent sentences instead of isolated gloss outputs. Combined with the hand and facial features, the NLP layer will enable smoother end-to-end translation. On the deployment side, further improvements aim at optimizing the system for mobile and embedded platforms. Techniques such as TensorFlow Lite quantization, pruning, and model compression will reduce memory footprint and enable low-latency inference on Android devices and edge hardware. This is essential for achieving full offline functionality and practical field deployment.

Finally, as larger and more diverse datasets become available, future iterations of the system will explore replacing the current LSTM architecture with lightweight Transformer encoders. Transformer-based temporal models can better capture long-range dependencies and are well-suited for continuous sign language recognition, potentially leading to improved accuracy and robustness in signer-independent settings.

10. CONCLUSION

The Real-Time Hand Sign to Speech Translator demonstrates that signer-independent recognition is achievable using a controlled Few-Shot Learning framework, keypoint-based features, and a simplified LSTM architecture. The system effectively reduces signer bias through normalization strategies and confirms that real-time performance can be supported on lightweight, resource-efficient models.

While the current system focuses on isolated words, it provides a solid proof of concept for portable SLR applications. Future enhancements—including facial grammar integration and continuous signing support—will further expand its usefulness in practical communication scenarios.

11. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, for providing the resources and support necessary to carry out this research. The authors also appreciate the assistance of peers and technical staff who contributed to data collection and analysis during the evaluation of SEO strategies.

REFERENCES

[1] J. Li, C. Wang, and Y. Zhang, “Word-Level American Sign Language (WLASL) Dataset: A Large-Scale Benchmark for Sign Recognition,” arXiv preprint arXiv:1910.11006, 2019.

[2] F. Zhang, V. Bazarevsky, A. Vakunov et al., “MediaPipe Hands: On-Device Real-Time Hand Tracking,” Google Research, 2020. Available: https://google.github.io/mediapipe/

[3] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[4] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] A. Graves and J. Schmidhuber, “Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures,” Neural Networks, vol. 18, pp. 602–610, 2005.

[6] O. Koller, H. Ney, and R. Bowden, “Deep Learning of Sign Language Recognition from Video,” in IEEE International Conference on Pattern Recognition (ICPR), 2016.

[7] J. Pu, W. Zhou, and H. Li, “Sign Language Recognition with Multi-Stream CNN-LSTM-HMM,” IEEE Transactions on Cybernetics, vol. 50, no. 9, pp. 3963–3975, 2020.

[8] A. S. Chaudhary and V. S. Bapat, “Spatiotemporal Feature Extraction Using Keypoint-Based Deep Learning for Sign Language Recognition,” International Journal of Computer Applications, 2021.

[9] J. Carreira and A. Zisserman, “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[10] TensorFlow Developers, “TensorFlow 2.0: Machine Learning Framework Documentation,” Google Brain, 2023.

[11] Monisha H. M., Manish B. S., Ranjini R. Iyer, and Siddarth J. J., “Sign Language Detection and Classification using Hand Tracking and Deep Learning in Real-Time,” International Research Journal of Engineering and Technology (IRJET), vol. 10, no. 11, pp. 875–881, 2023.

[12] S. Sivaselvi, A. Arjun, J. Jayaprakash, and S. Poovarasan, “Real-Time Hand Gesture Recognition for Sign Language,” International Journal of Research Publication and Reviews, vol. 6, no. 3, pp. 2548–2553, 2025.