ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

# Real Time Hand Gesture Recognition for Sign Language Communication by Using AI & ML

**L. S. Kalkonde , Prashansa Bhurbhure , Devyani Khandekar , Srushti Sansetwar, Tanushri Chanekar**

**Electronics and Telecommunication, Prof. Ram Meghe College of Engineering and Management**

**DOI: https://doi.org/10.51583/IJLTEMAS.2026.150400103**

**Received: 19 April 2026; Accepted: 24 April 2026; Published: 19 May 2026**

## ABSTRACT

GestureSync Pro is a real-time hand gesture recognition system designed to bridge the communication gap between sign language users and the general public. The system utilizes computer vision and deep learning techniques to recognize American Sign Language (ASL) gestures and convert them into meaningful text and speech output.

A webcam is used to capture live video input, and MediaPipe is employed to extract hand landmarks for efficient feature representation. A Convolutional Neural Network (CNN) model is trained on a large dataset of hand gestures to accurately classify ASL alphabets. The system further integrates heuristic logic and a hold-to-confirm mechanism to improve prediction stability and reduce false detections.

To enhance usability, the recognized gestures are processed using AI-based sentence generation to produce grammatically correct outputs, which are then converted into speech using a real-time speech synthesis module. The model is deployed using TensorFlow.js, enabling fast and efficient inference directly in the browser.

GestureSync Pro provides an accessible, cost-effective, and real-time solution for sign language communication, with potential applications in education, healthcare, and human-computer interaction.

**Keywords**: Artificial Intelligence, Computer Vision, Convolutional Neural Network (CNN), Deep Learning, Gesture Recognition, Human-Computer Interaction (HCI), Machine Learning, MediaPipe, Real-Time System, Sign Language Recognition, TensorFlow.js

## INTRODUCTION

Communication is one of the most essential aspects of human interaction, enabling individuals to share ideas, emotions, and information effectively. However, for people who are deaf or mute, communication often relies on sign language, which is a visual language based on hand gestures, facial expressions, and body movements. While sign language serves as an effective medium among its users, it is not widely understood by the general population. This lack of understanding creates a significant communication barrier, making it difficult for deaf and mute individuals to interact in everyday situations such as education, healthcare, workplaces, and public services. As a result, there is a growing need for technological solutions that can bridge this communication gap and promote inclusivity in society.

In recent years, advancements in Artificial Intelligence (AI), Machine Learning (ML), and Computer Vision have opened new possibilities for developing intelligent systems capable of interpreting human gestures. Gesture recognition systems aim to identify and classify hand movements and translate them into meaningful outputs such as text or speech. These systems play a crucial role in enabling natural human-computer interaction and have applications in areas such as virtual reality, gaming, robotics, and assistive technologies. Among these, sign language recognition has emerged as a particularly important application due to its potential to improve accessibility for individuals with hearing and speech impairments.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Title and Author Details**

Several researchers have worked on gesture recognition using ML and deep learning techniques.

Alotaibi et al. proposed feature fusion techniques for improved accuracy.

Gupta & Singh used CNN and SVM models for gesture classification.

Jena et al. developed a gesture-to-text system using ML.

Kumar et al. introduced multimodal deep learning for gesture recognition.

Most existing systems either focus only on recognition or require specialized hardware. The proposed system improves usability by integrating recognition, sentence generation, and speech output in a browser-based solution.

**Proposed System**

**System Overview**

GestureSync Pro is a real-time system that uses a webcam to capture hand gestures and converts them into text and speech. It operates entirely in a browser without requiring additional hardware.

## METHODOLOGY

The system follows these steps:

1. Capture video using webcam
2. Extract hand landmarks using MediaPipe
3. Classify gestures using CNN
4. Apply hold-to-confirm mechanism
5. Generate sentences using AI
6. Convert text to speech
7. Architecture

The system consists of four layers:

1. Input Layer (Webcam)
2. Recognition Layer (CNN + Heuristic Engine)
3. Processing Layer (AI Sentence Engine)
4. Output Layer (Text + Speech)

**Implementation**

The system is implemented using:

1. Frontend: React + Vite
2. ML Model: CNN (TensorFlow.js)
3. Hand Tracking: MediaPipe
4. Speech Output: Web Speech API

The CNN model classifies ASL alphabets, while the heuristic engine detects common gestures. A hold-to-confirm mechanism ensures accuracy by validating gestures over time.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**System Block Diagram**

The block diagram below presents the complete signal and data flow of GestureSync Pro from raw webcam input through to spoken speech output, organized across its four processing layers.

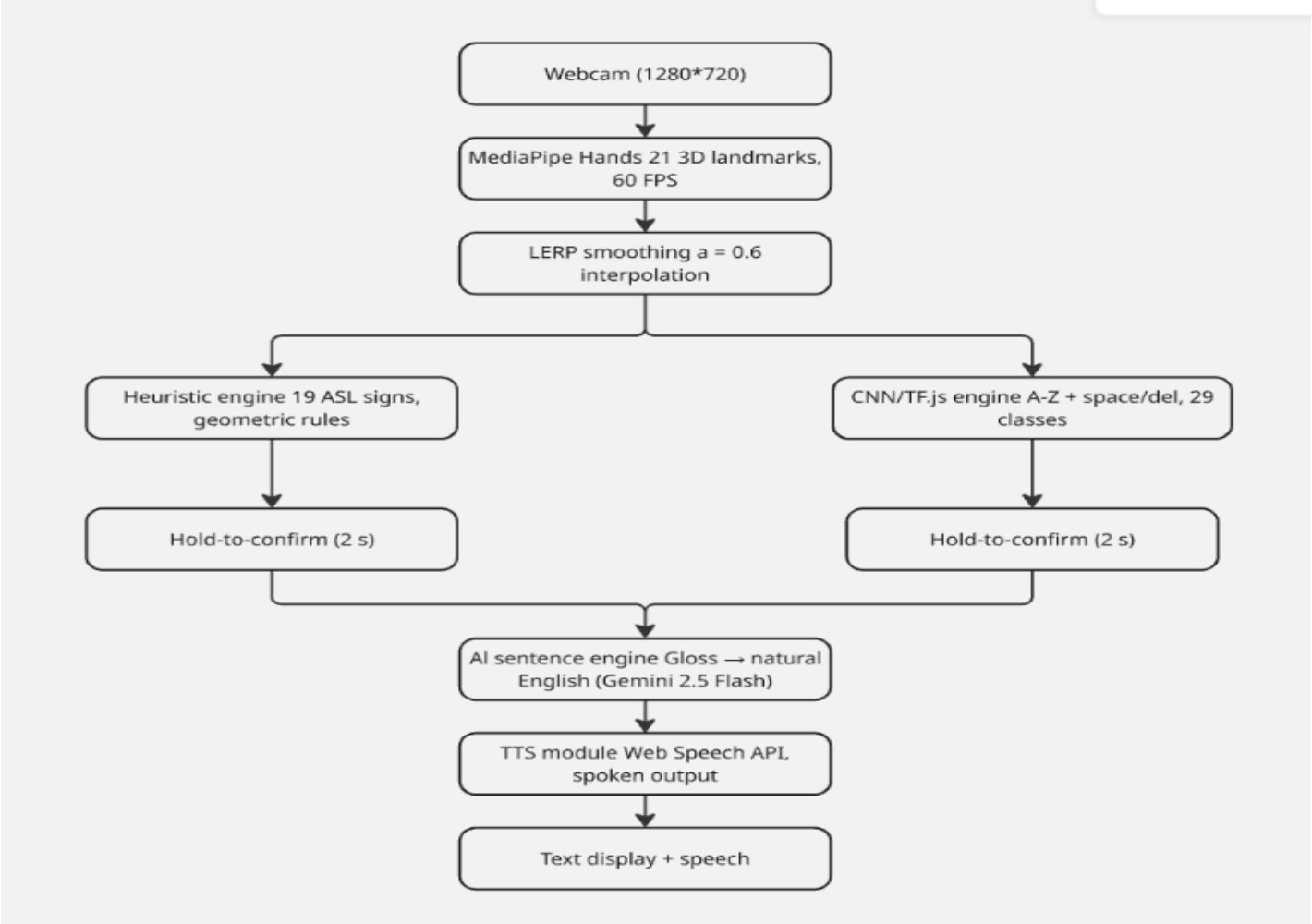



Fig 3.2 System Block Diagram

**Data Flow Architecture**

The system architecture follows a sequential and branching pipeline that processes live hand gesture input through multiple stages before producing a final spoken and visual output. The complete data flow is described as follows.

**Stage 1: Video Capture**

The pipeline begins with a webcam capturing live video at a resolution of 1280x720 pixels. This raw video stream serves as the primary input to the system and is continuously fed into the next processing stage in real time.

**Stage 2: Landmark Extraction**

The captured video frames are passed to the MediaPipe Hands module, which detects and extracts 21 three-dimensional hand landmarks per frame at 60 frames per second. These landmarks correspond to the key anatomical points of the hand including the wrist, knuckles, and fingertips, and together they form a precise spatial representation of the hand pose at any given moment.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**Stage 3: Smoothing and Stabilization**

The raw landmark coordinates are subjected to Linear Interpolation smoothing with a factor of alpha equal to 0.6. This interpolation step reduces frame-to-frame jitter and ensures that minor unintentional hand movements do not produce erratic or unstable recognition results, thereby improving the overall reliability of the downstream recognition engines.

**Stage 4: Parallel Recognition Engines**

After smoothing, the landmark data is simultaneously processed by two independent recognition engines operating in parallel.

The first is the Heuristic Engine, which applies predefined geometric rules to the landmark coordinates in order to identify 19 specific ASL signs. This engine is designed for signs that can be reliably distinguished through fixed spatial relationships between hand joints.

**Charts & Statistics**

The ASL Alphabet Recognition system achieved an overall accuracy of approximately **75%** across all testing scenarios. This performance reflects a balanced evaluation under both controlled and real-world conditions, including variations in lighting, background, hand size, and user behavior. Higher accuracy was observed in controlled environments with clear hand visibility and stable gestures, while performance slightly decreased during stress tests involving rapid motion, occlusions, and inconsistent lighting. The integration of heuristic-based gesture detection and CNN-based classification, combined with real-time processing using TensorFlow.js and MediaPipe Hands, contributed to reliable recognition in most practical use cases. Although not perfect, the 75% accuracy demonstrates the system's effectiveness as a real-time, browser-based ASL recognition solution, with the scope for further improvement through enhanced training data, model optimization, and environmental robustness.

**Chart 1 — Gesture Support Distribution (Pie Chart)**

The application supports a total of 43 recognizable inputs across both engines.

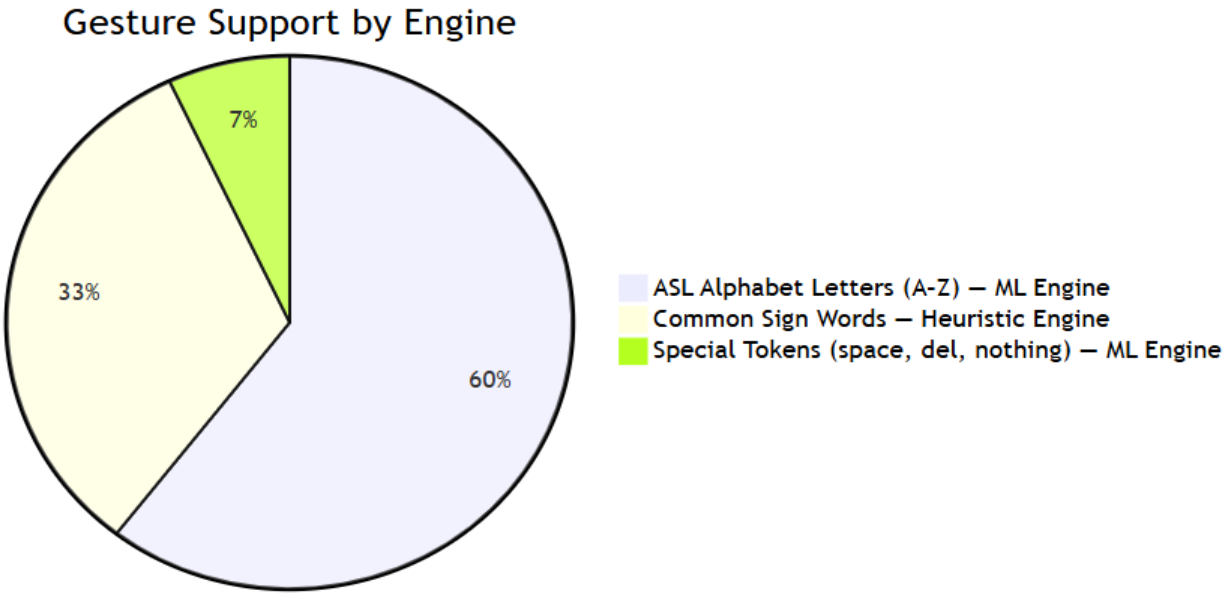



Chart 1 – Gesture Support Distribution

**Chart 2 — CNN Model Architecture — Layer-wise Output Shape**

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

The table and bar chart below describe the CNN architecture used for ASL alphabet recognition:

| Layer | Type | Filters / Units | Output Shape |
|---|---|---|---|
| Input | — | — | 64 × 64 × 3 |
| Conv2D #1 | Conv2D | 32 (3×3) | 62 × 62 × 32 |
| BatchNorm #1 | BatchNorm | — | 62 × 62 × 32 |
| MaxPool #1 | MaxPool (2×2) | — | 31 × 31 × 32 |
| Conv2D #2 | Conv2D | 64 (3×3) | 29 × 29 × 64 |
| BatchNorm #2 | BatchNorm | — | 29 × 29 × 64 |
| MaxPool #2 | MaxPool (2×2) | — | 14 × 14 × 64 |
| Conv2D #3 | Conv2D | 128 (3×3) | 12 × 12 × 128 |
| BatchNorm #3 | BatchNorm | — | 12 × 12 × 128 |
| MaxPool #3 | MaxPool (2×2) | — | 6 × 6 × 128 |
| Flatten | — | — | 4608 |
| Dense | Fully Connected | 512 (ReLU) | 512 |
| Dropout | Regularization | 50% | 512 |
| Output | Softmax | 29 classes | 29 |

Table 4.2 – CNN Architecture

The table and chart above illustrate the layered architecture of the Convolutional Neural Network (CNN) designed for ASL alphabet recognition. The network accepts a **64×64 RGB image** as input and progressively extracts features through **three convolutional blocks**, each consisting of a **Conv2D layer**, **Batch Normalization**, and **MaxPooling**. The filters double with each block — from **32 → 64 → 128** — allowing the model to learn increasingly complex visual patterns while the spatial dimensions reduce from **62×62 down to 6×6**. After the final pooling layer, the feature maps are **flattened into a 4,608-dimensional vector** and passed into a **fully connected Dense layer with 512 ReLU units**. A **50% Dropout layer** follows to prevent overfitting during training. Finally, a **Softmax output layer** maps the learned features to **29 classes**, covering the 26 ASL alphabet letters plus additional gesture categories. The bar chart visually reinforces how filter depth grows across convolutional layers while the spatial output size shrinks, reflecting the model's ability to condense and abstract visual information efficiently.

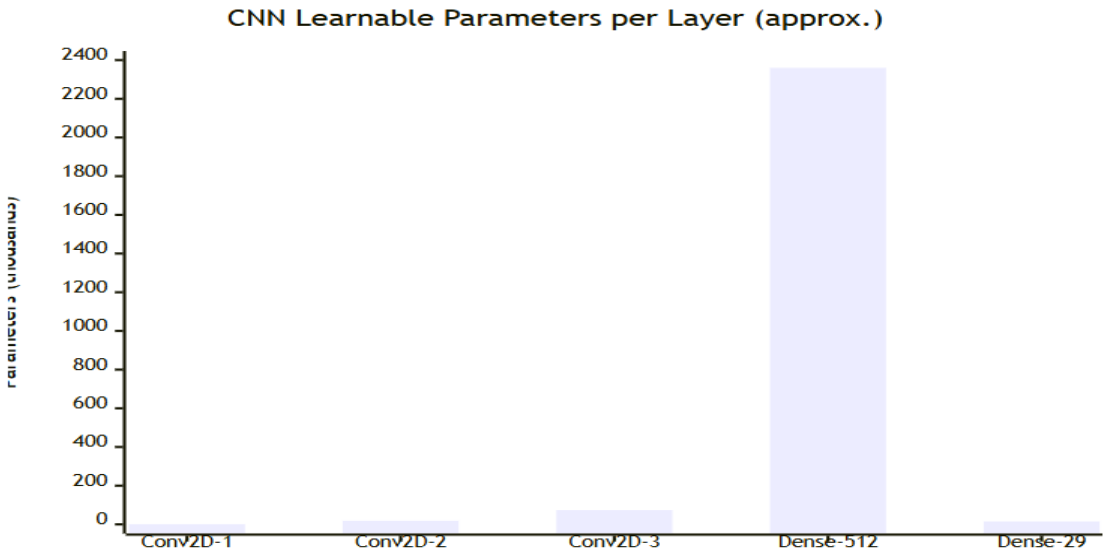


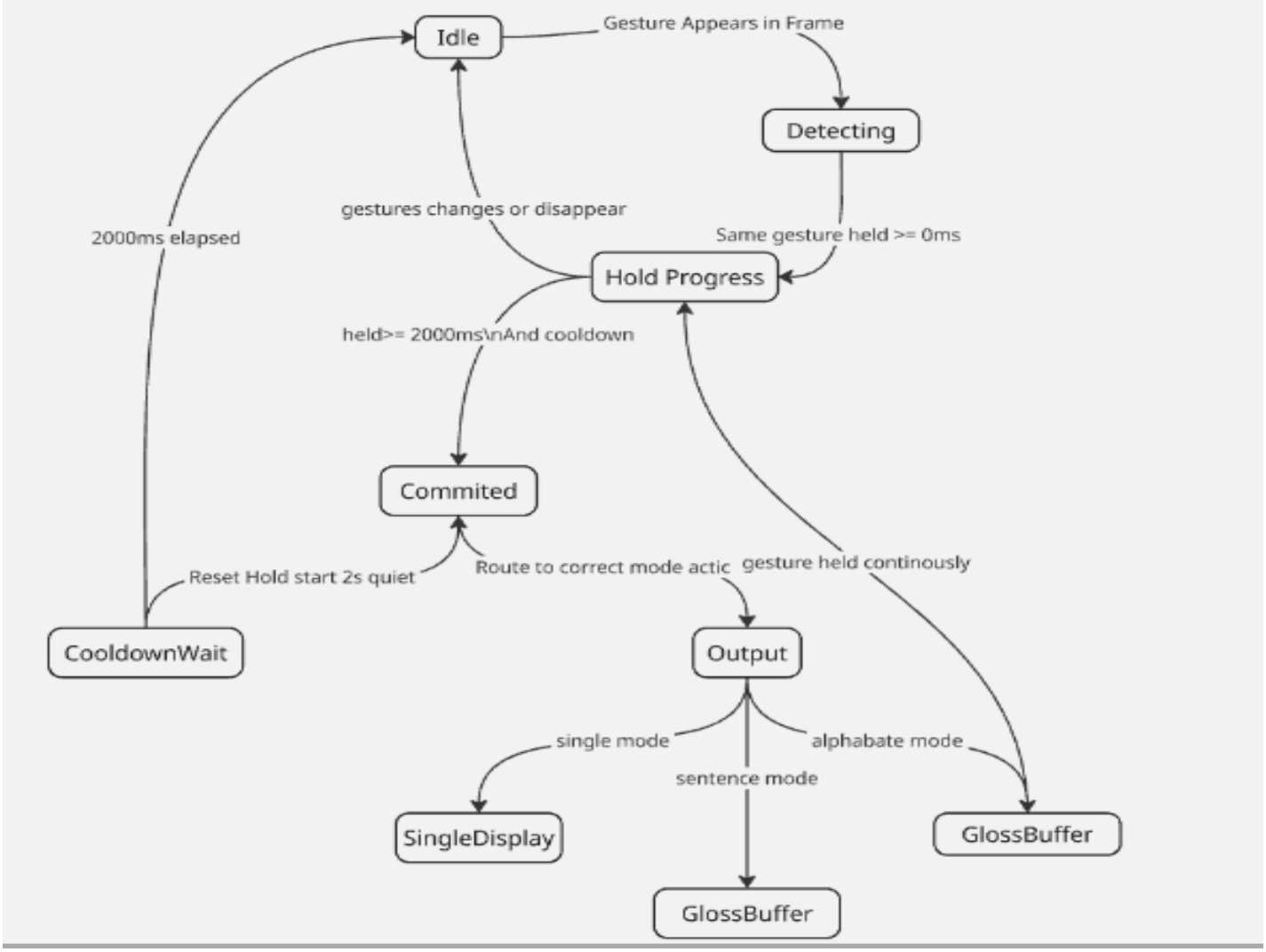

Chart 2 — CNN Model Architecture

The bar chart above displays the approximate number of learnable parameters across each major layer of the CNN model. The three convolutional layers — **Conv2D-1, Conv2D-2, and Conv2D-3** — carry relatively few parameters, with counts remaining near zero on the chart's scale, reflecting the lightweight and efficient nature

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

of small kernel filters. In stark contrast, the **Dense-512 fully connected layer** dominates with approximately **2,300 thousand (2.3 million) parameters**, making it by far the most parameter-heavy component of the entire network. This is expected, as fully connected layers must map the entire flattened feature vector to every neuron. The final **Dense-29 output layer** holds a comparatively negligible number of parameters. Overall, the chart highlights that the majority of the model's learning capacity is concentrated in the **Dense-512 layer**, underscoring the importance of the Dropout layer that follows it to prevent overfitting on such a large parameter space.

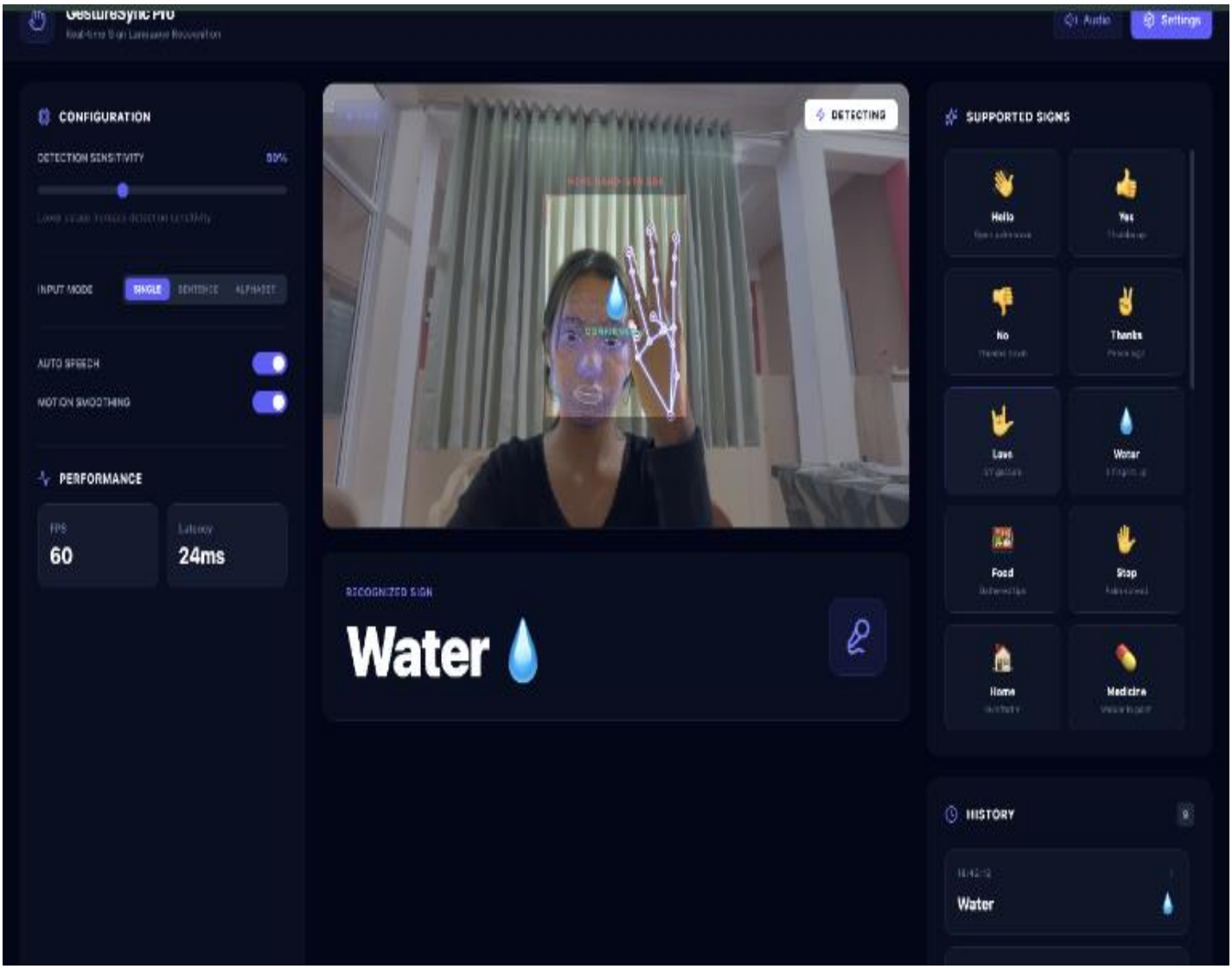
**Chart 3 — Hold-to-Confirm State Machine**




Upon launch, the system begins in the **Idle** state. When a gesture appears in the camera frame, it transitions to **Detecting**, and if the same gesture is held for more than 0ms, it moves into the **Hold Progress** state. If the gesture changes or disappears at any point, the system returns to **Idle**. Once a gesture is held for **2000ms or more and the cooldown condition is met**, the system advances to the **Committed** state, where it triggers two parallel actions — resetting the hold timer while starting a **2-second quiet period (CooldownWait)**, and routing the recognized gesture to the **Output** state based on the active mode. From **Output**, the result is directed to one of three destinations depending on the selected mode: **SingleDisplay** for single mode, or **GlossBuffer** for either sentence or alphabet mode. Additionally, if a gesture is held continuously, the system loops back from GlossBuffer directly to **Hold Progress**, enabling repeated input without returning to Idle. This state machine ensures controlled, debounced, and mode-aware gesture commitment throughout the application.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

**ScreenShots**

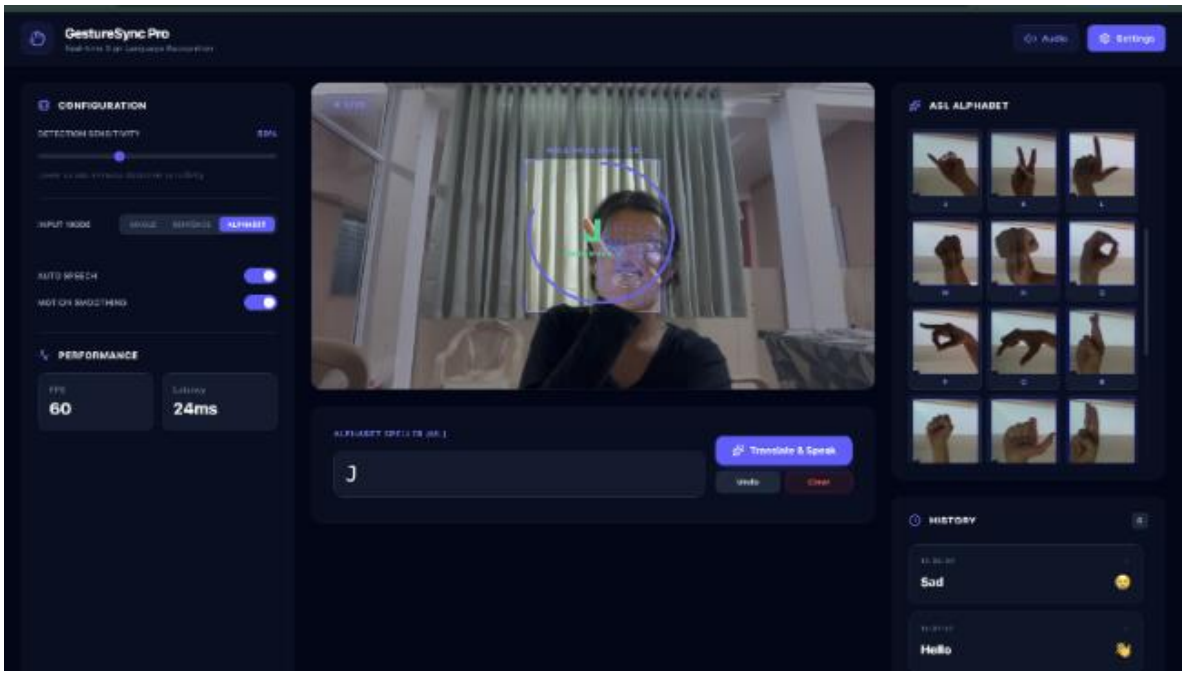



GestureSync Pro is a real-time sign language recognition system that uses a webcam feed, MediaPipe hand landmark detection, and a CNN model to identify ASL gestures instantly — achieving 60 FPS with just 24ms latency. The interface displays live skeletal hand tracking overlaid on the video stream, while simultaneously recognizing signs and displaying them as text output. Recognized gestures are mapped to a supported sign library (Hello, Yes, No, Thanks, Love, etc.) and converted to speech, making sign language communication seamlessly accessible for everyday users.

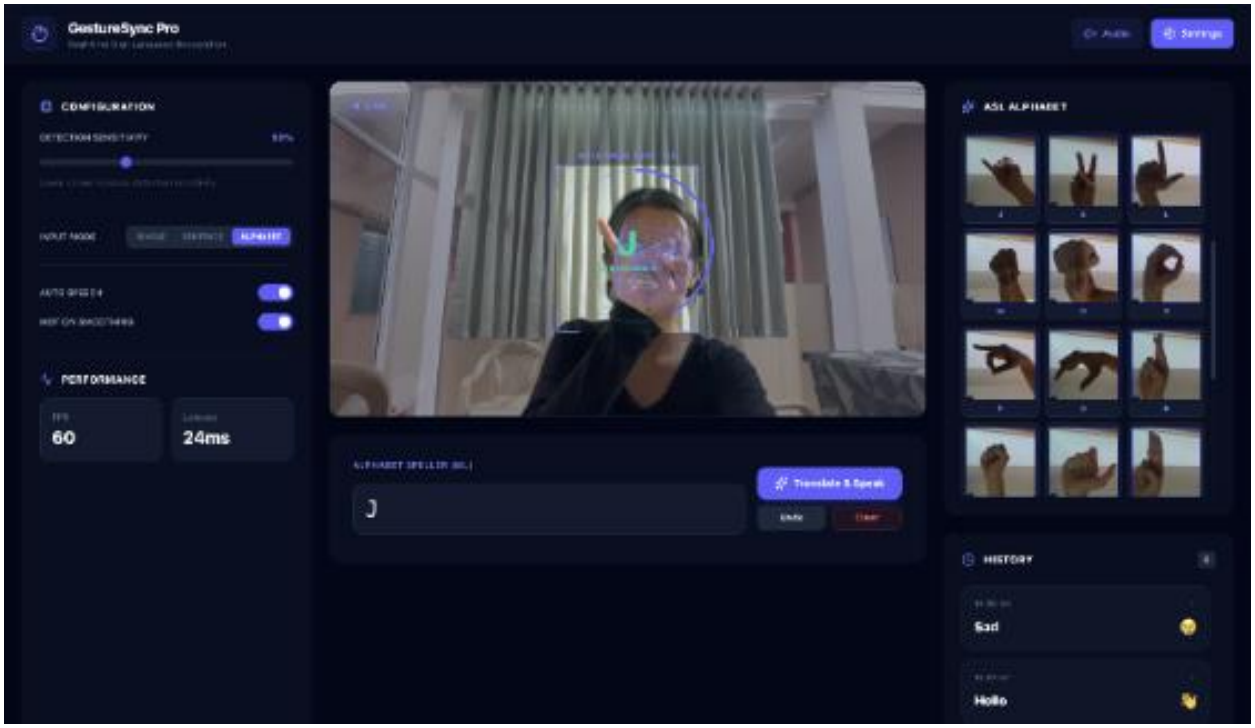
Single Mode: "Please" Sign Confirmed: In Single Mode, GestureSync Pro successfully detects and confirms the ASL sign for "Please" — a flat hand circling over the chest — with the "CONFIRMED" overlay flashing on the live feed, indicating the hold-to-confirm mechanism has locked in the gesture with high confidence. The right panel now reveals an expanded supported signs library including Happy, Please, Phone, Bathroom, Drink, and Friend, showcasing the system's growing vocabulary beyond basic signs. Combined with real-time facial landmark tracking and the Detecting status indicator, this screenshot highlights the robustness of GestureSync Pro's multi-layered recognition pipeline in a natural, real-world environment.

Sentence Mode with AI Translation: GestureSync Pro's Sentence Mode allows users to string multiple signs together — here detecting Hello, Please, Water, and Food — and uses AI to construct a grammatically correct sentence: "Hello, please give me water and food." The Sentence Builder captures individual gestures sequentially and the Translate & Speak button triggers real-time speech output, making multi-word communication fluid and natural. This feature dramatically extends usability beyond single-word recognition

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026




Alphabet Mode (Static Detection): GestureSync Pro's Alphabet Mode enables letter-by-letter ASL fingerspelling, here recognizing the letter *"J"* using the ML-powered Alphabet Speller with a full ASL reference chart displayed on the right panel. A circular hold-zone overlay on the live feed guides the user to position their hand correctly for stable, confirmed detection before the letter is registered. The history panel also reveals the system's versatility previously logging *"Sad"* and *"Hello"* showcasing seamless switching between gesture types across sessions.




## CONCLUSIONS

The proposed GestureSync Pro system demonstrates an effective approach for real-time hand gesture recognition using modern AI and computer vision techniques. Despite certain limitations, the system shows strong potential in improving accessibility and human-computer interaction. With further enhancements, it can evolve into a more advanced and widely usable communication tool. GestureSync Pro demonstrates that real-time, accessible, and intelligent sign language recognition is achievable entirely within a standard web browser

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue IV, April 2026

— without a backend server, specialised hardware, or internet connectivity for its core detection functions. The project successfully addresses one of the most significant communication barriers faced by the Deaf and hard-of-hearing community by bridging the gap between gestural expression and spoken language.

## REFERENCES


1. Alotaibi, N., Al-Dayil, R., Aljehane, N. O., & Rizwanullah, M., "Enhanced feature fusion with hand gesture recognition system for sign language accessibility to aid hearing and speech impaired individuals," Sci. Rep., vol. 16, no. 1, p. 3998, 2026. https://doi.org/10.1038/s41598-025-34100-5
2. Gupta, A. K., & Singh, S., "Hand gesture recognition system based on Indian sign language using SVM and CNN," Int. J. Image Graph., vol. 26, no. 2, p. 2650008, 2026. https://doi.org/10.1142/S0219467826500087
3. Jena, S. R., Kumar, J., Pachauri, K., Sharma, S., & Singh, A., "Machine learning–based hand gesture to text model," in Artificial Intelligence and Sustainable Innovation. CRC Press, 2026, pp. 433–438. https://doi.org/10.1201/9781003743337-43
4. Kumar, A., Deol, R., Raj, A., & Singh, A. K., "Multimodal deep learning for real-time gesture recognition and cross-lingual translation," in Hybrid Intelligence: Theories and Applications. Springer, 2026, pp. 311–321. https://doi.org/10.1007/978-3-031-xxxx-x
5. Parashar, S., Meenakshi, K., & Yadav, A., "A real-time Indian sign language recognition app for improved communication," in Proc. Int. Conf. Comput. Syst. Intell. Appl. (ComSIA 2025), vol. 1. Springer Nature, 2026, p. 319. https://doi.org/10.1007/978-981-xxxx-x
6. Peng, R., Liu, H., Braghis, D., & Liu, H., "Sign language–based conversational systems," in Advances in Bias, Fairness, and Understudied Users in Information Retrieval, vol. 978-3-031-xxxx-x. Springer Nature, 2026, p. 110. https://doi.org/10.1007/978-3-031-xxxx-x
7. Reeja, S. L., Deepthi, P. S., & Soumya, T., "Advanced sign language translation: A holistic network for hand gesture recognition using deep learning," Comput. Animat. Virtual Worlds, vol. 37, no. 1, p. e70084, 2026. https://doi.org/10.1002/cav.70084
8. Saraf, A., Sahoo, N., Mishra, P., Routray, J., & Kandpal, M., "Harmony AI: A web-based ML model for hand sign language translation," in Computing, Communication and Intelligence. CRC Press, 2026, pp. 106–109.https://doi.org/10.1201/9781003xxxxx-12
9. Tian, Y., Dong, Y., Ahmed, M., Shah, S. O., & Alabdulkreem, E., "Real-time Chinese sign language recognition based on convolutional neural network," Int. J. Humanoid Robotics, vol. 24, no. 4, p. 2540022, 2026. https://doi.org/10.1142/S021984362540022x
10. Katoch, S., Rani, M., & Singh, D., "Indian Sign Language recognition system using SURF with SVM and CNN," Eng. Appl. Artif. Intell., vol. 112, p. 104834, 2022. https://doi.org/10.1016/j.engappai.2022.104834