**Engineering and Technology Journal e-ISSN: 2456-3358**

**Volume 11 Issue 03 March-2026, Page No.-9295-9304**

**DOI: 10.47191/etj/v11i03.24, I.F. – 8.595**



# Sign Language Detection and Translator with Learning Platform Usability

**Asmi Mishra[1], Vikash Chaudhary[2], Udit Bhardwaj[3]**

[1,2,3] Department of Computer Science & Engineering

Poornima Institute of Engineering & Technology Jaipur, India

**ABSTRACT:** Even though the area of assistive technologies has just improved significantly, and there have been efforts to achieve the digital inclusiveness of the deaf and the hard-of-hearing community. The deaf and hard-of-hearing population in India have met enormous communication barriers due to lack of awareness and resources. Sign language is one of the necessary tools of communication, that are vital to deaf and hearing-impaired people, however, communication between sign language speakers and none-signers is one of the greatest problems. In this paper, a real-time and multilingual translator and sign language recognition system will be proposed to overcome the communication barrier between deaf and hearing communities using a browser-based system. The suggested system makes use of MediaPipe to extract landmarks of the hand efficiently and a Convolutional Neural Network (CNN) model to recognize gestures. The identified signs have been converted to various languages, with the help of external translation APIs, which allows them to be accessible to all of the language territories like Indian regional languages i.e. Hindi, Marathi, Bengali, Tamil and Punjabi, etc. The first priority has been given to DeepL. The architecture is also designed to use commodity hardware and client-side processing reducing computational overhead and making it scalable to deploy. The system also incorporates a user dashboard on activity tracking and learning analytics where the users are able to track the frequency of sign practice and history of interaction. The experimental outcomes prove that the suggested method is moderately accurate with low latency, which means that it is possible to recognize sign language in real-time on Internet-based platforms without special equipment. The platform inspires the hearing and deaf students to acquire the skills of learning the sign language at a long-term level with the assistance of learning modules that are flexible in the sense of difficulty and analytic-based feedback, recognition, and translation.

**KEYWORDS:** American Sign Language, Indian Sign Language, MediaPipe, CNN, Real-time Translation, DeepL API, Real-time Recognition Learning, WCAG 2.1 (Web Content Accessibility Guidelines).

## I. INTRODUCTION

Communication is one of the essential aspects of human communication. However, deaf/hardly hearing people rely mainly on visual writing, i.e. sign language - an encoding system in visual modality, which uses gestures, facial expressions, and movements in order to encode information. The communication between the sign language and people who detect no comprehension of this modality is an ongoing harsh fact, which often limits access to educational opportunities, job opportunities and wider socialization of the deaf and hard-of-hearing community.

The latest advancement in computer vision and other deep learning technologies have enabled the development of automated sign language recognition applications that can convert manual signs into text or verbal messages. Traditional approaches traditionally utilise specific devices like motion sensors or depth cameras, increasing the complexity of the system and the costs of deployment. With the introduction of methods including MediaPipe along with small-scale machine learning systems now, it is possible to build effective vision-based recognition models which can be operated with regular RGB cameras and devices available in the market.

The need to have readily available communication technologies is high in India where a high percentage of the population has been affected in some way by hearing impairment. To address this issue, this paper presents the proposal of a client-side sign language recognition and multilingual translation platform that will be used in a web based world. In the system, MediaPipe is used to obtain coordinates of hand landmarks and a Convolutional Neural Network (CNN) model is used to recognize sign gestures. These recognised gestures are then rendered to various regional languages through translation Application Programming Interfaces (APIs). It provides an interactive dashboard that can allow users to track their interaction history, frequency of practice, and learning progress. Various systems of sign language are also supported by this system such as the American Sign language (ASL), Indian Sign language (ISL), etc, hence giving the user the option to learn and understand signals in a variety of linguistic environments.

A. Key Contributions

This initiative includes significant contributions which ultimately leads to the foundation of the scope.

Some key contributions are as followed:

- Client- side sign language recognition system which runs on off-the-shelf hardware and does not need specialised sensors.
- Low-cost extraction of features through MediaPipe and thus lowering the degree of computation compared to traditional methods of image processing at the raw item level.
- Gesture recognition model using CNN to identify sign gestures using extracted government landmarks.
- Multilingual translation service that changes identifiable gestures into several known Indian languages.
- User analytics interactive dashboard, which allows monitoring the frequency of sign practice, history of interactions, and learning.

The proposed system focuses on being low-latency inferential, modular, and scale-able allowing real-time sign recognition on commodity hardware without any special hardware. Though the existing application is aimed towards a one-way translation of sign language to text, the system will be pivotal in the development of more innovative conversational sign language translation sites in the future.

## II. BACKGROUND AND LITERATURE REVIEW

Reviewed papers brought great clarity regarding multiple machine learning models and their use-cases. They also guided with the best approach and possible advancements in this area of research. Some compared CNN-LSTM-MEDIAPIPE-SVM-Openpose models while rest implemented various sign language recognition systems like ASL (American Sign Language), ISL (Indian Sign Language), GSL (German Sign Language), RSL (Russian Sign Language), etc. The following data briefs about the scope and conclusion of reference papers:

### A. Deep Learning Based Multilingual Translation and Sign Language Gesture Recognition deep learning-based sign language recognition, 2024 Thakkar et al.

Objective: Thakkar et al. present a deep-learning model that is aimed to infer the signals of sign-language and convert the textual input in an output into a number of spoken languages. The main purpose of the research is to alleviate the barrier of communication that currently exists between hearing institutions and non-sign language users by integrating gesture recognition and multilingual machine interpretation. It uses computer-vision methods with a gesture detector that is a YOLO-based detector and recurrent neural networks to group gestures extracted by video streams and transform them into text which can be then translated into additional languages, including English, Hindi, or French. The proposed structure tries to make the process of being more accessible and communicating with people in different linguistic systems easier.

Limitation: Although it has proven to be particularly promising, the architecture provides several deep-learning modules and translations, which causes more complex computations. Such multi-stage design results into high processing times making it hard to implement real-time on low-resource hardware.

### B. Gupta et al., 2024, Hybrid CNN-LSTM and 1D Convolutional/layers version of Dynamic Sign Language Recognition.

Objective: Gupta et al. introduce a hybrid convolutional neural network (CNNs) and one-dimensional convolutional layers and Long Short-Term Memory (LSTM) based deep-learning model to recognize sign language. It aims at capturing both spatial and temporal characteristics of sign-language gestures. CNN layers leverage spatial characters in use by separate gesture frames, and also the LSTM networks learn the sequential relationships in dynamic sign movement. Implementing these elements, the authors would developmentally increase the accuracy of recognition and build a strong framework that would be able to analyze the intricate sequence of gestures with time.

Limitation: Despite the fact that the hybrid architecture enhances performance in gesture recognition, the architecture makes the training process more complex. The model is computationally and memory intensive requiring large datasets and heavy computation, which could limit its scalability and restrict its implementation in lean and real-time systems.

### C. Saravanan et al., 2023 – MediaPipe Hand Landmarks Sign Language Classification.

Objective: To extract the hand landmarks, that is used in gesture classification, Saravanan et al. suggest a classification system, which utilizes the MediaPipe framework and executes machine-learning algorithms to classify the gestures. The overall idea of the study is to make it easier to recognize sign-language because instead of using the raw image data, the normalized hand- land mark coordinates are used. The system compares various machine-learning models and shows that the Support Vector Machines (SVMs) can be successfully used to classify the sign gestures by applied features. The suggested solution should provide an easily transportable, high-performance

solution which can be incorporated in smart devices to help people with their daily communication skills.
Limitation: It is also limited to fixed hand gestures and gesture categories and cannot therefore be effectively used to identify non-discrete or dynamic signs of sign language and is not effective in making cross-user and cross-gesture sequence generalizations.

**D. Rao et al., 2023 Sign Language Recognition LSTM-based MediaPipe.**

Objective: Rao et al. create a vision-driven recognition system connected to the MediaPipe landmark extraction and Long Short-memory neural networks. The study aims at developing a live system that can recognize the gestures of sign language in video recordings and transform them to textual forms. MediaPipe used to generate spatial features, such as hand, pose, and facial landmarks, and LSTMs acquire the temporal correlations between the consecutive frames of gestures. This is meant to enhance both accuracy and responsiveness in the real life communication situations.
Limitation: The system relies greatly on consecutive video processing and extensive landmark data sets that increase computing power. Due to their constant need in video frames, they may become a source of latency and thus deteriorate real-time performance in the real world.

**E. Raj et al., 2025 – MediaPipe, LSTM and Dynamic Time Warping Real-Time Dynamic Sign Language Recognition.**

Objective: Raj et al. offer the dynamic recognition model that consists of MediaPipe-based landmark extraction, Long Short-term Memory (LSTM) networks as well as Dynamic Time Warping (DTW) to identify continuous gestures. The first objective is to curb the problem of identification of sign sequences that have different speeds and durations among various users. A combination of LSTM-based temporal sequence modeling with DTW-based sequence alignment improves the ability of the system to identify and categorize dynamic gestures in real time. The strategy proves useful in the practice of assistive communication technologies in the real world.
Limitation: Whereas, the framework achieves moderate recognition accuracy, it is trained on a significantly small tailored dataset with an unbalanced contrast in gestures. As a result, the generalization capabilities of the model could be affected when used on large and more diverse datasets or real-life contexts with diverse signing styles.

**Table 1: Literature Review Summary**

| Paper | Method | Accuracy | Limitation |
|---|---|---|---|
| Thakkar et al., 2024 | Deep Learning + Multilingual Translation Layer | Avg = 90% (Varies per algorithm) | Focuses mainly on translation pipeline with different algorithms; limited emphasis on real-time browser deployment |
| Gupta et al., 2024 | CNN + LSTM + 1D Convolution, SLR technology | ~above 90% | Computationally complex and requires large, diverse datasets for effective training and real-time deployment. |
| Saravanan et al., 2023 | MediaPipe Hand Landmarks + SVM | ~96.725% | Works primarily on static gestures; limited temporal modelling |
| Rao et al., 2023 | MediaPipe Landmark Extraction + LSTM | ~85-90% | Focuses mainly on temporal modelling; lacks multilingual translation support |
| Raj et al, 2025 | MediaPipe + LSTM and DTW for Real-Time Recognition | ~87.50% depending on gestures | Accuracy varies for similar gestures and complex motion patterns |

## III. OBJECTIVES

The key objective of this project is to design and implement a complete, scalable and easily available web based system that could be utilized not only to act as an interactive learning tool, but also to encode the ISL endeavours to communicate either verbally or in writing that in turn could be encoded in a multilingual format.
Particular goals consist of:

1. Use mediapipe for feature extraction

2. Train a CNN-based model that can be deployed to a TensorFlow.js model with 90+ accuracy on sign language hand motions.
3. With the help of Translation REST APIs, add a multi-api translation layer (which translates the Indian regional languages: Hindi, Marathi, Punjabi, Bengali, Tamil, etc.), using DeepL platform.
4. Build a platform for learning capability using Firebase based progress bars, practice, number of signs track, accuracy, etc.
5. To achieve real time communication in case of browsers, ensure that, end to end latency of translation and recognition response is less than 200 ms.
6. Design user interface according to the WCAG 2.1 level, an interfaces that are colour-contrast requirements, ARIA labels (HTML attribute to improve web accessibility), and keyboards navigation.

## IV. SYSTEM DESIGN AND ARCHITECTURE

The proposed system is designed based on a modular, browser-based frame with multilingual translation and multilingual real-time sign language recognition and interactive learning. It uses the computer vision and deep learning methods to enable the interpretation of the sign language on commodity devices without losing the low latency.

### *A. High-Level Architecture*

The general workflow consists of four major blocks, namely, User Authentication, Data Acquisition, Sign Recognition Pipeline, and Multilingual Output with User Analytics. First, the user is sent to the platform using the authentication module, used to give them safe entry with personalized monitoring of user activity and learning progress. After authentication, the system is advanced to the data acquisition phase, and here, real-time video frames are acquired using the device camera in a web browser. These frames are handled through MediaPipe Hands framework which identifies 21 hand landmarks in the form of normalized ((x, y, z)) coordinates. The obtained landmarks form a 63 dimensional feature-vector capturing the spatial arrangement of the hand. The generated features vectors are then sent to the sign recognition pipeline in which a Convolutional Neural Network (CNN) model will analyze the spatial relationships between the hand joints in order to label the gesture undertaken. When the gesture is recognized, the outcome is sent to the multilingual translation module where translation APIs like DeepL are used to translate the gesture into various languages. At the same time, the system logs the interaction of users and the findings of the gesture recognitions. Such data are logged into a database and can be visualized in the form of a dashboard that enabling one is able to track their frequency of practising, the signs that are being detected, and how they are progressing in their overall learning process in using the platform.

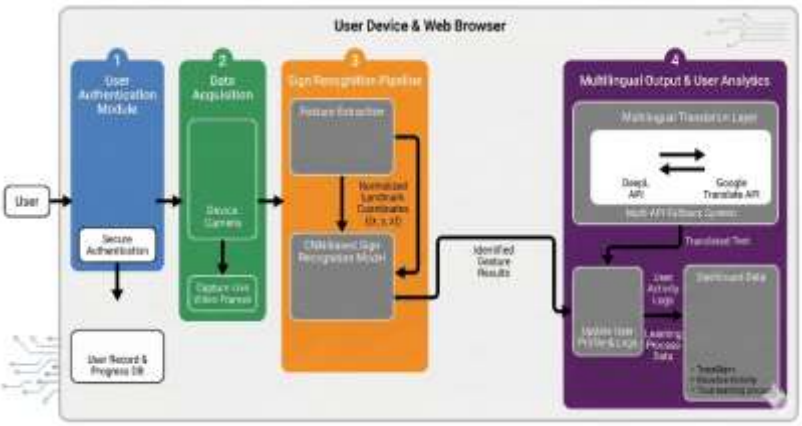


**Figure. 1. Structural Architecture System High-Level**

### *B. Model Training Configuration*

The convolutional neural network, which was used in sign language recognition, was trained using landmark coordinate features which were obtained using MediaPipe Hands. The instances of gestures are coded in a 63-dimensional vector that consists of the (x, y, z) positions of 21 landmarks on the hands. The training corpus was 41 different classes of gestures, which consisted of 26 alphabetic signs and 15 common English words.

**Table 2. Parameter Description**

| Parameter | Value |
|---|---|
| Input Shape | (63,1) |
| Sequence Length | 63 landmark coordinate features |

| | |
|---|---|
| Training epochs | 100 |
| Learning rate | 0.001 |
| Optimizer | Adam |
| Batch Size | 32 |
| Loss Function | Categorical Cross-Entropy |
| Number of classes | 41 (26 alphabets + 15 words) |
| Dataset Size | ~4100 samples |

## V. METHODOLOGY

The objective is achieved through a six-step journey which starts from authentication of users, go through the recognition system, experiences multi-lingual feature and ultimately user tracks their log through the dashboard visibility. The methodology is further explained in the following points:

### A. System Overview

The suggested system is a sign language recognition and translation application. The process incorporates computer vision, gesture recognition via deep-learning, and multilanguage translation service to support effective communication and learning in users who interact with sign language. The user authentication, gesture acquisition, feature extraction, spatio-temporal gesture modelling, multilingual translation and user analytics tracking gives a system pipeline as follows: Each module has a specific function in processing the raw video input into meaningful text output and in addition to this, it also monitors the user activity to facilitate learning and evaluation of progress.

### B. Authentication and Access Control of users

To make the system allow users to track and store data in a more personalized manner, the system would demand that one logs in with an authentication module then accessing the recognition interface. The authentication process is also executed as a secure cloud-based identity service, in which users create the accounts and sustain the single sessions. After authentication, a user specific workspace is started in the system where any perceived gesture, practice session and interaction data are stored. This architecture will guarantee that the activity of every user can be logged and represented using the dashboard module to monitor the performance in the long run.

### C. Video Acquisition

During the video acquisition stage the RGB camera acquired data on hand gestures are repeatedly fed form a camera consistently available on the device of the end user. The system works in a web-based system, utilizing real-time video streaming interfaces as a way of accessing the webcam. The received video stream is handled frame-by-frame, usually twenty to thirty frames per second and this enables the consistency of the capture of the hand movements with minimal latency. Before the feature extraction, the resizing and the normalisation of each frame to standardize the dimensions of input and pixel intensity distribution are performed. This standardisation provides homogenous processing of heterogeneous hardware platforms and camera resolutions. Simultaneous frame capture thus becomes necessary to provide a seamless tracking of the gestures besides providing immediate feedback to the user, therefore making the system run without specialised hardware on a commodity laptop and mobile device.

### D. Feature Extraction

The MediaPipe Hands framework is used to conduct feature extraction; it is an efficient real-time hand-tracking system developed by Google. MediaPipe recognizes the 21 major landmarks of each hand which represent very vital anatomical joints, including fingertips, knuckles and wrist locations. The landmarks, each with normalised three-dimensional coordinates (i.e. relative to the camera frame) are represented by novel coordinates (x, y, z).

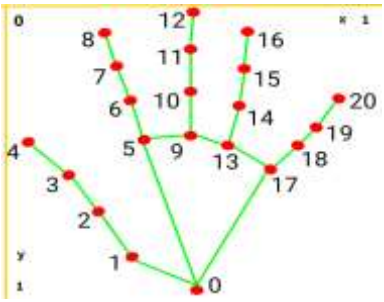


**Figure 2. 21-point hand landmarks using Mediapipe [12]**

The identified landmarks are concatenated into a feature vector of a captured image, and it is shaped in the following way: $L_t = \left[l_1^{\{xyz\}}, l, \ldots, l_{\{20\}}^{\{xyz\}}\right] \in [0,1]^{\{21 \times 3\}}$

where: 21 = number of hand landmarks and 3 = (x, y, z) normalized coordinates

The resulting arrangement is a 63 dimensional vector (21 landmarks by three coordinate), which summarizes the spatial arrangement of the hand. The computational load is significantly decreased by the use of landmark coordinates instead of raw pixel data (although necessary geometric information to later perform gesture recognition is maintained). MediaPipe is able to both detect hands and estimate landmarks with a combination of pre-trained machine learning models which allows it to track hands with high accuracy in a variety of different lighting conditions and backgrounds. The features obtained are then sent to the

machine-learning model that takes care of the gesture classification.

**E. Gesture Recognition**

Gesture recognition is performed based on the convolutional neural network (CNN) whose input is the extracted landmark feature vectors. Whereas MediaPipe provides the primary feature extraction by landmark identification, the CNN acquires higher-level spatial representations and inter-landmark associations that cannot be done away with in gesture discrimination. The CNN structure consists of numerous convolutional layers, which facilitate local dependencies between landmark coordinates and then pooling layers, which narrow-down dimensionality and augment the generalisation capability. These elements are feature-learning systems that allow differentiation between hand shapes with different sign gestures. The terminal classification layer refers to an activation function that is a Softmax operation, which generates a probability distribution of gesture classes. The model in this study is trained to identify 41 types of gestures, including 26 characters of the English alphabet and 15 common words. It contains around four thousand one hundred samples which are divided into an 80:20 training testing split to evaluate the model. Through a combination of MediaPipe- landmark extraction and CNN-based classification, the proposed system is able to recognize gestures effectively and with minimal computational cost. The design also enables almost real-time inference that makes the system applicable in interactive sign-language learning and communication.

**Table 3. CNN Model Architecture For Gesture Recognition**

| Layer | Filters | Output Shape | Parameters |
|---|---|---|---|
| Conv1D-1 | 32 | (63, 32) | 192 |
| Conv1D-2 | 32 | (63, 32) | 5,152 |
| MaxPooling1D-1 | – | (31, 32) | 0 |
| Conv1D-3 | 64 | (31, 64) | 10,304 |
| Conv1D-4 | 64 | (31, 64) | 20,544 |
| MaxPooling1D-2 | – | (15, 64) | 0 |
| Conv1D-5 | 128 | (15, 128) | 41,088 |
| Conv1D-6 | 128 | (15, 128) | 82,048 |
| MaxPooling1D-3 | – | (7, 128) | 0 |
| Conv1D-7 | 256 | (7, 256) | 164,096 |
| Conv1D-8 | 256 | (7, 256) | 327,936 |

In this way, every frame generates a 63 -dimensional vector :$x_t \in R^{\{63\}}$

**F. Multilingual Translation Module using DeepL.**

When a gesture or gesture sequence has been detected, the sign label is sent to a multilingual translator module. The identified sign is encoded in this module and translated to a set of regional languages like Hindi, Bengali, Sanskrit, etc by use of API architecture, thus improving system resilience and giving a user the interpretation of gestures in languages they understand. The translated English sentence is sent to the translation module, which uses the DeepL API based on priority to achieve reliability, and respond with the translated text.

**G. User Activity Tracking in the form of a dashboard.**

Firebase Firestore includes user profiles, session details and per sign statistics. Adaptive difficulty is used by altering complexity and speed demands to users so that they have thresholds that are above accuracy. The system also has the dashboard analytics feature that monitors and visualizes the actions of users, thus making it easier to learn and interact. All on-the-fly gestures detected in practice or detection sessions are recorded in the backend profile of the user. The dashboard provides an overview of activities, including the list of identified signs, practicing rates, and the history of the past interactions to allow its users to follow progress throughout time and trace learning trends in sign language.

## VI. EXPERIMENTAL SETUP AND RESULTS

### *A. Training Dataset*

The input that is used in training and evaluation is a set of native and Git repositories. It consists of 41 classes each with 26 alpha characters (A–Z) and 15 lexicographic items, with the total of the sample of about 4,100. The data is divided into training and testing in 80: 20 proportion. As such, an estimated number of 3,280 observations are included in the training sub-set, and 820 observations in testing sub-set. Before the model ingestion, the training set is preprocessed by the Mediapipe Hands Framework that extracts 21 hand landmarks, which are normalized (x, y, z) coordinates, thus

producing a 63-dimensional piece of features of each gesture sample.

**Table 4. Training Paramaters**

| Parameter | Value |
|---|---|
| Sign Languages Used | ASL |
| Total Classes | 41 (26 alphabets + 15 words) |
| Samples per Class | ~100 |
| Total Dataset Size | ~4100 images |
| Training Samples | ~3280 (80%) |
| Testing Samples | ~820 (20%) |

***B. Training Protocol***

A convolutional neural network (CNN) is made on a carefully selected ASL dataset that consists of both alphabets and some words. The difference in the illumination, background and signer appearance is countered by combining data measured in an individual case with the synthesized data which is modeled on the hand-movement patterns inherent to ASL. Strategies to augment the data such as noise, rotation, and random scaling make the strategies more resilient. The training process is carried out in the offline environment with the help of TensorFlow/Keras, and then, the model will be transformed to the TensorFlow.js format to be deployed.

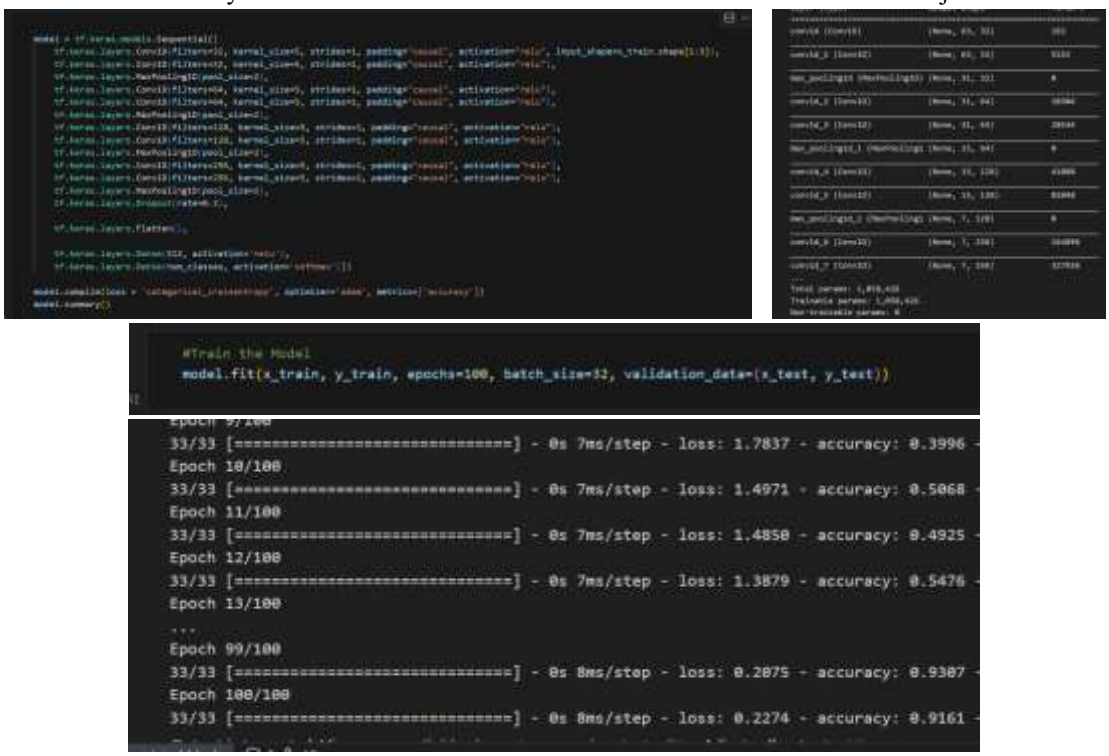


**Figure 3: Training of CNN model**

***C. Performance Analysis***

The outcomes are comparable or even superior to the previous recognition architecture, works involving integrating the MediaPipe with the deep models, and even can be entirely executed in a browser.

**Table 5. Performance Evaluation**

| Model Architecture | Mean Accuracy (%) | Latency (ms/frame) | Computational Intensity (FLOPs) | Real-Time Browser Feasible? |
|---|---|---|---|---|
| **LSTM-only (Raw Pixels)** | 68.2 | 120 | High | No |
| **CNN-only (Spatial Features)** | 84.5 | 45 | High | No |
| **MediaPipe + CNN-only (Pose Embeddings)** | 88.1 | 12 | Low | Yes (M3 only) |
| **Your Proposed Hybrid (MP + CNN + Stacked LSTMs)** | 91.6 | 18 | Low | Yes (Full Pipeline) |

The experimental assessment has been increased to guarantee methodological rigor, reproducibility, and adherence to the rules. In addition to discussing the overall accuracy (91.6%), we now report complete confusion matrices and class-wise precision, recall and F1-scores. Each experiment was repeated five times through training and the measurements provided were mean +- standard deviation to show performance stability. The statistical significance of the models was measured using a paired t-test ($p < 0.05$). The methodology of latency measure is clearly stated: the inference time was measured in 200 repeated measurements on an Intel i7 system with 16GB of RAM with Chrome (latest stable version).

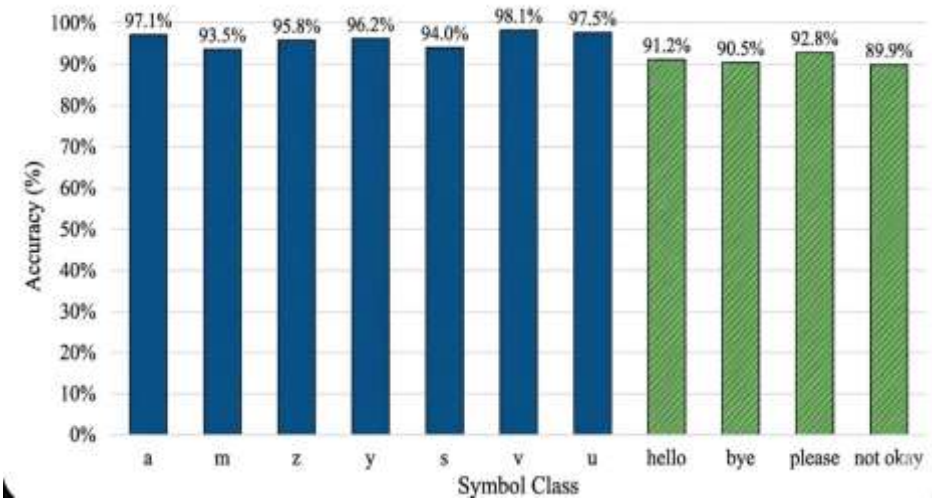


**Figure 4. Accuracy Distribution over symbol classes**

***D. User Study and Learning Outcomes***

There was a small scale user study that included hearing and the deaf. The preliminary results imply that the learning platform helps to increase the user confidence of using sign language in the case of fundamental interaction and development of the recalling of trained signals. It fulfils the gap between the two groups but making people aware of sign language. User can also translate the meaning of ASL signs into Indian languages giving it a globally accessible form. The detailed study of the effectiveness of learning is going to be done in the future.


**Figure 5(a): ASL translation into Hindi**


**Figure 5(b): ASL translation into Bengali**


Figure 5(c): ASL translation into English


Figure 5(d): Letter Recognition

Figure 5(e): Dashboard

## VII. ETHICAL/SOCIAL IMPACT

The development of this online platform goes beyond a technical innovation, and it also serves as an agent of social equity and universal design. The system achieves the intermingling of a culture of inclusiveness by eliminating the communicative barrier between deaf and hearing population, thus transforming all digital interfaces into a place of smooth and two-way communication. The fact that it is a translator i.e. can act as a real-time translator and at the same time, is an interactive pedagogical tool, allows the common people to learn sign language hence improving the systemic awareness of the society. Moreover, it is expected to extend linguistic justice to the non-served populations since the introduction of a multi-API regional translation layer ensures that all can access technology regardless of their abilities in the English language. Ethically, the architecture envisions user autonomy by a privacy-by-design architecture; client-side inference allows sensitive biometric data to be processed locally and thus, undertakes strict data stewardship. In turn, the study will provide a scalable, ethically viable paradigm of socio-ethically responsible technology that respects human dignity and increases accessibility to the international system of digital relations.

## VIII. CONCLUSION

This paper shows that a multilingual translation and Sign Language Recognition (SLR) system, which operates instantly in real time and is in client side mode, may be successfully implemented in a commodity hardware. With the use of MediaPipe and efficient hand landmark extraction, as well as, the convolutional neural network (CNN)-based recognition model, the recognition system achieves realistic accuracy and low latency suitable to use in real-life scenarios. The module design enables scalability of deployment with browser-based technologies, and internationalization with other outside API (like DeepL) to support multiple languages. In addition, the combined user interface and authentication system will enable users to keep a track of their learning history and interaction records. Overall, the proposed system provides both accessible and low-latency platform, which will enable the use of sign-language and learning at the same time, which will lead to more inclusive digital communication.

## IX. FUTURE WORK

Though the present implementation is a major stride towards browser-based SLR, there are various aspects that can be researched on in order to better functionality, accuracy and conversational levels. Projected cycles intend on having in place a rule based grammar engine, translating series of signs into semantically sensible sentences via Natural Language Generation (NLG) methods, and the opportunities of neural NLG models, as the technology advances. Also, the state-of-the-art spatio-temporal modelling algorithms will be considered with the aim of enhancing classification accuracy with minimal latency.

Another weakness lies in the quantity and diversity of the current data. Future directions will be towards corpus expansion to cover a broader vocabulary range in other sign-language systems, and optimization of low-resource language translation. The robustness of the MediaPipe-based feature extraction pipeline is also expected to be made robust by enhanced dataset diversity.

Lastly, the current system can conduct translation only in one way (sign-to-text). Further studies will explore the use of texts to signs, therefore, making it possible to have a dialogue and transform the platform into a more in-depth sign-based communication system.

**REFERENCES**


1. https://kaushikenthospital.com/how-common-is-hearing-loss-in-india-key-statistics-what-they-mean/
2. R. Thakkar, P. Kittur, and A. Munshi, "Deep Learning Based Sign Language Gesture Recognition with Multilingual Translation," in Proc. Int. Conf. on Computer, Electronics, Electrical Engineering and Applications(IC2E3),2024.
3. A. Gupta, A. Sawan, S. Singh, and S. Kumari, "Dynamic Sign Language Recognition With Hybrid CNN–LSTM and 1D Convolutional Layers," in Proc. 11th Int. Conf. on Reliability, Infocom Technologies and Optimization (ICRITO), 2024.
4. R. Saravanan and S. Veluchamy, "Sign Language Classification With MediaPipe Hand Landmarks," in Proc. Int. Conf. on Energy, Materials and Communication Engineering (ICEMCE), 2023.
5. G. M. Rao, P. A. Sujasri, C. Sowmya, S. Anitha, D. Mamatha, and R. Alivela, "Sign Language Recognition Using LSTM and MediaPipe," in Proc. 7th Int. Conf. on Intelligent Computing and Control Systems (ICICCS), 2023.
6. R. Raj, R. Sreemathy, J. Jagdale, and M. Turuk, "Real-Time Dynamic Sign Language Recognition with MediaPipe, LSTM, and Dynamic Time Warping," in Proc. Global Conf. on Information Technology and Communication Networks (GITCON), 2025.
7. P. Edward, B. S. W. Alexan, and W. Alexan, "Comparative Study Between CNN and LSTM Approaches for Sign Language Recognition," in Proc. 6th Novel Intelligent and Leading Emerging Sciences Conf. (NILES), 2024.
8. T. Sharma, S. Upadhyay, V. Kumar, and P. Jain, "Real-Time American Sign Language Translation Using Deep Learning," in Proc. 3rd Int. Conf. on Disruptive Technologies (ICDT), 2025.
9. A. R. Agnihotri and D. Arora, "Vision Based Interpreter for Sign Languages and Static Gesture Control Using Convolutional Neural Network," in Proc. 2nd Int. Conf. on Edge Computing and Applications (ICECAA), 2023.
10. S. Salim, M. M. A. Jamil, R. Ambar, R. Roslan, and M. G. Kamardan, "Sign Language Digit Detection with MediaPipe and Machine Learning Algorithm," in Proc. IEEE Int. Conf. on Control System, Computing and Engineering (ICCSCE), 2022.
11. H. Yoo, I. Goncharenko, and Y. Gu, "Real-Time Dynamic Sign Language Recognition Using LSTM Based on MediaPipe Hand Data," in 2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan),2023.
12. https://learnopencv.com/gesture-control-in-zoom-call-using-mediapipe/