INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Check for updates

OPEN ACCESS

Received: 19-12-2024

Accepted: 05-02-2025

Published: 28-02-2025

Citation: Mishra R, Angne G, Gawde N, Khamkar P, Utekar S (2025) SignSpeak: Indian Sign Language Recognition with ML Precision. Indian Journal of Science and Technology 18(8): 620-634. https://doi.org/10.17485/IJST/v18i8.4049

* **Corresponding author**.

ravita.mishra@ves.ac.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

SignSpeak: Indian Sign Language Recognition with ML Precision

Ravita Mishra[1*], Gargi Angne[2], Nidhi Gawde[2], Preeti Khamkar[2], Sneha Utekar[2]

1 Assistant Professor, Department of Information Technology, Vivekanand Education Society's Institute of Technology, Mumbai, 400074, Maharashtra, India
2 Department of Information Technology, Vivekanand Education Society's Institute of Technology, Mumbai, 400074, Maharashtra, India

Abstract

Objectives: To develop an accessible educational platform for Indian Sign Language (ISL) recognition, bridging communication gaps using advanced machine learning techniques, and promoting inclusivity for the hearing-impaired community. **Methods:** The study utilized Random Forest for classifying ISL letters and numbers with 1200 images per class and Long Short-Term Memory (LSTM)/Large Language Model (LLM) for gesture-based word and sentence recognition using 120 custom images. Feedback from Jhaveri Thanawala School for the Deaf validated the approach. **Findings:** The Random Forest model achieved 99.98% accuracy in recognizing ISL letters and numbers. LSTM and LLM models demonstrated 87% accuracy in translating gestures into meaningful sentences. The dynamic learning and quiz modules improved user engagement, facilitating effective ISL mastery. Feedback from Jhaveri Thanawala School confirmed its real-world usability. These results enhance prior works by offering an integrated, highly accurate platform to promote ISL adoption, enabling better societal inclusivity for individuals with hearing disabilities. **Novelty:** Signova integrates gesture recognition with sentence generation which makes it a unique ISL recognition system, achieving high accuracy while providing interactive tools for learning and practicing ISL to foster its accessibility.

Keywords: Indian Sign Language; Sign-Language Recognition; Random Forest; Long Short-Term Memory; Large Language Model

1 Introduction

India has a diverse linguistic landscape, in which Indian Sign Language (ISL) plays a crucial role in communication for the deaf and hearing-impaired community. Like other languages, it is an independent language with complex grammar. However, the communication gap between ISL users and those unfamiliar with sign language presents significant challenges to daily interactions and social inclusion.

By 2050, nearly 2.5 billion people are projected to have some degree of hearing loss, and at least 700 million will require hearing rehabilitation. Over 1 billion young adults are at risk of permanent hearing loss, which is avoidable and caused by unsafe listening practices. In India, there are significant discrepancies in the count of hearing-impaired individuals: the 2011 Census reported five million, the National Association of the Deaf counts 18 million, and the World Health Organization estimates nearly 63 million. Despite these numbers, only 5% of deaf children are in school, and deaf adults struggle to secure employment [1]. It has been argued that while India's government has focused heavily on modernizing the country with technological resources and infrastructure, the needs of the DHH residents of India have yet to be addressed [2,3].

The promotion of sign language is essential. A census conducted in 2011 revealed that out of 13.4 million individuals between the ages of 15–59 with a hearing disability in India 73.9% were marginal workers. It is challenging for people of the DHH community to obtain an education, as they are unable to learn basic skills that the hearing population would learn at a young age [3,4]. It is a fundamental right of the deaf person to acquire information through Sign Language [5].

Our system aims to provide a free, efficient, and accurate way to convert ISL into text, aiding the hearing-impaired community and facilitating communication with others. The objectives include enhancing communication accessibility and inclusion for individuals with speech and hearing disabilities and promoting awareness and recognition of Indian Sign Language (ISL). Addressing the lack of widespread ISL knowledge among the hearing population and the limited availability of accessible ISL translation tools, this initiative aims to connect the deaf and hearing-impaired community and foster learning and interaction with ISL. The system attempts to fill the void by providing a free educational platform that provides real-time translation with learning and practice modules since such a platform does not exist currently.

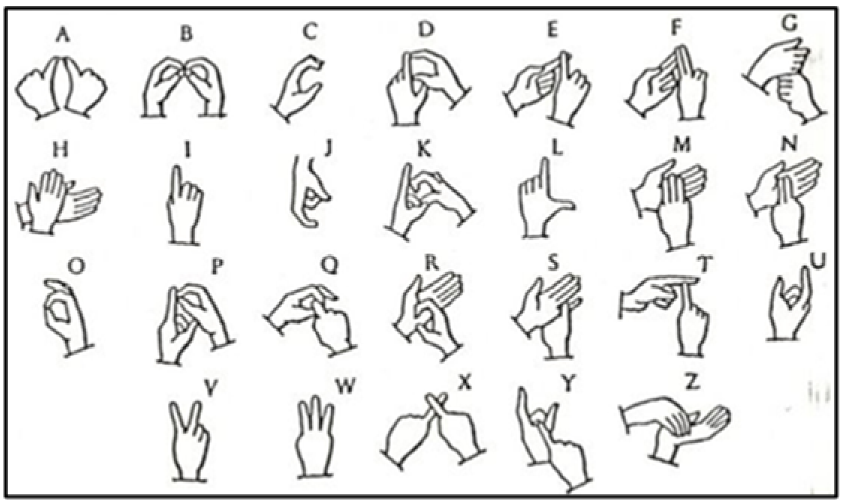


Fig 1. Signs for alphabets in Indian Sign Language

2 Literature Survey

Table 1 describes the advantages and disadvantages of the existing papers. The significant difference is that these advancements are mainly in other sign languages rather than Indian Sign Language.

Table 1. Pros and Cons of the above-discussed papers

Sr. No.	Title of the Paper	Author and Year	Advantages	Disadvantages
1.	A Survey on Sign Language Recognition Systems	Shruty M. Tomar, Dr. Narendra M. Patel, Dr. Darshak G. Thakore, 2021 [6]	Wide range of approaches and techniques for sign language recognition, including CNNs, LSTM, RNN, and hybrid models like CNN-HMM.	Dependency on Dataset Composition. Sensitivity to Skin Tone Differences
2.	Problems in the application of Google Translate as a learning media in translation	Brahmana et al. (2020) [7]	Easy to use and integrate	The translation is inaccurate and inappropriate sometimes and requires internet connectivity

Continued on next page

Table 1 continued

3.	Indian Sign Language recognition system using SURF with SVM and CNN	Shagun Katoch et al. (2022) [8]	Accurately recognizes ISL alphabets and numbers, improving communication with text and audio outputs	Struggles with diverse gestures and lacks a comprehensive ISL dataset, impacting accuracy.
4.	Recognition of Indian Sign Language (ISL) Using Deep Learning Model	Sakshi Sharma et al. (2022) [9]	High accuracy with a robust CNN and diverse dataset, enhancing real-time sign language recognition.	Depends on dataset quality, is sensitive to environmental changes, and requires significant computational power.
5.	An integrated media pipe-optimized GRU model for Indian sign language recognition	B. Subramanian et al. (2022) [10]	Achieves high accuracy and efficient real-time hand gesture tracking with an optimized GRU and MediaPipe, enhancing learning and usability.	Performance depends on dataset quality, is sensitive to environmental changes, and requires significant computational resources.
6.	Continuous Sign Language Recognition System Using Deep Learning with MediaPipe Holistic	Sharvani Srivastava et al. (2024) [11]	The use of LSTM networks allows for continuous recognition of signs, making it suitable for live interactions.	The accuracy of sign recognition may be affected by varying lighting conditions, backgrounds, or occlusions, which can pose challenges in real-world applications.
7.	Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling	Nandi et al. (2023) [12]	Achieves high accuracy with optimized learning and scalability for real-time sign language recognition, enhanced by diffGrad and stochastic pooling.	Performance depends on dataset quality, is limited to static gestures, and requires significant computational resources.
8.	Enhanced Deep Semantic Structure Modelling (EDSSM) in Job Recommendation.	Ravita Mishra et al. (2021) [13]	Achieved 94.5% accuracy and 3% performance improvement.	Scalability depends upon the dataset size and features.
9.	Inductive Learning Approach in Job Domain, Advanced ILA	Ravita Mishra et al. (2023) [14–17]	Achieves high accuracy and Scalability in Different datasets.	Performance depends on dataset size and feature values and requires a distributed setup for scale data.
10.	Using an LLM to Turn Sign Spottings into Spoken Language Sentences	Ozge Mercanoglu Sincan and Necati Cihan Camgoz and Richard Bowden, 2024 [18]	The framework leverages pre-trained LLMs for accurate sign language translation, enhanced by sign-text alignment for improved semantic accuracy, achieving accurate performance on SLT benchmarks.	The framework's performance is challenged by video quality, limited paired sign-text data, and the inherent complexity of sign language, affecting its ability to capture accurate translations.
11.	Empowering Deaf-Hearing Communication: Exploring Synergies between Predictive and Generative AI-Based Strategies towards (Portuguese) Sign Language Interpretation	T Adão, J Oliveira, S Shahrabadi, H Jesus, M Fernandes, Â Costa, V Ferreira, MF Gonçalves Journal of Imaging, 2023 [19]	Combines LSTM and LLM models with data augmentation and buffer-based interaction for accurate sign language interpretation, leveraging ChatGPT for coherent sentence generation in an accessible web-based service.	Limited to 50 Portuguese Sign Language signs, may face real-time latency issues, and relies on ChatGPT, which has content filtering and accuracy limitations.

2.1 Problem Statement and Proposed Solution

Despite the growing recognition of Indian Sign Language (ISL) as a critical communication tool for the deaf and hearing-impaired community, its translation and recognition still face significant challenges. Current systems often struggle with accuracy, scalability, and accessibility due to issues like dataset quality, environmental variations, and the complexity of gestures. While machine learning techniques such as KNN, Random Forest, and CNNs have shown potential, they remain constrained by problems like overfitting, sensitivity to lighting, and reliance on extensive training data. Additionally, the lack of standardized datasets and comprehensive ISL resources widens the communication gap between deaf and hearing individuals, limiting societal inclusivity.

To tackle these challenges, our project aims to develop a robust and scalable system for real-time ISL-to-text translation. By leveraging advanced machine learning techniques and a well-curated ISL gesture dataset, we seek to enhance communication accessibility for individuals with speech and hearing disabilities. Furthermore, this initiative aims to bridge the awareness gap by promoting understanding and use of ISL among the broader population, fostering greater inclusion and interaction between deaf and hearing communities.

3 Methodology

To effectively tackle the challenge of recognizing and learning sign language, we propose developing a web-based solution with advanced features specifically designed to meet the needs of the deaf and hard-of-hearing community. The methodology below outlines various stages to create a comprehensive and user-friendly sign language recognition and education platform.

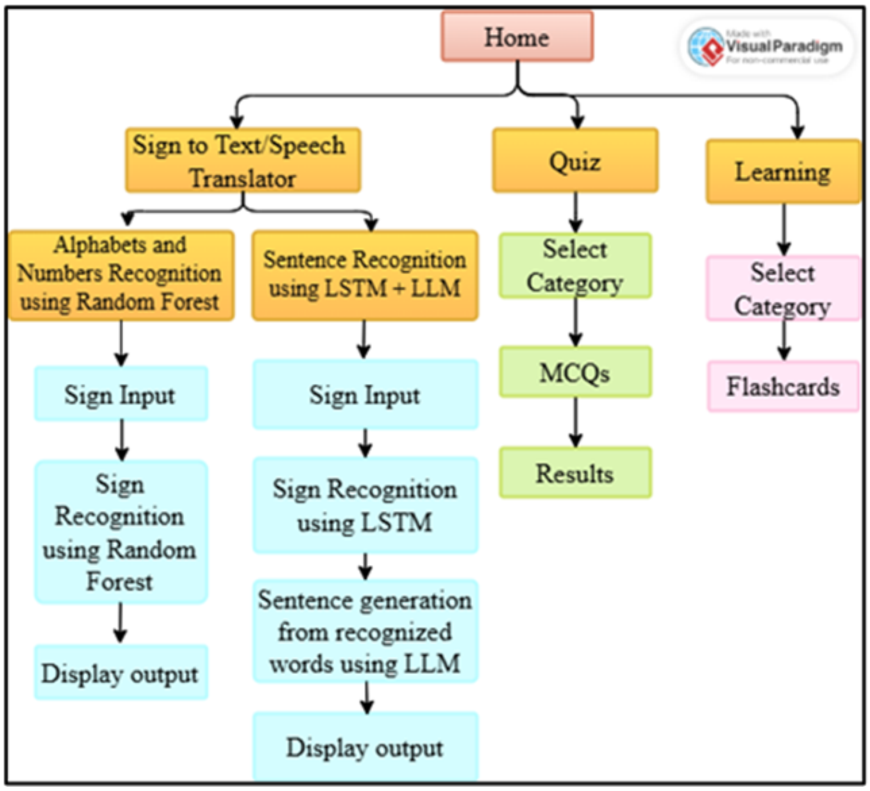


Fig 2. Flow diagram of the proposed system for the Indian Sign Language translating and learning platform

Figure 2 illustrates the flow of our proposed system, comprising three essential modules: the Sign-to-Text Translator, the Quiz Module, and the Learning Module. The Translator module allows users to input signs via their webcam, which are then converted into both text and speech. The Regions of Interest (ROI) are identified and tracked through skin segmentation using OpenCV and gesture capture with MediaPipe. The recognition process for letters and numbers begins when a user makes a hand gesture, which is captured and processed to classify them using the Random Forest algorithm. For word recognition, an LSTM model is employed, identifying individual words from gestures. Recognized words are then used to construct meaningful sentences via a Large Language Model (LLM), which outputs the corresponding text and speech. The Quiz Module enhances learning by offering multiple-choice questions across categories such as alphabets and numbers, allowing users to test their knowledge, review results, and learn from mistakes, making it an engaging and effective tool for reinforcing concepts. The Learning Module provides a comprehensive collection of categorized flashcards covering various concepts, ensuring effective revision and helping users build a strong foundation in Indian Sign Language (ISL). Together, these modules create a dynamic learning experience, combining self-assessment with targeted practice to support mastery of ISL.

3.1 Data Preparation

Data preparation involves collecting and preprocessing the raw data and ensuring that it is stored in the format suitable for training the model.

3.1.1 Data Collection and Creation

Our project uses two datasets for Indian Sign Language (ISL) recognition, focusing on alphabets, numbers, words, and sentence formation. The alphabet and number dataset, sourced from Kaggle (link:https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl/data) [20], contains 42,000 images across 35 classes (26 for letters A-Z and 9 for numbers 0-9), with 1,200 images per class. The dataset has variations in hand shapes, sizes, and signing styles, and preprocessing steps like normalization, quality enhancement, and augmentation techniques were applied to improve data quality.

For word and sentence recognition, a custom dataset was created with 18 commonly used words such as "hello," "I", "Indian" and "good morning." Each word was captured in 30 video instances to showcase variations in signing styles. MediaPipe's holistic model tracked 1,662 keypoints across the hands, face, and body, capturing detailed gesture movements, body posture, and facial expressions. These keypoints were stored in .npy files for efficient access during model training. By utilizing these datasets, our project supports accurate recognition of ISL alphabets, numbers, words, and phrases, enhancing inclusivity for the deaf community.

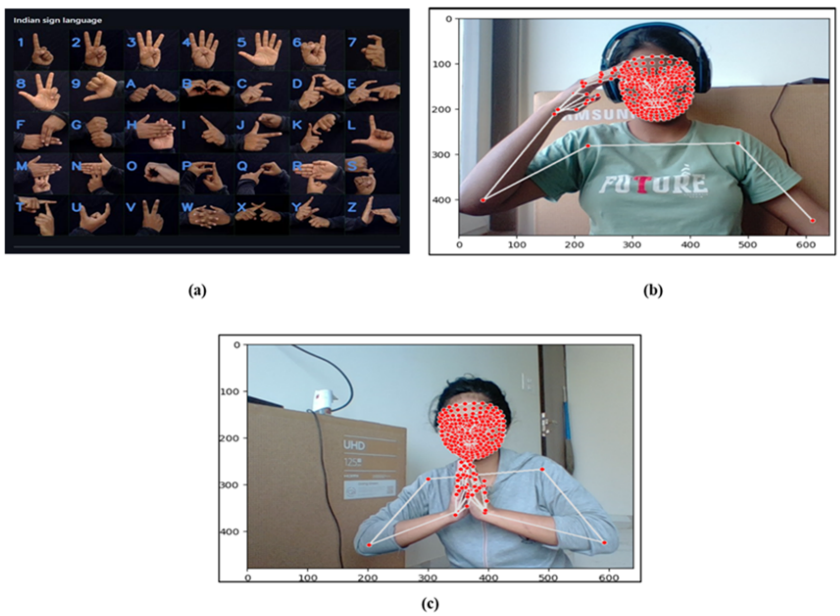


Fig 3. (a)Hand gestures for ISL alphabets (A-Z) and numbers (0-9), (b) & (c) MediaPipe keypoint graphs showing facial, body, and hand landmarks during gestures

3.1.2 Challenges in Real-Time Sign Recognition

Developing an Indian Sign Language (ISL) recognition system involves overcoming several challenges to ensure robustness and inclusivity. One significant challenge is environmental variability in real-world scenarios. Changes in lighting conditions, background clutter, and camera angles can significantly impact gesture recognition accuracy. For instance, dim lighting or highly reflective surfaces may obscure hand landmarks, while a complex background can confuse the model during gesture detection. Addressing this requires advanced preprocessing techniques, robust feature extraction, and training the models under diverse conditions to improve adaptability.

The Kaggle dataset for alphabets and numbers mainly represents respective hand shapes and signing styles, while the custom word dataset reflects the gestures of a small group of people. This limited diversity can affect the system's accuracy when recognizing gestures from different users or cultural backgrounds. Expanding the dataset to include more varied signing styles, hand sizes, and environments is important to improve the model's performance and real-world usability.

3.2 Alphabet and Number Recognition using Random Forest

This module is designed to recognize static sign language gestures representing alphabets and numbers by utilizing a Random Forest classifier that is trained on a diverse Kaggle dataset of hand gesture images, which are preprocessed for optimal feature extraction. By leveraging the power of Random Forest, which excels in handling complex datasets and feature importance, the model aims to achieve high accuracy and efficiency in classifying the various hand gestures ensuring reliable and robust performance for sign language recognition tasks.

3.2.1 Image Preprocessing

Image preprocessing enhances image quality, reduces noise, and prepares data for better algorithm performance. Common steps include resizing, normalizing, grayscale conversion, and applying filters. Data augmentation, like rotations and flips, increases diversity and improves model generalization, ensuring consistency and focus on key features.

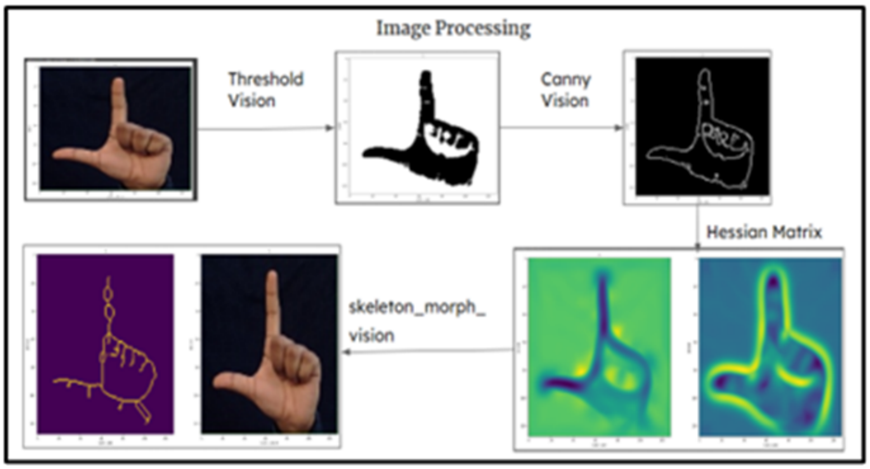


Fig 4. Image Preprocessing: Threshold Vision, Canny Vision, Hessian Matrix, and Skeleton Morph Vision

Figure 4 depicts the image preprocessing workflow comprising four key transformations: Simple Vision, Threshold Vision, Canny Vision, and Skeleton Morph Vision. Simple Vision begins by reading an image using cv2.imread and converting it from BGR to RGB with cv2.cvtColor, ensuring the image is compatible with computer vision tasks. Threshold Vision converts the RGB image to grayscale and applies a binary threshold at an intensity level of 90 using cv2.threshold with the cv2.THRESH_BINARY_INV option. This step simplifies the image for edge detection and feature extraction. Canny Vision enhances edge visualization by applying the Canny edge detection algorithm to the thresholded image, using thresholds of 10 and 100 to highlight edges while reducing noise. Lastly, Skeleton Morph Vision generates skeletonized images by converting the RGB image to grayscale, applying binary thresholding, and extracting skeletal frameworks. This captures object contours and central lines, which are useful for structural and morphological analysis. Together, these techniques provide robust preprocessing for computer vision tasks.

3.2.2 Model Selection

The keypoints for hands were extracted from images and then passed to the models as input for training and testing. Multiple models i.e. decision tree, k-nearest neighbor, and random forest were examined for their accuracy and robustness on this dataset. The evaluation and results of these models are discussed in **section 4.1**.

The Random Forest Classifier was chosen for alphabet and number recognition for its robustness, accuracy, and ability to handle complex datasets. This ensemble learning method combines predictions from multiple decision trees, reducing overfitting and improving generalization. Hyperparameters such as the number of trees (n_estimators) and bootstrap sampling were carefully tuned to optimize model performance. The optimal number of trees we got is 25 for this dataset.

3.2.3 Training and Testing

The real-time hand gesture recognition pipeline consists of six key steps, combining computer vision and machine learning techniques for accurate gesture identification. Figure 5 illustrates this process.

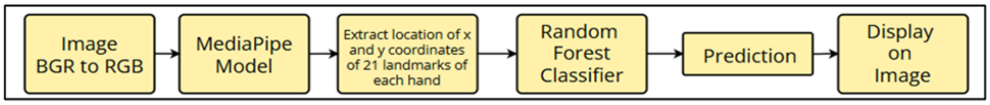


Fig 5. Pipeline model showing the flow of real-time sign recognition using Random Forest Model

The first step converts the input image from BGR (Blue-Green-Red) to RGB (Red-Green-Blue) color space, as shown in Figure 4. This transformation is essential because frameworks like MediaPipe perform optimally with RGB images, which align with human visual perception for accurate analysis. In the second step, the RGB image is processed through MediaPipe, a robust framework for real-time hand tracking. The model detects hands in the image and isolates key regions for further analysis, focusing processing on relevant areas. Step three extracts the x and y coordinates of 21 key landmarks for each hand, including finger joints and palm positions, as depicted in the final step of Figure 3. These landmarks provide a detailed structural representation of the hand. In the fourth step, a pre-trained Random Forest classifier analyzes the spatial arrangement of the hand landmarks. Known for handling complex patterns, this model predicts the specific gesture being performed by converting numerical data into meaningful gesture labels. Step five determines the predicted gesture based on the classifier's analysis, offering insights into the user's hand movements. Finally, step six visually overlays the predicted gesture onto the original input image. Displaying the gesture name or symbol near the detected hand, as seen in Figure 3, enhances user interaction and provides immediate feedback.

3.3 Word and Sentence Recognition using LSTM and LLM

This module focuses on recognizing dynamic gestures for words and forming meaningful sentences using an LSTM model for word recognition, coupled with an LLM for sentence generation. Keypoint data for gestures was captured using Mediapipe from live video streams, enabling accurate temporal pattern recognition and contextual sentence formation.

3.3.1 Keypoint Extraction

Keypoint extraction, including detection and description, is crucial for various multimedia applications[21]. Landmark extraction is key for leveraging detected points in machine-learning tasks. The Mediapipe model detects key landmarks for the face, pose, and hands, which are stored for further use. These landmarks, such as face_landmarks, pose_landmarks, left_hand_landmarks, and right_hand_landmarks, contain coordinates for specific body points, including facial features, body joints, and finger positions. This sequential data is ideal for training LSTM layers to recognize temporal patterns in gestures. Using keypoints instead of raw video reduces computational complexity by focusing on essential features and eliminating unnecessary background details. This speeds up processing and enhances model performance by providing cleaner input, efficiently capturing the spatial and temporal dynamics needed for accurate gesture recognition.

3.3.2 LSTM Model for Word Recognition

The LSTM model is designed for dynamic word recognition in Indian Sign Language, utilizing its strength in processing sequential data and identifying patterns over time. The input data consists of keypoints from video sequences, representing gestures for each word, captured across 30 frames per gesture using MediaPipe. Each keypoint includes x, y, and z coordinates, which are normalized, and corresponding labels are encoded for classification. The dataset is divided into 95% for training and 5% for testing.

The LSTM architecture comprises three main layers. The first two layers contain 64 and 128 units, respectively, and use ReLU activation to introduce non-linearity by outputting positive values or zero. These layers return sequences, preserving temporal relationships in the data. The third LSTM layer, with 64 units, does not return sequences; instead, it summarizes long-term patterns in the input. Following these, two dense layers with 64 and 32 units refine the extracted features using ReLU activation. A final softmax layer predicts the gesture by selecting the class with the highest probability.

The model is trained using the Adam optimizer, which ensures efficient parameter updates, and categorical cross-entropy, which is suitable for multi-class classification problems. Categorical accuracy measures the percentage of correct predictions. After 150 training epochs, the model achieved high accuracy in recognizing gestures, striking a balance between complexity and performance.

3.3.3 LLM For Sentence Formation

The integration of a Large Language Model (LLM) into our sign language recognition project was inspired by two key studies. The first, "Empowering Deaf-Hearing Communication: Exploring Synergies between Predictive and Generative AI-Based Strategies towards (Portuguese) Sign Language"[19], demonstrated how combining LSTM models with LLMs improved Portuguese Sign Language translation, enabling coherent sentence formation from recognized signs. This motivated us to explore LLMs for bridging gesture-based communication with natural language. The second study, "Using an LLM to Turn Sign Spottings into Spoken Language Sentences"[18], showcased the Spotter+GPT hybrid system, which enhanced translation accuracy by transforming spotted signs into coherent sentences. This reinforced the role of LLMs in generating contextually appropriate sentences that align with our objectives.

We experimented with BERT and GPT-2 on Hugging Face. BERT's contextual understanding was strong but unsuitable for sentence generation, while GPT-2 produced coherent short sentences but lacked contextual depth. GPT-4 via OpenAI API excelled in fluency and accuracy, but its limited free tier was impractical. We chose the Gemini-1.5-Flash API for its balance of performance, fluency, and a more generous free usage tier. For example, it generated the sentence "Namaste! I am Indian" from the inputs "Namaste ", "I " and"Indian " making it ideal for real-time integration in our system.

3.4 Confusion Matrix and Performance Measures

The confusion matrix summarizes the performance of a classification model by providing insights into correctly and incorrectly labeled instances. It includes four key elements: True Positives (TP), correctly predicted positive cases; False Negatives (FN), positive cases incorrectly predicted as negative; False Positives (FP), negative cases incorrectly predicted as positive; and True Negatives (TN), correctly predicted negative cases. This matrix helps evaluate model performance effectively.

Accuracy: Accuracy measures the overall correctness of the model by calculating the proportion of true results (both true positives and true negatives) among the total number of cases. Equation (1) depicts the accuracy measure.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision: Precision evaluates the accuracy of positive predictions, focusing on the proportion of true positives among all predicted positives. **Equation 2** shows the precision formula.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall: Recall or sensitivity indicates the model's ability to identify all relevant instances by calculating the proportion of true positives among all actual positives. **Equation 3** depicts the recall measure.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1-Score: F1-Score provides a balanced measure of precision and recall by computing their harmonic mean, which is useful when both metrics need to be considered simultaneously. Equation (4) shows the harmonic mean of precision and recall.

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \tag{4}$$

4 Results and Discussion

This section presents the results of our experiments on sign language recognition and LSTM-LLM-based sentence generation. We evaluated traditional machine learning models, including KNN, Random Forest, and Decision Tree classifiers, alongside advanced sequence models like LSTM networks and LLM. Metrics such as accuracy, PR curves, and F1 scores assessed performance, with PR curves highlighting precision-recall trade-offs and learning curves identifying overfitting or underfitting. Traditional classifiers performed well in basic gesture recognition, while the LSTM-LLM combination excelled in understanding temporal dependencies and generating coherent sentences, proving ideal for robust language translation systems. Random Forest stood out among traditional models for its balance of accuracy, scalability, and interpretability, making it suitable for recognizing alphabets and numbers in sign language. Advanced models like Gradient Boosting and algorithms such as SVM and Naive Bayes were excluded due to their complexity and limitations in handling high-dimensional gesture data.

4.1 Alphabets and Numbers Recognition

This section presents the results of sign language recognition experiments using KNN, Random Forest, and Decision Tree classifiers, evaluated based on accuracy, Precision-Recall (PR) curves, F1 scores, and learning curves. PR curves measured the balance between precision and recall, while F1 scores provided a balanced evaluation for imbalanced data. The KNN classifier achieved the highest accuracy but struggled with scalability. The Decision Tree, while efficient, exhibited overfitting issues. The Random Forest classifier effectively balanced accuracy and generalization, making it a robust choice for scalable and accurate sign language recognition systems.

4.1.1 F1 Score Evaluation

Precision measures the proportion of true positive results among all positive predictions, while recall evaluates the proportion of true positives among all actual positives. The F1 score is the harmonic mean of precision and recall, providing a balanced measure of both. These metrics help assess a model's performance, particularly in imbalanced datasets.

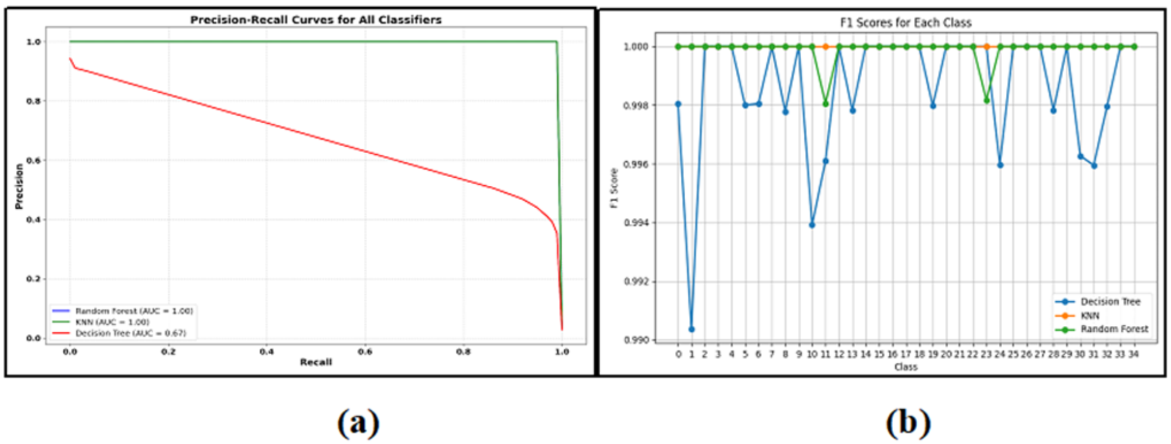


Fig 6. (a) PR curve and (b) F1 score for KNN, decision tree and Random forest classifier

Figure 6**(a)** presents the precision-recall curves for Random Forest, KNN, and Decision Tree models. Both Random Forest and KNN show near-perfect curves with high precision and an AUC of 1.00, indicating excellent performance. In contrast, the Decision Tree model's precision declines as recall increases, resulting in an AUC of 0.67, reflecting its struggle to balance precision and recall. This highlights Random Forest as the superior classifier for handling class imbalances and maintaining robust performance. Figure 6 **(b)** shows the F1 scores for the models. Decision Trees exhibit variability in F1 scores, suggesting overfitting. KNN achieves a perfect F1 score of 1 across all classes, implying potential overfitting. Random Forest performs consistently well, with F1 scores mostly near 1, demonstrating its reliability and robustness compared to the other models.

4.1.2 Learning Curve Evaluation

A learning model in Machine Learning shows how prediction error changes with the size of the training set. **Bias** is the difference between a model's average prediction and the true value; high-bias models, like Linear Regression, oversimplify, resulting in errors on both training and testing sets but are faster and simpler. **Variance** reflects how much predictions change with different training data; models like Decision Trees and SVMs have high variance. The goal is to balance **bias** and **variance**—the bias-variance trade-off. Learning curves help determine the optimal training data size to achieve this balance.

Figure 7 shows the learning curves for Random Forest, Decision Tree, and KNN classifiers, illustrating how accuracy evolves with more training data. The Decision Tree exhibits high training accuracy but significantly lower validation accuracy, indicating overfitting. The KNN model shows moderate performance, with validation accuracy improving as more data is added, suggesting better generalization over time. In contrast, the Random Forest classifier maintains high accuracy in both training and validation, with a small gap between the curves, indicating strong generalization and avoiding both overfitting and underfitting. This makes Random Forest a robust choice for this dataset.

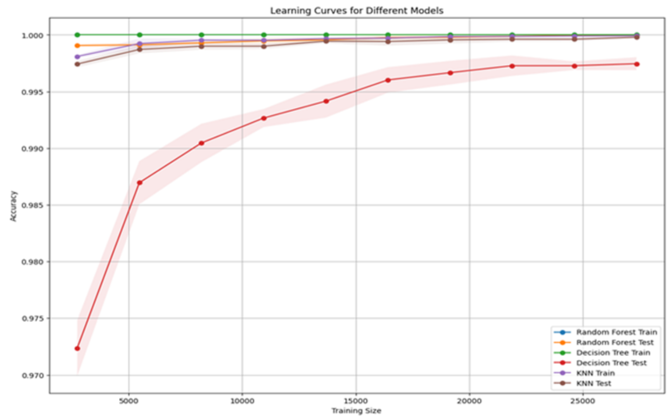


Fig 7. Learning Curve for KNN, Decision Tree & Random Forest Classifier

4.1.3 Performance Evaluation

Signova enhances ISL accessibility with a 99.98% accurate Random Forest classifier for gesture-to-text translation. The performance evaluation revealed the following accuracy rates: The evaluation showed Random Forest achieved 99.98% accuracy with balanced performance, outperforming KNN and Decision Tree. Learning curves show Random Forest's balanced performance, while Decision Trees need adjustments and KNN benefits from more training data.

Table 2. Performance Metrics Comparison for KNN, Decision Tree and Random Forest Classifier (Train-Test Split: 70:30)

Algorithm / Metrics	KNN	Decision Tree	Random Forest
Accuracy	99.99	99.79	99.98
Precision	99.99	99.80	99.98
Recall/ Sensitivity	99.99	99.80	99.98
F1-Score	99.99	99.80	99.98

The Random Forest classifier is the best for sign language recognition due to its stable AUC-PR as shown in Figure 7, ability to handle complex features, and consistent performance. It offers practical advantages beyond its impressive accuracy. Its ability to provide feature importance insights makes it highly interpretable than the other algorithms, helping to identify the most critical keypoints for classification. Additionally, Random Forest is inherently robust to noisy data, as its ensemble averaging reduces the impact of outliers or misclassifications. Compared to KNN, it offers better adaptability, and unlike Decision Trees, it avoids potential bias. This balance of simplicity, power, and reliability makes Random Forest an excellent choice for practical applications.

In machine learning, hyperparameters like the number of trees (n_estimators) impact performance by reducing variance. Increasing trees improves results until performance gains diminish and costs rise. The optimal number is found through evaluation and cross-validation, balancing performance with resource constraints.

The graph in Figure 8 illustrates how the accuracy of a machine learning model varies with the number of trees (n_estimators) in a Random Forest ensemble. As the number of estimators increases from 0 to approximately 25, the accuracy improves significantly, approaching near-perfect levels. Beyond this point, the accuracy stabilizes, with only minimal improvements observed as more estimators are added. This demonstrates diminishing returns in model performance after a certain threshold. In this project, the accuracy reached a peak of 99.98% with an optimal number of 25 estimators. Adding more estimators beyond this value did not yield any significant accuracy gains, confirming the tradeoff between computational efficiency and model performance.

Figure 9 shows the real-time output of the alphabet recognition system. Each sub-image represents the detected hand pose corresponding to a different letter of the alphabet. Key points and connections highlight the distinct configurations for each letter, demonstrating the system's capability to accurately recognize and classify hand gestures.

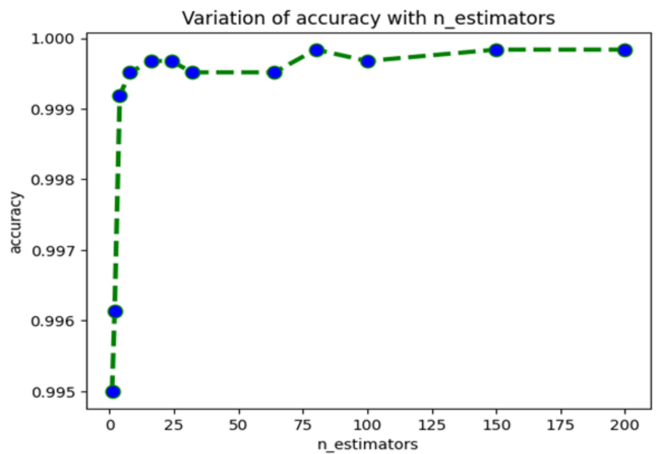


Fig 8. Variation of accuracy with n_estimators (number of trees)

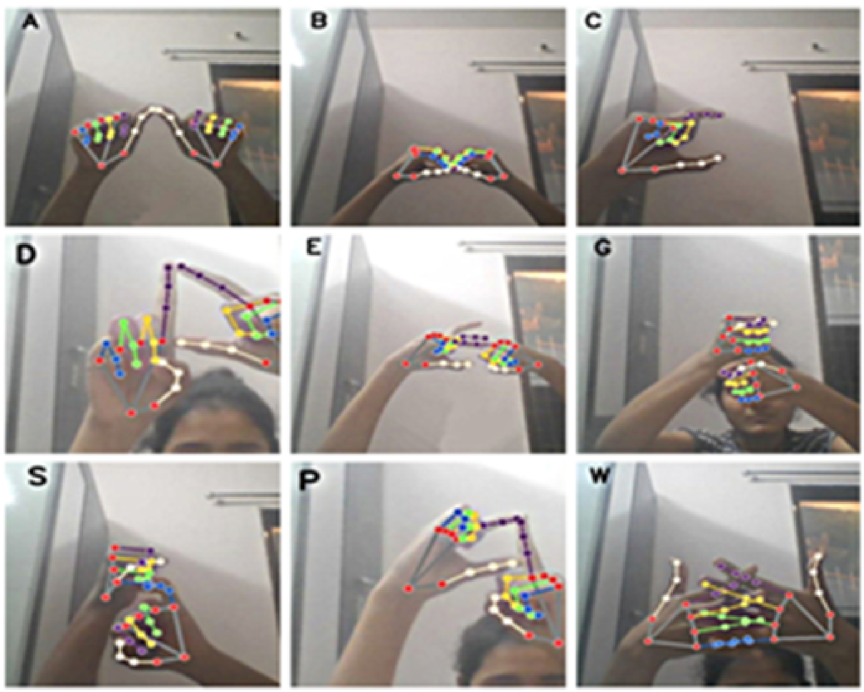


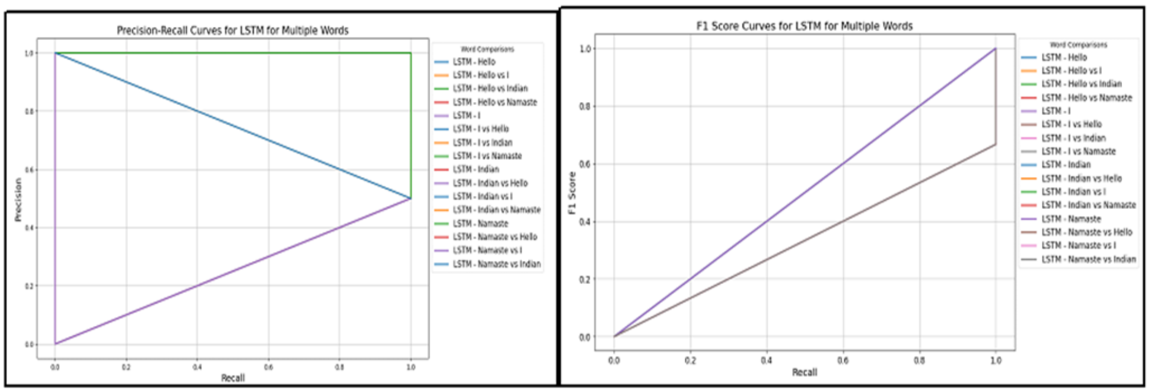
Fig 9. Real-time output of alphabet recognition using OpenCV and mediapipe

4.2 Words and Sentence Recognition

In this section, we present the results of our LSTM-based sentence recognition experiments, highlighting the model's ability to capture temporal dependencies in gesture sequences. This was enhanced by integrating a Large Language Model (LLM) to translate gestures into words and construct sentences. Key metrics, including accuracy, Precision-Recall (PR) curves, and F1 scores, were analyzed. The PR curve balanced precision and recall, while the F1 score evaluated gesture recognition with imbalanced data. Learning curves helped identify overfitting or underfitting. Results showed good accuracy in recognizing gestures and generating sentences, but challenges with scalability and overfitting due to LSTM complexity and limited data. Dropout regularization and early stopping addressed these issues, improving robustness. The discussion highlights the trade-offs of using LSTM and LLM for sentence recognition and provides insights into building a scalable sign language recognition system.

4.2.1 F1 Score Evaluation

Precision quantifies the accuracy of positive predictions made by the model, indicating the fraction of true positive results out of all predicted positives. [Equation (2)] Recall [Equation (3)], on the other hand, measures how well the model captures all relevant instances by calculating the fraction of true positives out of all actual positives. The F1 score [Equation (4)] combines both precision and recall into a single metric, offering a harmonic mean that emphasizes both the ability to make accurate predictions and the ability to identify all relevant instances, making it particularly useful when there is an imbalance in the dataset.



(a) (b)

Fig 10. (a) PR curve and(b) F1-Score curve for the LSTM model shows the precision-recall balance for all the actions (words)

Figure 10**(a)** shows Precision-Recall curves for word comparisons using the LSTM model. The LSTM - Hello (blue line) starts with precision at 1.0, but drops to 0.5 as recall increases, indicating a decline in relevance. The LSTM - Hello vs Indian (green line) maintains a constant precision of 1.0, while the LSTM - I (purple line) shows gradual improvement, with precision rising to 0.6 as recall reaches 1.0. Figure 10 **(b)** shows the relationship between F1 score and recall. The LSTM - Hello (dark purple line) maintains a strong balance, with the F1 score improving as recall increases, while the LSTM - Indian vs Namaste (gray line) struggles with balancing precision and recall. Strong curves indicate robust performance, while flatter ones suggest weaker results.

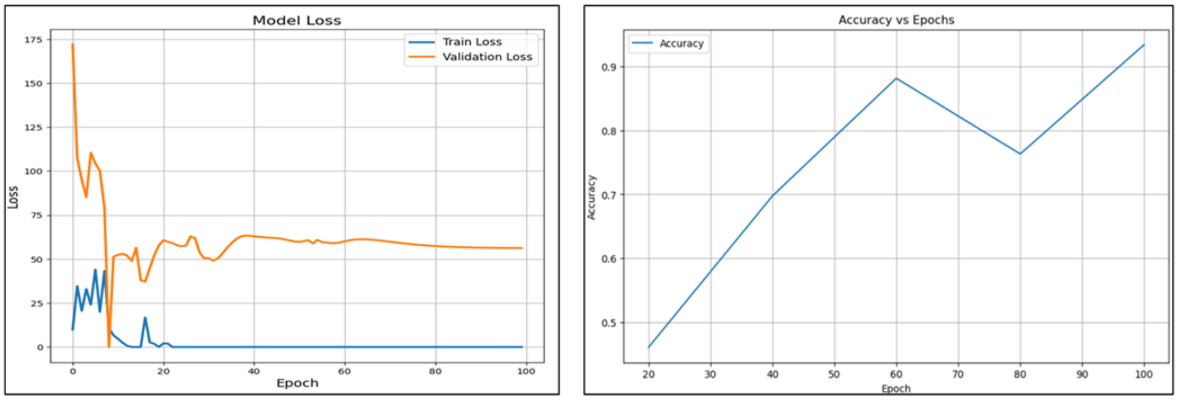
4.2.2 Learning Curve Evaluation

A learning curve plots a model's performance over time, with error or accuracy on the y-axis and training iterations (epochs) on the x-axis. It consists of two curves: the training curve, showing performance on the training data, and the validation curve, indicating generalization to unseen data. A well-behaved curve typically shows a decrease in both training and validation errors as training progresses.

Table 3. The average training loss, validation loss, and accuracy after each 20 epochs

Epoch Range	Average Train Loss	Average Validation Loss	Average Accuracy
1-20	13.78	69.86	24.34
21-40	0.19	57.48	69.73
41-60	7.82	60.69	88.15
61-80	6.37	59.66	76.31
81-100	5.99	56.59	93.42

Figure 11 shows the learning curves of an LSTM model trained over 100 epochs, depicting the loss values for both training and validation datasets. The training loss declines sharply after the fifth epoch, indicating effective learning, while the validation loss fluctuates, suggesting potential overfitting. Overfitting occurs when the model becomes too specialized in the training data, leading to poor performance on unseen data. To improve generalization, techniques like dropout regularization, increasing the training dataset size, adjusting the learning rate, and early stopping can be applied. Early stopping halts training when validation loss increases, even if training loss continues to decrease, enhancing the model's stability and performance on unseen data.



(c) (d)

Fig 11. The learning curve for the LSTM model shows a decrease in the categorical cross-entropy loss up to 100 epochs

4.2.3 Performance Evaluation

This section evaluates the performance of the LSTM-based gesture recognition model using accuracy, precision-recall (PR) curves, F1 scores, and learning curves. The accuracy varied across word actions, with models like "LSTM - Hello" achieving perfect precision but lower recall, leading to an F1 score of 0.50. Models like "LSTM - I" and "LSTM - Indian" had a more balanced performance, with an F1 score of 0.56. The PR curves (Figure 10 **(a)**) and F1 score curves (Figure 10 **(b)**) illustrate these trade-offs. Learning curves (Figure 11 **(c)**) reveal initial efficient learning, but overfitting occurs as validation loss fluctuates and accuracy diverges. To address this, techniques like dropout regularization and early stopping are recommended. Despite its effectiveness, the LSTM model faces scalability challenges due to computational complexity and overfitting, particularly with larger datasets or more complex sequences. Future work should focus on optimizing the model's architecture, enhancing generalization, and exploring transfer learning. In conclusion, the LSTM model achieves 93% accuracy but needs improvement in balancing precision, recall, and generalization for more complex sign language gestures.

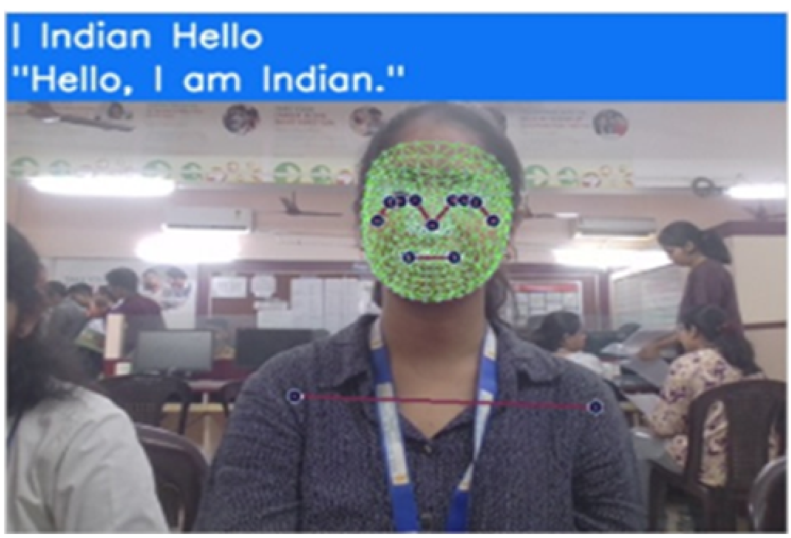


Fig 12. Real-time output of word recognition and converting it into sentences

Figure 12 illustrates the real-time output of the gesture-to-sentence recognition system, translating sign language gestures into text and meaningful sentences. The interface displays recognized words ("I", "Indian", and "Hello") identified by the LSTM model, which captures the temporal dependencies of gesture data. These words are then passed to the Gemini-1.5-Flash Large Language Model (LLM), which refines them into a coherent sentence, such as "Hello, I am Indian." The system operates in two stages: the LSTM model focuses on accurate word recognition based on gesture motion, while the LLM forms a grammatically correct and contextually appropriate sentence. This dual-stage mechanism allows the system to translate real-time gestures into fluent sentences, highlighting its potential for real-world communication between sign and non-sign language users. The approach represents a significant advancement in bridging communication gaps in sign language recognition.

4.3 Comparison with previous works and novelty

Signova distinguishes itself from existing initiatives aimed at improving the accessibility of Indian Sign Language (ISL) by integrating advanced machine learning techniques with user-centric educational tools, with a primary focus on its innovative sentence generation capabilities. Unlike DEF-ISL[22], developed by SriJay Software Solutions Pvt Ltd, which offers a vast repository of over 50,000 signs without real-time gesture recognition, and, which relies on certified interpreters for on-demand ISL interpretation, Signova employs a Random Forest classifier to achieve an impressive 99.98% accuracy in translating hand gestures into text. This capability facilitates direct communication without interpreter dependency. Similarly, SignAble[23] offers an on-demand ISL interpretation service by connecting users with certified interpreters through video calls, addressing the shortage of interpreters in India but relying heavily on human resources. This capability is enhanced by the integration of Long Short-Term Memory (LSTM) networks with the Gemini-1.5 Flash API, allowing for the seamless conversion of ISL gestures into coherent sentences and improving the overall communication experience for users. Sentence recognition, a key challenge in sign language, has been explored in languages like Portuguese[19] and American Sign Language[18]. However, there has been limited focus on it in Indian Sign Language. Signova fills this gap by pioneering a system that not only recognizes individual signs but also understands the context and structure necessary to generate complete sentences. This integration allows for real-time processing of gestures, facilitating not just word recognition but also the construction of meaningful sentences from sign language inputs, which is crucial for effective communication and bridging the gap between the deaf and hearing communities. Signova offers an all-in-one platform with a translation module, a learning tool with flashcards, and a quiz module for interactive learning. This unique combination of real-time translation, educational tools, and assessment features makes Signova a comprehensive solution for ISL communication, fostering inclusivity and engagement while empowering users to learn ISL effectively.

5 Conclusion

Signova marks a significant advancement in bridging communication gaps between the hearing and hearing-impaired communities. What makes it novel is its vision as an educational platform that provides learning and practice along with translation since an all-inclusive educational platform for ISL does not exist yet. Utilizing machine learning, Signova stands out by offering a free robust platform for translating Indian Sign Language (ISL) into text and speech, enhancing communication accessibility. The platform's interactive learning modules, quizzes, and educational content further enrich its utility, serving as a real-time communication tool and an educational resource for ISL. The strength lies in the successful use of the Random Forest Classifier ensures high accuracy and reliability for recognizing letters and numbers. While the LSTM model plays a key role in detecting and recognizing words. Ultimately, Signova aims to create a more inclusive society by minimizing communication barriers and enabling fuller participation for individuals with hearing impairments.

Future developments include optimization by employing the YOLOv8 model recognition of signs for letters and numbers in any environment with reduced latency along with making the LSTM model more robust and accurate with a faster response by manipulating its layers; offload real-time video processing including extracting landmarks to user devices to reduce server load and improve scalability. User feedback was based on the overall latency and increase in the accuracy of the LSTM model, which shall in turn be achieved through the optimizations discussed previously. The use of YOLOv8 was suggested through user feedback. The future scope holds a module that will convert text or speech to ISL.

References

1) Deafness and hearing loss. 2024. Available from: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.
2) Varshney S. Deafness in India. *Indian Journal of Otology*. 2016;22(2):73–76. Available from: https://doi.org/10.4103/0971-7749.182281.
3) Wasim. Status of Deaf and Hard of Hearing (DHH) individuals in India. Free UPSC IAS Preparation for Aspirants. 2024. Available from: https://forumias.com/blog/status-of-deaf-and-hard-of-hearing-dhh-individuals-in-india/.
4) Abhishek M, Nagarkar AN, Nagarkar NM. Challenges in Education and Employment for Hearing Impaired in India" (PDF). *Journal of Disability Management and Special Education*. 2018;1(1):34–39. Available from: https://www.jodys.info/journal/january_2018/04_Jodys_January_2018.pdf.

5) Daily Current Affairs on NCERT textbooks now available in Indian sign language for UPSC Civil Services Examination (General Studies) Preparation. 2024. Available from: https://abhipedia.abhimanu.com/Article/IAS/MjI5ODgz/NCERT-textbooks-now-available-in-Indian-sign-language-Social-Issues-IAS.
6) Tomar SM, Patel NM, Thakore DG. A Survey on Sign Language Recognition Systems. *International Journal of Creative Research Thoughts*. 2021;9(3):4431–4434. Available from: https://www.ijcrt.org/papers/IJCRT2103503.pdf.
7) Brahmana CRPS, Sofyan R, Putri DM. Problems in the application of Google Translate as a learning media in translation. *Language Literacy: Journal of Linguistics, Literature, and Language Teaching*. 2020;4(2):384–389. Available from: https://jurnal.uisu.ac.id/index.php/languageliteracy/article/view/2893/pdf.
8) Katoch S, Singh V, Tiwary US. Indian Sign Language recognition system using SURF with SVM and CNN. *Array*. 2022;14:1–9. Available from: https://doi.org/10.1016/j.array.2022.100141.
9) Sharma S, Singh S. Recognition of Indian Sign Language (ISL) Using Deep Learning Model. *Wireless personal communications*. 2022;123:671–692. Available from: https://doi.org/10.1007/s11277-021-09152-1.
10) Subramanian B, Olimov B, Naik SM, Kim S, Park KH, Kim J. An integrated mediapipe-optimized GRU model for Indian sign language recognition. *Scientific Reports*. 2022;12(1):1–16. Available from: https://doi.org/10.1038/s41598-022-15998-7.
11) Srivastava S, Singh S, Pooja, Prakash S. Continuous Sign Language Recognition System Using Deep Learning with MediaPipe Holistic. *Wireless Personal Communications*. 2024;137:1455–1468. Available from: https://doi.org/10.1007/s11277-024-11356-0.
12) Nandi U, Ghorai A, Singh MM, Changdar C, Bhakta S, Pal RK. Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling. *Multimedia Tools and Applications*. 2023;82(7):9627–9648. Available from: https://doi.org/10.1007/s11042-021-11595-4.
13) Mishra R, Rathi S. Efficient and Scalable Job Recommender System Using Collaborative Filtering. In: ICDSMLA 2019;vol. 601 of Lecture Notes in Electrical Engineering. Singapore. Springer. 2019;p. 842–856. Available from: https://doi.org/10.1007/978-981-15-1420-3_91.
14) Mishra R, Rathi S. Inductive Learning in Job Recommendation. *International Journal of Intelligent Systems and Application in Engineering*. 2022;10(2):242–251. Available from: https://ijisae.org/index.php/IJISAE/article/view/1829.
15) Mishra R, Rathi S. Enhanced DSSM (Deep Semantic Structure Modelling) Technique for Job Recommendation. *Journal of King Saud University - Computer and Information Sciences*. 2022;34(9):7790–7802. Available from: https://doi.org/10.1016/j.jksuci.2021.07.018.
16) Pandey M, Mishra R. Identity Resolution In Social Network Using Recommender System. In: e-Conference on Data Science and Intelligent Computing. 2020;p. 97–102. Available from: https://www.tcetmumbai.in/Multicon/2020/ECDSIC%202020-PROCEEDING%20with%20ISBN.pdf.
17) Ravita R, Rathi S. Inductive Learning Approach in Job Recommendation". *International Journal of Intelligent Systems and Applications in Engineering*. 2022;10(2):242–251. Available from: https://ijisae.org/index.php/IJISAE/article/view/1829.
18) Sincan OM, Camgoz NC, Bowden R. Using an LLM to Turn Sign Spottings into Spoken Language Sentences. *arXiv*. 2024;p. 1–6. Available from: https://arxiv.org/abs/2403.10434.
19) Adão T, Oliveira J, Shahrabadi S, Jesus H, Fernandes M, Costa A, et al. Empowering Deaf-Hearing Communication: Exploring Synergies between Predictive and Generative AI-Based Strategies towards (Portuguese) Sign Language Interpretation. *Journal of Imaging*. 2023;9(11):1–30. Available from: https://doi.org/10.3390/jimaging9110235.
20) Arikeri P. Indian Sign Language (ISL). *Kaggle*. 2021. Available from: https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl/data.
21) DKNAS: A Practical Deep Keypoint Extraction Framework Based on Neural Architecture Search. In: 2022 International Conference on Robotics and Automation (ICRA). IEEE. 2022. Available from: https://doi.org/10.1109/ICRA46639.2022.9812101.
22) DEF-ISL App - Deaf Enabled Foundation. . Available from: https://def.org.in/def-isl-app/.
23) SignAble App. . Available from: https://signable.live/.