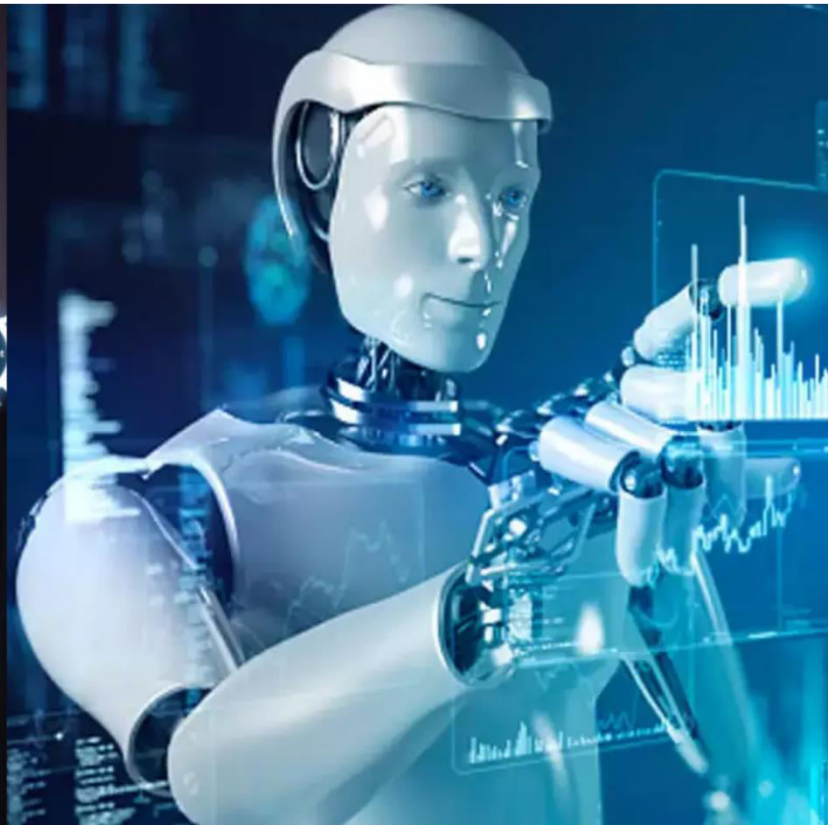
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 4, April 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404614|

AI based Sign Language Translator

Vaishali Bhusari[1], Aanchal Agnihotri[2], Chinmay Attarde[3], Yash Bendale[4]

Assistant Professor, Department of Computer Engineering, K.C. College of Engineering and Management Studies, India[1]

Researcher (Student), Department of Computer Engineering, K.C. College of Engineering and Management Studies, India[2,3,4]

ABSTRACT: Millions of people who are deaf or hard of hearing face communication barriers in their daily lives. Sign language serves as a primary mode of communication for them, but the lack of widespread understanding among non-signers creates a gap. To address this, we present a real-time Sign Language Translator using Node.js and Tensor Flow that bridges communication between sign language users and the hearing population. This web-based system captures hand gestures via a webcam and processes them using deep learning models built with Tensor Flow. These models, trained on sign language datasets, recognize gestures and translate them into readable text or speech output. The backend, developed with Node.js, ensures efficient real-time data processing and smooth interaction between components. The translator supports dynamic and static gestures from sign languages like ASL and ISL, with potential for further expansion. It is accessible via web and mobile platforms, making it suitable for everyday use in public spaces, education, healthcare, and more. By enabling two-way communication without the need for a human interpreter, this system promotes accessibility, inclusion, and social integration for the deaf and hard of hearing community. It demonstrates the power of AI and web technologies in driving social impact.

KEYWORDS: Sign language translation**,** gesture recognition**,** machine learning**,** deaf and hard of hearing**,** Node.js**,** Tensor Flow

I. INTRODUCTION

Communication is an essential aspect of human interaction, yet for millions of people who are deaf or hard of hearing, effective communication with the hearing population remains a significant challenge. While sign language is a powerful and expressive medium for the deaf community, the majority of the population does not understand it, often leading to isolation, misunderstandings, and limited access to essential services. With the rapid advancements in technology, especially in artificial intelligence and machine learning, there is a growing opportunity to bridge this communication gap through automated sign language translation. This project presents a real-time Sign Language Translator that utilizes Node.js and TensorFlow to recognize and translate hand gestures into text or speech. The system is designed to interpret both static and dynamic signs using deep learning models trained on large datasets of sign language gestures. It captures real-time video input through a webcam, processes the frames using TensorFlow-powered models, and converts the recognized signs into readable text or synthesized speech. Node.js powers the backend to ensure a fast, scalable, and responsive application, capable of handling live data processing and communication between the frontend and machine learning components. The application aims to be accessible through both web and mobile platforms, making it practical for real-world environments such as hospitals, schools, workplaces, and public service areas. By enabling two-way communication between deaf and hearing individuals without the need for a human interpreter, this project seeks to enhance social inclusion, accessibility, and independence. It also serves as a valuable educational tool for those learning sign language, contributing to a more empathetic and inclusive society. The integration of modern technologies in this translator demonstrates how innovation can empower underrepresented communities and promote equal opportunities in communication.

II. LITERATURE SURVEY

1. Early Vision-Based Approaches rely on static gesture recognition techniques, such as template matching and color segmentation. These methods are relatively simple and allow for basic exploration of vision-based sign recognition systems. Their major advantage lies in their ease of implementation and low computational

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404614|

demands. However, they are highly sensitive to environmental factors like lighting conditions, background clutter, and interference, which limit their accuracy and practicality in real-world applications.

2. Hidden Markov Models (HMM) are used to model the sequential nature of dynamic gestures in sign language. By focusing on the temporal sequence of hand movements, HMMs improve the recognition of dynamic gestures. Despite their effectiveness in handling sequential data, they suffer from high computational complexity, making real-time implementation difficult. These models often require careful tuning and substantial processing resources to function effectively.

3. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) architectures are designed to capture the temporal dependencies in sign language by processing sequential data. They excel at modeling time-dependent patterns such as hand movement sequences. This allows for more accurate gesture recognition. However, these models face challenges with very long sequences and tend to be computationally intensive, making them harder to optimize for real-time performance.

4. Real-Time Gesture Recognition incorporates convolutional neural networks (CNNs), RNNs, and LSTMs to process live video input. This approach enables low-latency gesture recognition, which is essential for live communication scenarios. The strength of this method lies in its speed and suitability for interactive use. Nevertheless, it struggles with variability in gestures across individuals and environments, posing difficulties in maintaining consistent accuracy.

5. Multimodal Approaches integrate multiple cues—such as hand gestures, facial expressions, and body posture—to achieve more accurate and context-aware sign language translation. These approaches significantly improve recognition by considering a richer set of input data. However, the integration of different data modalities increases system complexity and demands higher computational resources, which can make real-time processing more challenging.

6. Public Datasets like ASL and RWTH-PHOENIX play a crucial role in training machine learning models for sign language recognition. These datasets offer labeled data and a variety of gestures that are essential for model development and evaluation. While they provide a good foundation, the primary limitation is the lack of diversity in sign languages and environmental conditions, which restricts the generalizability of trained models across different users and settings.

III. METHODOLOGY

The Sign Language Translator follows a systematic methodology integrating computer vision, deep learning, and natural language processing to ensure efficient recognition and translation of sign language gestures. The overall process is broken down into key phases, as detailed below.

3.1 Data Collection & Preprocessing

The initial stage involves selecting suitable datasets for training. Publicly available datasets such as RWTH-PHOENIX-Weather (for German Sign Language), the American Sign Language (ASL) Dataset, and the Indian Sign Language (ISL) Dataset serve as excellent resources. Additionally, custom datasets can be created using webcam or smartphone inputs to collect diverse gesture samples tailored to specific applications.

Preprocessing includes multiple steps to prepare the data for training. Background removal is performed using OpenCV to isolate the hand from the rest of the frame. MediaPipe is employed for real-time hand landmark detection, which significantly improves the precision of gesture tracking. Feature extraction is then carried out using Convolutional Neural Networks (CNNs), enabling the system to capture critical characteristics such as hand shape, position, and movement.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404614|

3.2 Model Development & Training

In this phase, CNNs are used to extract spatial features from sign language images. These features help the system identify hand orientations, shapes, and positions with high accuracy. For recognizing dynamic and continuous gestures, Long Short-Term Memory (LSTM) networks or Recurrent Neural Networks (RNNs) are utilized. These models are particularly effective for sequential data and ensure contextual understanding of the gestures.

The training process is conducted using deep learning frameworks like TensorFlow or PyTorch. To enhance the robustness and generalization of the model, data augmentation techniques are applied. Additionally, hyperparameter tuning is performed to optimize the model's performance and accuracy.

3.3 System Integration & Real-Time Processing

The integrated system begins by capturing real-time video frames from a webcam or smartphone camera. These frames undergo preprocessing before being fed into the gesture classification pipeline. The trained deep learning model processes each frame to detect and classify hand gestures accurately.

Once a gesture is recognized, the system generates corresponding text output that is displayed to the user. To further improve communication, especially with individuals who may not read the text, a Text-to-Speech (TTS) module converts the text into spoken words, enabling real-time audio feedback.

3.4 User Interface Development

The user interface is designed to be simple and intuitive, allowing users to view real-time translations of their gestures. It also includes options to select different sign languages such as ASL, BSL, and ISL, enhancing its usability across various linguistic regions.

For broader access, the system is made compatible with both web and mobile platforms. Web-based applications are developed using frameworks like Flask or Django combined with React, while mobile versions are implemented using Flutter or React Native to ensure cross-platform functionality.

3.5 Testing & Evaluation

To evaluate the model’s accuracy, standard performance metrics such as confusion matrix, precision, recall, and F1-score are used. These metrics provide insight into how well the model performs and allow for comparisons with existing sign language recognition systems.

User testing is also a critical component. Feedback is gathered from deaf and hard-of-hearing users to understand the system’s real-world effectiveness. Insights from this testing phase are used to refine and improve both the model and the user experience.

3.6 Deployment & Future Enhancements

The final system is deployed as a fully functional web and mobile application. It can also be integrated with assistive devices and accessibility tools to support a wider range of users.

Looking ahead, future enhancements include expanding support for multiple sign languages, improving gesture recognition under challenging conditions such as low lighting, and incorporating AI-powered lip-reading to complement hand gesture recognition for more accurate translations.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

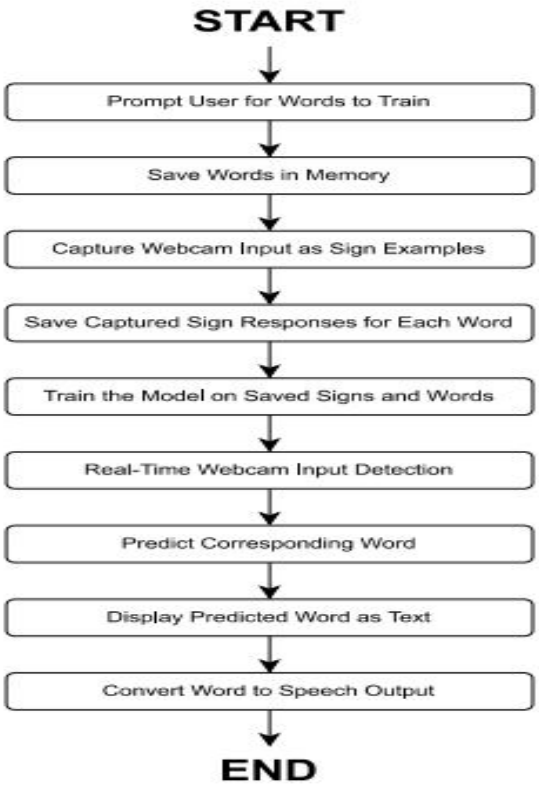
|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404614|

IV. PROJECT DESIGN

Flowchart:



Explanation of Flowchart:

1. Start

- This is the entry point of the application.
- When the system is launched, it prepares to guide the user through setting up the sign recognition process.

2. Prompt User for Words to Train

- The system asks the user to input the specific words or commands they want it to recognize using sign language.
- Example: The user might input words like Hello, Thank You, Yes, No, Help, etc.
- These words will define the vocabulary the system needs to learn.

3. Save Words in Memory

- The system saves this list of words locally (in memory, browser storage, or database).
- These saved words become labels or classes that the machine learning model will be trained on.
- Each word is associated with a unique sign performed by the user.

4. Capture Webcam Input as Sign Examples

- For each word, the system activates the webcam and asks the user to perform the corresponding sign multiple times.
- During this, it records images or video frames or extracts hand landmark/keypoint data using libraries like TensorFlow.js HandPose or MediaPipe Hands.
- The goal here is to collect training examples for each word (i.e., many samples of each hand sign).

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404614|

5. Save Captured Sign Responses for Each Word

- The data captured from the webcam (images or landmarks) is stored and labeled with the correct word.
- This dataset becomes the training dataset.
- For example, 30 frames of the hand gesture for "Hello" will be saved and labeled as Hello.

6. Train the Model on Saved Signs and Words

- A machine learning model (usually a classification model) is trained using the collected sign examples.
- It learns to distinguish between different hand gestures and associate them with the correct words.
- This training can happen on the browser itself (using TensorFlow.js) or on the server.

7. Real-Time Webcam Input Detection

- Once training is complete, the system enters real-time mode.
- The webcam continues to monitor hand movements.
- Each new frame or keypoint data is sent to the model for analysis.

8. Predict Corresponding Word

- The model analyzes the real-time input and predicts which trained word the hand sign corresponds to.
- For instance, if the user signs "Thank You", the model detects it and outputs "Thank You" with high confidence.

9. Display Predicted Word as Text

- The predicted word is shown on the screen so the user and others can read the recognized sign.
- This allows users with hearing/speech difficulties to communicate visually.

10. Convert Word to Speech Output

- A Text-to-Speech (TTS) system converts the predicted word into audio.
- This allows the sign language to be heard as spoken language, helping in conversations with people who don't know sign language.

11. End

- The process can loop continuously for ongoing detection or end based on user interaction.

V. ACKNOWLEDGEMENT

We are deeply indebted to our respected guide, **Prof. Vaishali Bhusari**, for her invaluable guidance, constant encouragement, and insightful suggestions throughout the course of our research. Her expert knowledge, patience, and unwavering support played a crucial role in shaping our ideas and bringing this project to fruition. We sincerely appreciate the time and effort she invested in reviewing our work and providing constructive feedback at every stage, which greatly enhanced the quality and direction of our study.

VI. CONCLUSION

The sign language translator is a powerful tool designed to bridge the communication gap between hearing and non-hearing individuals. By allowing users to train the system with specific words and their corresponding hand signs, it creates a personalized and adaptive model that accurately recognizes gestures through real-time webcam input. The system processes these gestures, predicts the associated word using machine learning, and provides both text and speech output. This dual-mode response ensures that the signed message is understood by a wider audience, including those unfamiliar with sign language. Its ability to learn and adapt to new signs makes it a flexible and scalable solution for various communication needs. The translator not only fosters inclusivity but also empowers individuals with speech or hearing impairments by giving them a voice in both virtual and real-world conversations. Overall, it is a significant step toward accessible, human-centered technology that enhances interaction and understanding for all.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 4, April 2025

|DOI: 10.15680/IJIRSET.2025.1404614|

REFERENCES

1. Koller, O., Zargaran, S., Ney, H., & Bowden, R. (2019). Deep Sign: Enabling Robust Sign Language Recognition with Hybrid CNN-HMM and Transformer Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).
2. Zhang, X., Tao, Y., & Sun, H. (2021). *Real-Time Hand Gesture Recognition Using CNN-LSTM Networks for Sign Language Translation*. Journal of Computer Vision and Applications.
3. Lugaresi, C., et al. (2019). *MediaPipe Hands: On-Device Real-Time Hand Tracking*. Google AI Blog. Retrieved from https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html
4. Kumar, P., et al. (2020). *A Deep Learning-Based Approach for Indian Sign Language Recognition*. International Journal of Computer Science and Artificial Intelligence.
5. Kaggle. (n.d.). *ASL Alphabet Dataset*. Retrieved from https://www.kaggle.com/grassknoted/asl-alphabet
6. Forster, J., Schmidt, C., & Ney, H. (2014). *Continuous Sign Language Recognition: RWTH-PHOENIX-Weather Corpus*. Proceedings of the International Conference on Language Resources and Evaluation (LREC).
7. Bradski, G. (2000). *The OpenCV Library*. Dr. Dobb's Journal of Software Tools.
8. Abadi, M., et al. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Retrieved from https://www.tensorflow.org
   Paszke, A., et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Advances in Neural Information Processing Systems (NeurIPS).
9. Warden, P. (2018). *Speech Commands Dataset: Recognizing Spoken Words with Deep Learning*. IEEE Transactions on Audio, Speech, and Language Processing.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details