Indian Sign Language Interpreter Using AI

Parth Patankar[*1], **Ashraf Shikalgar**[*2], **Shubham Palkar**[*3], **Shaikh Noor Alam**[*4], **Prof. Sanjay Jadhav**[*5]

[*1,2,3,4] *Student at Mahatma Gandhi Mission College of Engineering and Technology, Mumbai, Maharashtra, India.*

[*5]*Associate professor at Mahatma Gandhi Mission College of Engineering and Technology, Mumbai, Maharashtra, India.*

---***---

Abstract - We herein introduce a real-time Indian Sign Language (ISL) interpreter system based on deep learning that translates hand gestures to human-legible and hearable representations. Using MediaPipe for landmark detection and a Convolutional Neural Network (CNN) developed using TensorFlow/Keras, the system detects ISL handshapes that correspond to letters (A–Z) and numbers (0–9). The predictions are rendered in a real time using a Flask-based website, and optionally spoken out using Google Text-to-Speech (gTTS). The tool can be a valuable assistant for those who use ISL by making it easy to communicate across digital channels. The system is intended to be lightweight, deployable, and easy to use, thus fitting for educational, healthcare, and accessibility assistive uses.

***Key Words*:** Indian Sign Language (ISL), Deep Learning, TensorFlow, MediaPipe, Flask, Computer Vision, Real-Time Translation, Accessibility

1. INTRODUCTION

Communication is a fundamental part of human interaction, yet millions of people around the world who are hearing or speech-impaired struggle daily to express themselves due to a lack of accessibility and general awareness of sign language. Indian Sign Language (ISL), which uses hand gestures, facial expressions, and movements, is the main mode of communication for many in India's deaf community. However, the absence of standardized tools for automatic sign recognition and translation has severely limited inclusive communication, learning, and employment opportunities.

In India, real-time ISL interpreters are scarce. Human interpreters are expensive and rarely available in places like public venues, classrooms, hospitals, or government offices.

To bridge this divide, this project introduces an AI-powered ISL Interpreter that converts hand gestures into readable text and spoken words in real time. It integrates MediaPipe for detecting hand landmarks, TensorFlow/Keras for classifying gestures, Flask for web deployment, and gTTS for voice output. Requiring only a basic webcam and a browser, the system is designed to be lightweight, easy to use, and platform-independent.With applications in education, public services, and digital communication, the ISL Interpreter moves us closer to a more inclusive digital environment — one where communication isn't constrained by physical or auditory limitations.

2. RELATED WORK

A variety of research studies and initiatives have addressed the topic of sign language recognition, predominantly for American Sign Language (ASL) and overall gesture detection. Traditional machine learning approaches like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees were used in the initial systems, based heavily on hand-engineered features such as skin color segmentation, contour analysis, and motion tracking. While these techniques provided acceptable accuracy on limited datasets, they faltered against real-world variation and dynamic movement.

With the advent of deep learning, Convolutional Neural Networks (CNNs) transformed gesture recognition by learning spatial features from images automatically. Researchers, for example, accomplished high accuracy using large datasets and trained models in 'Real-Time American Sign Language Alphabet Recognition using Deep CNNs.' The models were usually computationally demanding and only worked for static gestures and did not account for the temporal dimension of communication.

Recent innovations have used Google's MediaPipe for hand landmark detection in real time, which supports lightweight and effective gesture recognition pipelines. There have been studies which used MediaPipe keypoints together with machine learning classifiers like Random Forests or MLPs and achieved encouraging performance with reduced computing overhead. Other more significant contributions have used LSTM networks and

Temporal Convolutional Networks (TCNs) to identify sequential-based dynamic gestures by modeling sequential inputs.

While all these developments, so far, are skewed heavily towards supporting only ASL and do not support Indian Sign Language (ISL), very few of these efforts have presented an end-to-end integrated system that combines gesture recognition, sentence formation, and real-time speech synthesis. Our system stands out by being particularly focused on ISL alphabet and numeral recognition, using real-time keypoint extraction, deep learning-classification, and effortless web deployment along with text-to-speech functionality, targeting usage across a variety of settings.

3. PROPOSED SYSTEM

The proposed model is a real time Indian Sign Language (ISL) detection and translation model and aims to assist communication between a hearing impaired individual with one who cannot sign. This mechanism harnesses computer vision, deep learning, and vocal synthesis to identify ISL hand gestures and translate into text and voice. MediaPipe The system is based on MediaPipe, a ML pipeline from Google that seeks to minimize the complexity of building and optimizing the flow of data and processing elements that goes into a ML model. These hand landmarks – typically 21 per hand – act as the custom-trained deep learning model's input features that we created using TensorFlow/Keras. The model is trained on Indian Sign Language Gestures data with labels, and can effectively recognize a large range of signs, from alphabets to words to daily usage phrases.

When a gesture is classified the predicted label is sent to the Flask backend which takes care of server-side processing and passing responses between the machine learning model and the frontend user interface. The translated label (in English and Hindi text) is rendered to the screen and it's corresponding audio message is generated using Google Text-to-Speech (gTTS) as well. This allows non-deaf users to understand in real-time the meaning of the signed gesture. Due to a lightweight design of the entire system, it appears deployable in a non-intrusive fashion.

Several advantages are provided by this system. It is available, low-cost, and easy to use, and requires no more equipment than a simple webcam. It goes a long way to breaking down communication walls for the deaf community, and is a step towards inclusivity. Furthermore, being modular and able to scale, the model can be extended to larger sign-set in future, and can potentially support other languages or even a sentence level translation. But some factors like gesture similarity, cover noise, complex hand gesture orientation and lighting condition may degrade the recognition performance. To address these, future work will explore more diverse datasets, reducting noises, fine-tuning classifier, and potentially introduce NLP for generating grammatical correct full sentence translations.

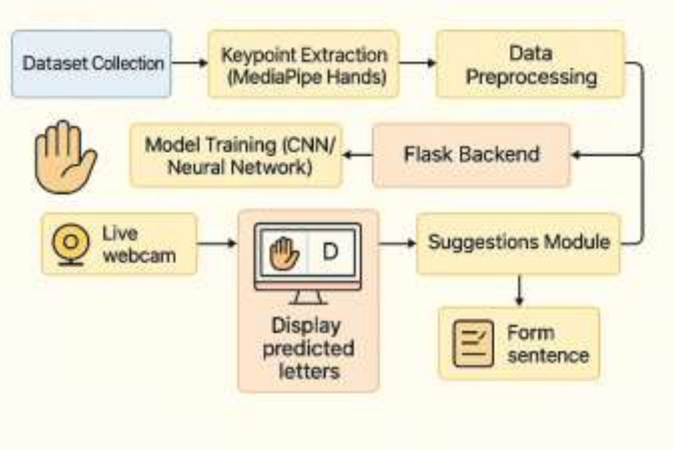

Dataset Collection
Keypoint Extraction (MediaPipe Hands)
Data Preprocessing
Model Training (CNN/ Neural Network)
Flask Backend
Live webcam
D
Display predicted letters
Suggestions Module
Form sentence

Fig - 1: System Architecture of Indian Sign Language Interpreter

4. METHODOLOGY

A structured pipeline that includes data collection, preprocessing, model training, real-time detection, translation, and audio output is the methodology used in this project. The first step in the system is data acquisition, which involves creating or gathering a unique dataset of Indian Sign Language (ISL) gestures. Images or video frames of various hand gestures that represent words, phrases, or alphabets make up this dataset. To aid in supervised learning, each gesture has the proper label. The dataset is preprocessed by resizing the frames, normalizing pixel values, and extracting hand landmarks using MediaPipe Hands, a potent real-time hand tracking tool that offers 21 important landmarks per hand, for improved accuracy and performance.

These landmark coordinates are recorded and saved as numerical vectors in the feature extraction step that follows.

The machine learning model uses these vectors as its input features. TensorFlow/Keras is used to create a deep learning model, usually a fully connected neural network or a convolutional neural network (CNN) trained on the landmark vectors. To guarantee robustness and minimize overfitting, the model is trained across several epochs using strategies like data augmentation, dropout regularization, and validation splitting.

Following training, Flask, the backend framework that manages data flow between the ML model and user input (webcam feed), is used to integrate the model into a real-time system. In order to detect hand landmarks in real-time, the system records live video input from a webcam, processes it using MediaPipe, and then feeds the data into a trained model for gesture classification. Google Text-to-Speech (gTTS) is then used to translate the predicted gesture into meaningful text and then into audio. A straightforward and responsive web interface is used to display the final output, which includes both text and speech, to the user.

Real-time performance and latency are major areas of focus throughout the process, making sure that the system reacts in a matter of milliseconds for a seamless user experience. To assess the system's capacity for generalization, it is also tested on a range of users in a variety of lighting and background scenarios. To evaluate the model's performance, its accuracy, precision, and recall metrics are computed. The model hyperparameters are adjusted, the dataset is enlarged, and the frontend interface is improved for greater usability based on feedback and results.

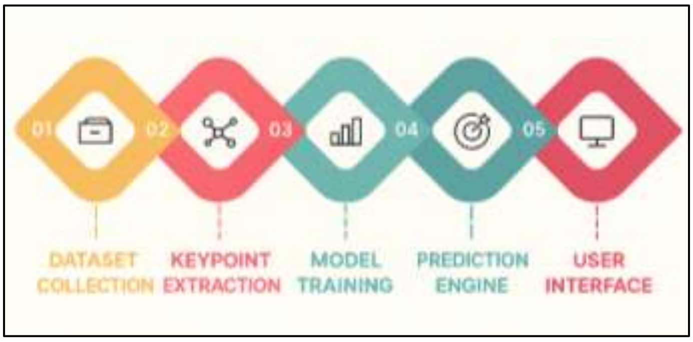


Fig - 2: Project Module Flow

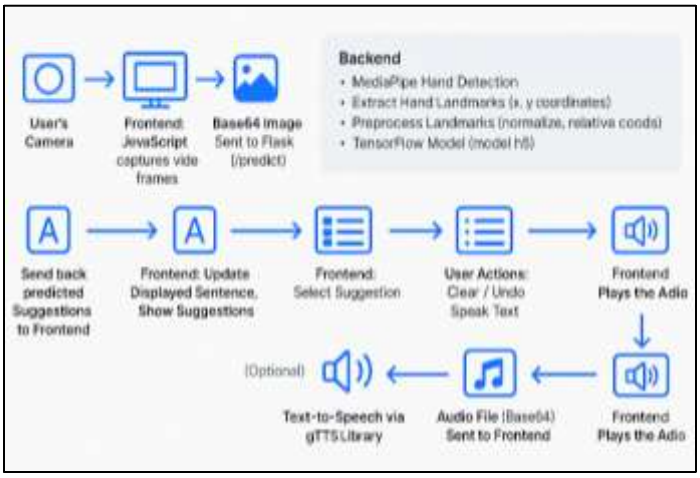

User's Camera
Frontend: JavaScript captures vide frames
Base64 Image Sent to Flask (/predict)
Backend
• MediaPipe Hand Detection
• Extract Hand Landmarks (x, y coordinates)
• Preprocess Landmarks (normalize, relative coods)
• TensorFlow Model (model h5)
Send back predicted Suggestions to Frontend
Frontend: Update Displayed Sentence, Show Suggestions
Frontend: Select Suggestion
User Actions: Clear / Undo Speak Text
Frontend Plays the Adio
Frontend Plays the Adio
Audio File (Base64) Sent to Frontend
(Optional)
Text-to-Speech via gTTS Library

Fig - 3: ISL Workflow

5. IDENTIFICATION OF GAP

The majority of current solutions are either restricted to offline processing and static datasets or concentrate on American Sign Language (ASL), despite the growing interest in sign language recognition systems. The deaf and hard-of-hearing community in India uses Indian Sign Language (ISL) extensively, but it is still notably underrepresented in research, publicly accessible datasets, and real-time recognition tools. Many of the models that are currently on the market are unsuitable for daily use because they are limited to alphabet recognition or require additional hardware, such as gloves or depth sensors. Furthermore, real-time recognition systems frequently do not integrate with audio output, which restricts their applicability in dynamic, real-world communication situations.

Additionally, there aren't many easily available and reasonably priced resources that can help ISL users and non-signers communicate, particularly in public service settings like hospitals, banks, and schools. Most importantly, direct adaptation from ASL-based models is ineffective because ISL has distinct gestures and grammar that are very different from other sign languages. The need for a reliable, portable, and scalable ISL interpretation system is underscored by these shortcomings in language-specific modeling, real-time performance, and multimodal output (text + speech).

6. RESULT

Given are a few snapshots of the Implemented Project

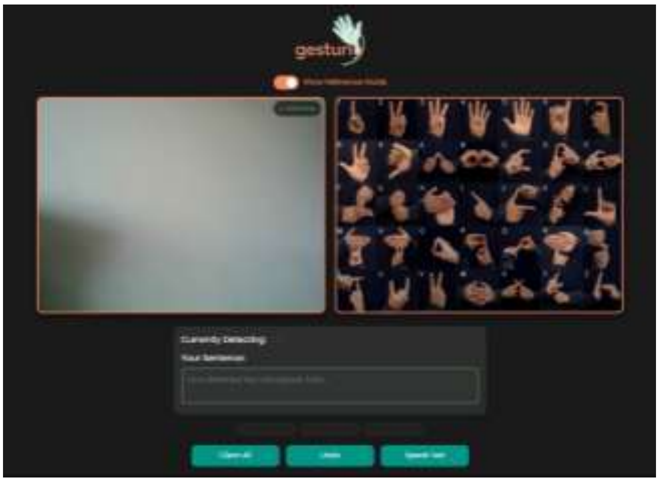
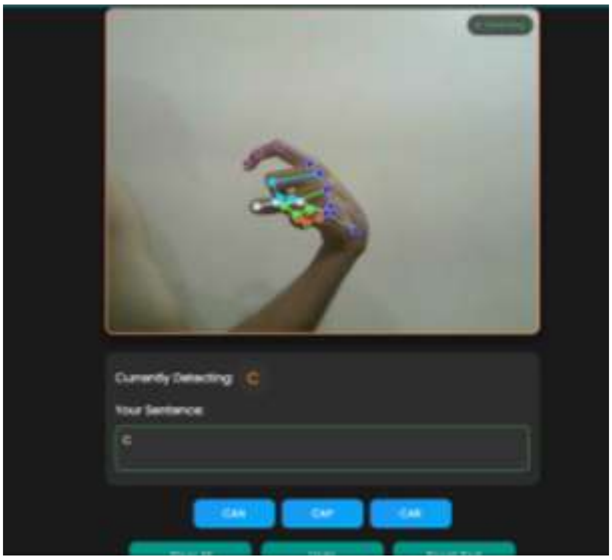


Fig - 4: Initial Page


Currently Detecting: C

Your Sentence:

C

Fig - 5: Detecting Letter 'C'

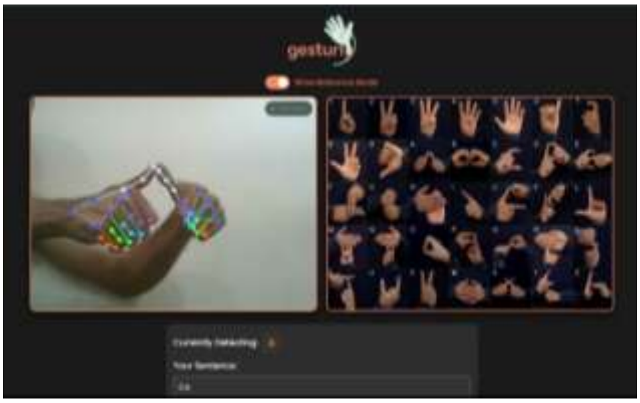
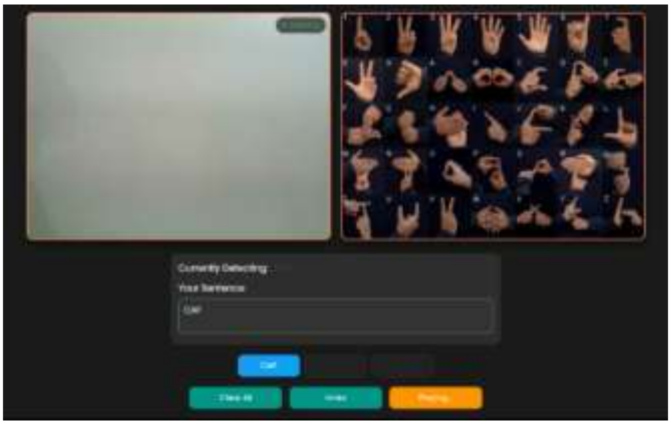


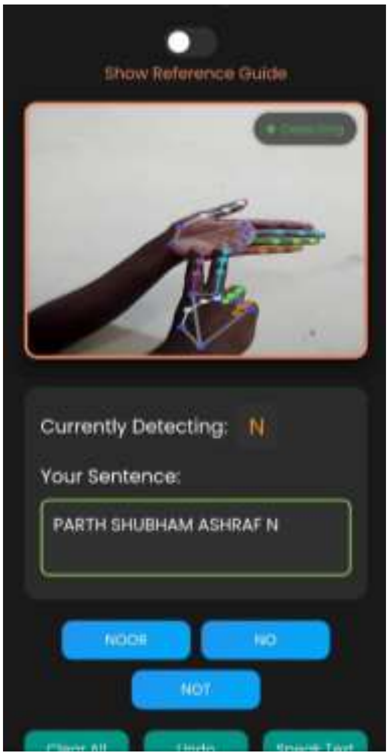
Fig - 6: Detecting Letter 'A'


Currently Detecting:

Your Sentence:

CAP

Fig - 7: Suggesting & Playing the Word 'CAP'



Currently Detecting: N

Your Sentence:

PARTH SHUBHAM ASHRAF N

Fig - 8: Project Interface on Mobile

7. CONCLUSION

In order to close the communication gap between the general public and users who are hearing or speech impaired, this project presents a real-time AI-driven Indian Sign Language (ISL) interpreter. The system offers an end-to-end pipeline from gesture capture to speech synthesis, utilizing MediaPipe for effective landmark detection, TensorFlow/Keras for precise deep learning classification, Flask for simple deployment, and gTTS for voice delivery. Because the system only requires a standard webcam and browser, it is accessible, affordable, and scalable. The interpreter provides real-time textual and audio feedback while accurately identifying 36 ISL gestures that correspond to alphabets and numbers. Future advancements like mobile deployment, vocabulary expansion, multilingual feedback, and dynamic gesture recognition are also made possible by it.

All things considered, this project is a significant step toward accessible communication technologies and gives ISL users the confidence and freedom to communicate at higher levels of education, the workplace, and society. The ISL interpreter pledges to make a lasting and positive impact on society by guaranteeing accessibility, inclusion, and equal opportunities for the differently-abled with further growth and social interaction.

8. REFERENCES

[1] Starner, T., Weaver, J., & Pentland, A. (1998). Real-Time American Sign Language Recognition Using Desk and Wearable Computer-Based Video. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[2] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR).

[3] Zhang, J., & Liwicki, M. (2012). Real-Time American Sign Language Alphabet Recognition Using Deep CNNs. Proceedings of the 24th International Conference on Artificial Neural Networks.

[4] Google Research. (2020). MediaPipe: Cross-Platform, Customizable ML Solutions for Live and Streaming Media. Retrieved from https://google.github.io/mediapipe/

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.

[6] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv preprint arXiv:1803.01271.

[7] TensorFlow Authors. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Retrieved from https://www.tensorflow.org/

[8] Google Cloud Text-to-Speech API. (n.d.). Documentation and Overview. Retrieved from https://cloud.google.com/text-to-speech