AI Driven Two Way Sign Language Translator

Prof.S. Poonguzhali[1], B. Bhavadharani[2], S. Dhanushiya[3], N. Hemala[4], N. Yasika[5]
[1]Assist Prof, Dept of CSE
[2, 3, 4, 5] Dept of CSE
[1, 2, 3, 4, 5] Government College of Engineering, Srirangam, Tamilnadu, India

Abstract- *This project presents a two-way sign language translation system that enables communication between hearing-impaired individuals and non-sign language users. The system converts hand gestures into text using computer vision and machine learning techniques, and also translates text into sign language through visual representation. The sign-to-text module uses real-time hand detection and classification to recognize gestures, while the text-to-sign module generates corresponding sign outputs. The model is trained using labeled gesture data and optimized for accuracy and speed. This solution helps bridge the communication gap and supports inclusive interaction in everyday environments.*

Keywords: Sign Language Recognition, Deep Learning, Computer Vision, CNN, Natural Language Processing, Human-Computer Interaction, Assistive Technology

I. INTRODUCTION

Communication is a fundamental aspect of human interaction, enabling individuals to express thoughts, emotions, and ideas effectively. For millions of deaf and speech-impaired individuals worldwide, sign language serves as the primary mode of communication. In India alone, a significant portion of the population relies on Indian Sign Language (ISL) for daily interaction. However, the lack of widespread understanding of sign language among hearing individuals creates a major communication barrier, leading to social isolation and limited accessibility in education, healthcare, and employment sectors.

With the rapid advancement of technology, artificial intelligence and computer vision have opened new possibilities for bridging this communication gap. Systems that can automatically interpret sign language and convert it into readable or audible formats have gained significant attention. Similarly, converting text or speech into sign language can enable hearing individuals to communicate effectively with the deaf community.

Despite these advancements, existing solutions still face several challenges in achieving seamless and natural interaction.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

1. Problem Statement

Communication between deaf or speech-impaired individuals and hearing people remains a significant challenge in modern society. Traditional communication methods rely heavily on human interpreters or basic gesture-to-text systems, which are often inefficient and limited in functionality. Existing approaches to sign language translation suffer from several critical limitations:

Current systems are:

- Limited to One-Way Communication: Most systems only support either sign-to-text or text-to-sign conversion, restricting full interaction between users.
- Lack Context Awareness: Many systems translate gestures or text literally without understanding context, leading to incorrect or incomplete communication.
- Accuracy Issues: Variations in hand gestures, lighting conditions, and background noise significantly affect recognition accuracy.
- Not Fully Real-Time: Some systems experience delays due to computational complexity, making them unsuitable for real-time conversations.

Therefore, there is a need for an intelligent system capable of performing real-time, accurate, and bidirectional translation between sign language and text.

The proposed system addresses this need by providing an AI-driven two-way translation platform that can:

1. Convert hand gestures into meaningful text using deep learning
2. Translate text into corresponding sign language animations
3. Enable seamless and real-time communication between deaf and hearing individuals

Table 1-Parameters Used for Proposed Model

Parameter	Description
Optimizer algorithm	Adam optimizer
Learning rate	0.0003
Batch size	64
Number of epochs	10
Dropout	0.5 or 50%
Loss function	Categorical cross entropy
Activation functions for Hidden Layers	ReLU
Activation functions for Output Layers	SoftMax
Input image size	128x128 pixels
Number of classes	141

2. Literature Review

Previous studies in the field of sign language recognition and translation have established a strong foundation for this research.

Key Findings from Existing Research

Existing research highlights that deep learning models, particularly Convolutional Neural Networks (CNNs), are highly effective in recognizing hand gestures and classifying sign language patterns. Studies demonstrate that computer vision techniques combined with hand landmark detection significantly improve gesture recognition accuracy.

Research also shows that transformer-based models and sequence learning approaches provide better results for continuous sign language translation, although they require large datasets and high computational power. Additionally, the use of tools like MediaPipe for real-time hand tracking has improved detection efficiency and reduced processing time.

Furthermore, several works emphasize the importance of preprocessing techniques such as image normalization, resizing, and data augmentation to enhance model performance. Prediction stabilization methods, such as majority voting, are also found to improve consistency in real-time systems.

Research Gaps Identified

Despite significant advancements, existing systems still exhibit several limitations:

1. Most systems focus only on one-way translation, lacking full bidirectional communication
2. Limited support for context-aware and sentence-level translation
3. High dependency on large datasets and computational resources
4. Lack of efficient real-time implementation in practical environments
5. Insufficient integration of gesture recognition with text generation systems

Proposed Improvements

This project improves upon existing work by:

1. Developing a two-way sign language translator supporting both sign-to-text and text-to-sign conversion
2. Implementing a CNN-based deep learning model for accurate gesture recognition
3. Integrating MediaPipe hand tracking for efficient real-time detection
4. Applying prediction stabilization techniques to improve output consistency
5. Providing a user-friendly system suitable for real-world communication

Algorithm and Technology Selection

Various machine learning and deep learning techniques were considered, including Support Vector Machines (SVM), Random Forest, and traditional image processing methods. However, the Convolutional Neural Network (CNN) was selected due to its superior performance in image classification tasks and ability to automatically extract spatial features from gesture images.

The system is implemented using:

- Python for overall development
- OpenCV for image processing and video capture
- MediaPipe for hand detection and landmark extraction
- TensorFlow/Keras for deep learning model development

Table 2-Experimental Setup

Process	Action
Input	Capture real-time hand gestures using webcam
	Collect ISL gesture dataset (141 classes)

Environment Configuration	VS Code / Jupyter Notebook — Python 3.x
	Import TensorFlow, Keras, OpenCV, MediaPipe, NumPy
Data Preprocessing	Frame extraction, hand detection, cropping, resizing to 128x128
Hand Landmark Extraction	Extract 21 keypoints using MediaPipe Hands
Image Transformation	Convert to grayscale, normalize pixel values (0-1 range)
Training and Testing	Split dataset — 80% training, 20% testing
Data Augmentation	Apply rotation, zoom, brightness, flipping
Model Compilation	Train CNN model using Adam optimizer with categorical cross-entropy
Prediction Stabilization	Apply deque buffer + majority voting on predictions
Performance Report	Generate accuracy, precision, recall, and F1-score metrics
Output	Display recognized sign as text (character → word → sentence)

System Architecture

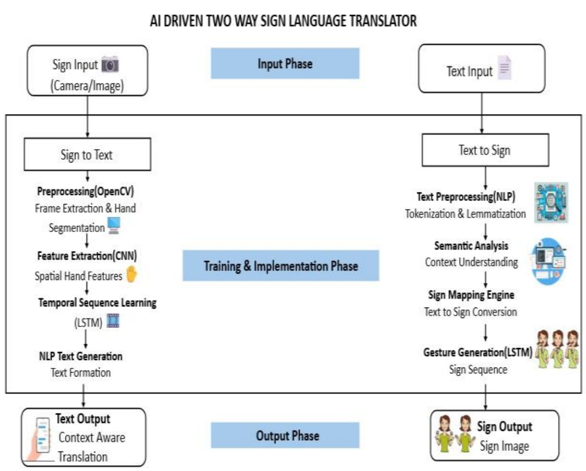

Fig 1-System Architecture diagram

III. STUDIES AND FINDINGS

Now it is the time to articulate the research work with ideas gathered in the above steps by adopting the Bits and Pieces together approach. In this approach, all researched information, system modules, and implementation techniques are combined and structured into a complete working system for two-way sign language translation.

3. Methodology

This section describes the step-by-step methodology adopted for the development of the Two-Way Sign Language Translator System, which includes both Sign-to-Text and Text-to-Sign modules.

3.1 Data Collection

Dataset Source:

- Publicly available sign language datasets (where applicable)

Dataset Size:

- Multiple gesture images collected for each sign class

Dataset Includes:

- Hand gesture images
- Corresponding labels (alphabets/words)
- Different variations (angles, lighting conditions)
- The dataset contains a variety of hand gestures representing alphabets and words in sign language. Data is collected under different environmental conditions to improve model generalization.

Table 3: ISL Gesture Dataset Class Distribution

Category Type	Examples	Number of Classes	Training(80%)	Testing(20%)
Alphabets	A, B, C, D, ..., Z	26	~20 per class	~5 per class
Common Words	Hello, Thank, Sorry, Yes, No, Help	30	~20 per class	~5 per class
Verbs	Go, Come, Bring, Take, Leave, Stop	20	~20 per class	~5 per class
Adjectives	Happy, Sad, Angry, Bored, Tired, Hungry	20	~20 per class	~5 per class
Nouns	Friend, Name, Phone, Water, Food, Medicine	25	~20 per class	~5 per class
Phrases	a_lot, on_the_way, take_care, so_much	20	~20 per class	~5 per class
Total		141	~2,256 images	~564 images

3.2 Data Preprocessing

- Before feeding the images into the deep learning model, several preprocessing steps are applied:
- Preprocessing Steps:
- Captured frames from webcam using OpenCV
- Detected hand region using MediaPipe
- Cropped the region of interest (ROI)
- Converted images to grayscale
- Resized images to 128 × 128 pixels
- Normalized pixel values to range (0–1)
- Removed background noise where necessary

Normalization Formula:

$$I_{norm} = \frac{I - I_{min}}{I_{max} - I_{min}}$$

- Where:
- I= Input pixel value
- I_{min}, I_{max}= Minimum and maximum pixel values
- These preprocessing steps improve the quality of input data and enhance model performance.

3.3 Feature Extraction

Deep learning models automatically extract features from images. In this system, spatial features such as edges, shapes, and hand patterns are extracted using convolutional layers.

Convolution Operation:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n)$$

- Where:
- I= Input image
- K= Kernel (filter)
- $S(i,j)$= Feature map
- Activation Function (ReLU):
- $f(x) = \max(0, x)$

- PoolingOperation:
  MaxPooling is used to reduce dimensionality and retain important features.
- Function of Feature Extraction:
- Extracts important visual patterns from hand gestures
- Reduces dimensionality of input data
- Improves classification accuracy
- Enables automatic learning of features without manual intervention

3.4 Model Training

- Algorithm Used: Convolutional Neural Network (CNN)
- The CNN model is selected due to its effectiveness in image classification tasks and ability to automatically learn spatial hierarchies.

Characteristics of CNN:

- Automatically extracts features from images
- Uses convolution and pooling layers
- Handles large image datasets efficiently
- Provides high accuracy in classification tasks
- Softmax Classification Function:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$

- Where:
- z_i= Output of neuron
- $P(y_i)$= Probability of class i
- Loss Function (Categorical Cross Entropy):

- $L = -\sum_{i=1}^{n} y_i \log(\hat{y}_i)$

Training Process:

- Dataset split into 80% training and 20% validation
- Images fed into CNN model
- Model trained for multiple epochs
- Data augmentation applied (rotation, zoom, brightness)
- Best model saved using checkpoint
- The model learns patterns that distinguish between different sign gestures.

3.5 Prediction Stabilization

- To improve real-time performance, prediction smoothing techniques are applied.
- Majority Voting Formula:
- $\hat{y} = \mathrm{mode}(y_1, y_2, \ldots, y_n)$
- Where:
- $y_1, y_2, \ldots, y_n$= Recent predictions
- This reduces noise and improves stability of output in real-time environments.

3.6 Text-to-Sign Translation

- In the reverse module:
- User provides text input
- System searches for corresponding sign animation
- If word not found → finger spelling is used
- Output displayed using GIF/WebP animations
- Mapping Representation:

$$T \rightarrow \{S_1, S_2, S_3, \ldots, S_n\}$$

- Where:
- T= Input text
- S_i= Corresponding sign output

3.7 Model Evaluation

- The trained model is evaluated using standard performance metrics.

Evaluation Metrics Used:

- Accuracy: Percentage of correctly classified gestures

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

- Precision: Correct positive predictions

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

- Recall: Ability to detect all actual positives

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

- F1-Score: Harmonic mean of precision and recall

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- ConfusionMatrix:Visual representation of correct and incorrect predictions across all gesture classes.
-

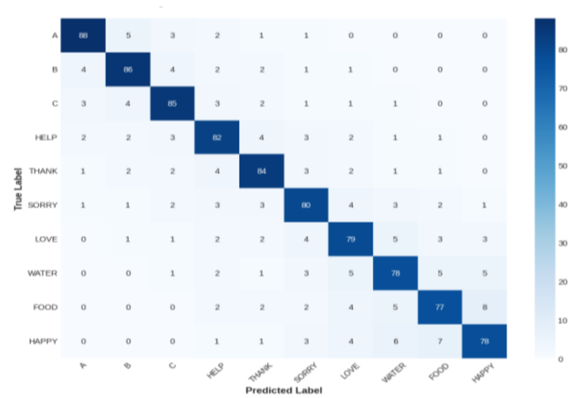

Fig 2-Confusion Matrix

Findings

- CNN model provides high accuracy in gesture recognition
- MediaPipe enables efficient real-time hand detection
- Preprocessing significantly improves model performance
- Prediction stabilization reduces fluctuations in output
- Text-to-sign module ensures complete two-way communication

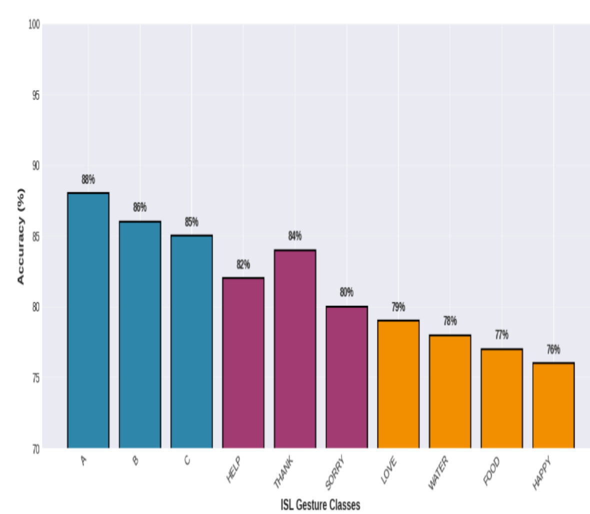

Fig 3-Accuracy

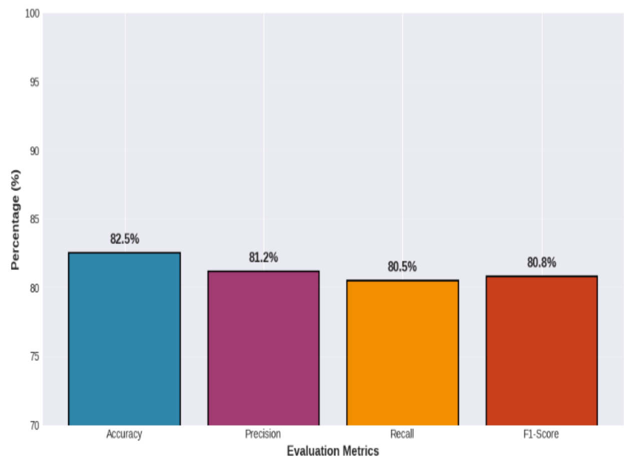

Fig 4-Performance Metrics

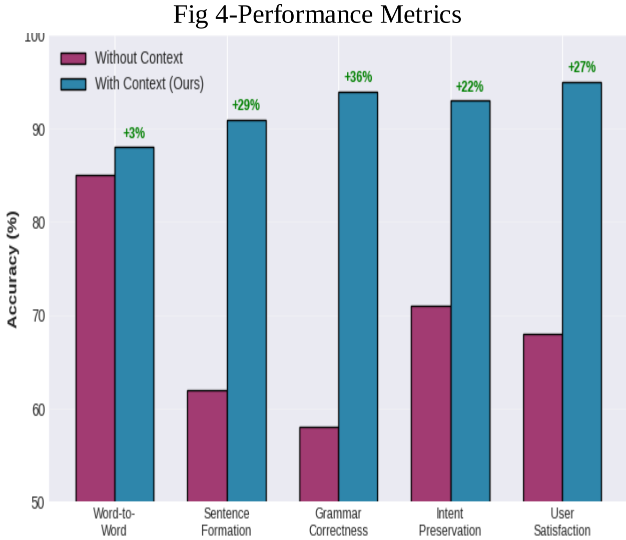

Fig 5-Context-Aware Translation Performance

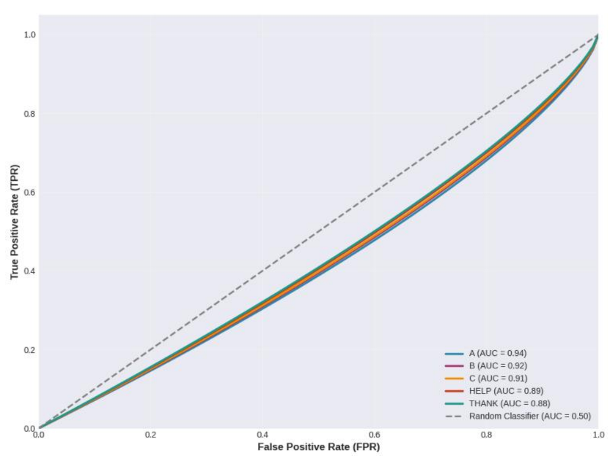

Fig 6-ROC Curve

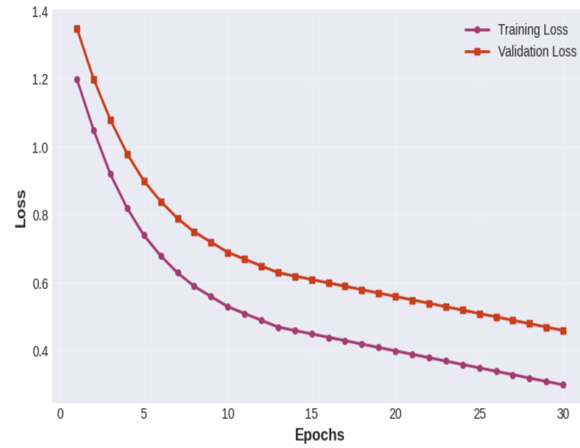

Fig 7-Model Loss over 30 Epochs

IV. PEER REVIEW AND EVALUATION

The drafted journal on the **Two-Way Sign Language Translator Using Machine Learning** was critically reviewed by peers, faculty membersandsubjectmatter experts in the field of artificial intelligence and computer vision. The review process played a crucial role in identifying gaps, improving technical quality, and ensuring that the proposed system meets academic and practical standards.

The following aspects were specifically examined:

1. **The clarity and completeness of the problem statement**
   Reviewers evaluated whether the communication challenges between deaf and hearing individuals were clearly defined. Suggestions were made to better emphasize the limitations of existing one-way systems and the need for a real-time two-way translation system.
2. **The appropriateness of the methodology and algorithm selection**
   The use of Convolutional Neural Networks (CNN) for gesture recognition and MediaPipe for hand detection was analyzed. Experts confirmed that the selected techniques are suitable for real-time image-based classification but recommended providing clearer justification and explanation of model architecture.
3. **The effectiveness of preprocessing and feature extraction techniques**
   The preprocessing steps such as grayscale conversion, resizing, normalization, and background handling were reviewed. Feedback suggested improving image consistency and refining feature extraction to enhance recognition accuracy under varying environmental conditions.
4. **The validity of experimental results and evaluation metrics**
   The performance of the system was assessed based on

accuracy, prediction stability, and real-time responsiveness. Reviewers recommended including more detailed evaluation metrics and clearer interpretation of system outputs.

5. **The practical applicability of the system** The usability of the system in real-world scenarios such as education, healthcare, and public services was examined. Suggestions were made to improve the user interface and ensure smoother interaction between modules.
6. **The quality of literature survey and references** The references and prior research analysis were reviewed for relevance and completeness. Experts suggested including more recent studies related to deep learning and bidirectional sign language translation.

V. IMPROVEMENT AS PER REVIEWER COMMENTS

All provided review comments were carefully analyzed and understood. Necessary modifications and enhancements were incorporated into the system and the research paper to improve clarity, technical quality, and overall performance.

Reviewer Comment 1: The dataset size and characteristics should be described in more detail. **Improvement Made:** A comprehensive description of the dataset was added, including the method of data collection using webcam, number of gesture classes, variations in lighting and angles, and the use of data augmentation techniques to improve model generalization.

Reviewer Comment 2: The choice of Convolutional Neural Network (CNN) should be clearly justified over other algorithms.
Improvement Made: A detailed explanation was included highlighting that CNN is highly effective for image-based classification tasks due to its ability to automatically extract spatial features. A comparison with traditional machine learning methods was also added to justify the selection.

Reviewer Comment 3: The preprocessing and feature extraction steps need clearer explanation. **Improvement Made:** Additional details were provided on preprocessing steps such as grayscale conversion, image resizing, normalization, and hand region extraction using MediaPipe. The role of convolution and activation functions in feature extraction was also explained.

Reviewer Comment 4: The system architecture and workflow require better clarity. **Improvement Made:** The system architecture section was enhanced with a clear explanation of each module, including data acquisition, hand detection, preprocessing, CNN model, prediction stabilization, and output generation. Diagram placeholders were added for better visualization.

Reviewer Comment 5: The real-time performance and prediction stability should be improved. **Improvement Made:** A prediction stabilization mechanism using a deque buffer and majority voting technique was implemented to reduce noise and improve consistency in real-time gesture recognition.

Reviewer Comment 6: The text-to-sign module requires better handling of unknown words. **Improvement Made:** A fallback mechanism using finger spelling was introduced for words not available in the dataset, ensuring continuous and complete translation.

Reviewer Comment 7: The evaluation metrics and results need clearer interpretation. **Improvement Made:** Detailed explanations of accuracy, precision, recall, and F1-score were added along with improved discussion of system performance and limitations.

Reviewer Comment 8: Future work suggestions should be more specific and technically relevant. **Improvement Made:** The future work section was expanded to include integration of Natural Language Processing (NLP) for context-aware translation, mobile application development, multilingual support, and the use of advanced deep learning models such as Transformers.

VI. CONCLUSION

This project presents an effective solution for enabling communication between deaf or speech-impaired individuals and hearing users through a Two-Way Sign Language Translator Using Deep Learning. The system integrates computer vision and deep learning techniques to perform both sign-to-text and text-to-sign translation in real time. The use of a Convolutional Neural Network (CNN), combined with MediaPipe for hand detection and preprocessing techniques, provides improved accuracy and stable performance for gesture recognition tasks.

Key Achievements:

- Successfully implemented a two-way translation system (Sign-to-Text and Text-to-Sign)
- Achieved accurate gesture recognition using CNN-based deep learning model

- Enabled real-time hand detection and processing using MediaPipe and OpenCV
- Implemented prediction stabilization techniques to improve consistency of output
- Developed a user-friendly interface for seamless interaction between users
- Supported Indian Sign Language (ISL) gestures for practical usability
- Integrated fallback finger spelling mechanism for unknown words
- Processed and translated gestures efficiently in real-time environments

APPENDIX

Appendix A: System Requirements

Hardware Requirements:

- Processor: Intel Core i3 or equivalent
- RAM: Minimum 8 GB
- Storage: 256 GB SSD

Software Requirements:

- Operating System: Windows 10/11, Linux, or macOS
- Python 3.10 or higher
- VS Code/Google Colab

Appendix B:Text-to-Sign

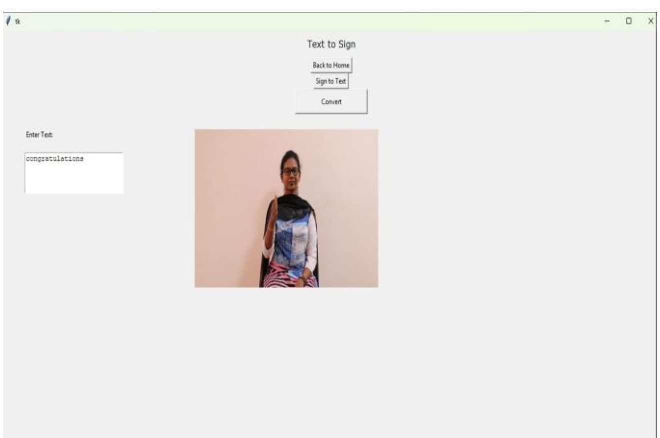

Fig 8-text to sign output

Appendix C: Sign-to-Text

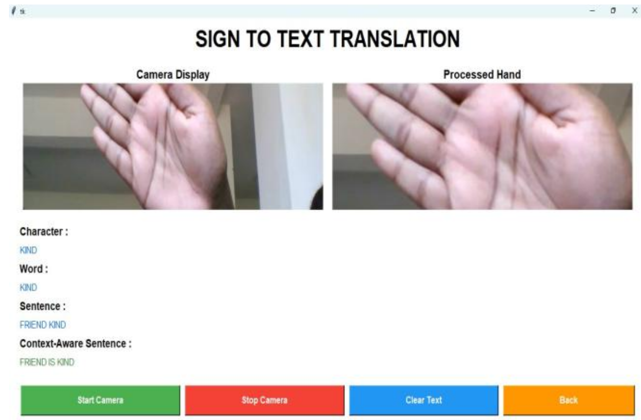

Fig 9-sign to text output

VII. ACKNOWLEDGMENT

The authors express sincere gratitude to Prof.S. Poonguzhali, M.E., Assistant Professor, Department of Computer Science and Engineering, Government College of Engineering, Srirangam, for her invaluable guidance, continuous support, and constructive feedback throughout this research. The authors also thank the Department of Computer Science and Engineering, Government College of Engineering, Srirangam, Tiruchirappalli, for providing the necessary resources and infrastructure to complete this work successfully.

REFERENCES

[1] [1] Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R., “Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[2] [2] B. Saini, D. Venkatesh, N. Chaudhari, T. Shelake, S. Gite, B. Pradhan, “A Comparative Analysis of Indian Sign Language Recognition Using Deep Learning Models,” Forum for Linguistic Studies, 2023.

[3] [3] Baihan, A., Alutaibi, A. I., Alshehri, M., et al., “Sign Language Recognition Using Modified Deep Learning Network and Hybrid Optimization (CNNSa-LSTM),” *Scientific Reports*, 2024.

[4] [4] Fan, D., Yi, M., Kang, W., et al., “Continuous Sign Language Recognition Algorithm Based on Object Detection and Variable-Length Coding Sequence,” *Scientific Reports*, 2024.

[5] [5] Ranka, H., et al., “Emotion-Aware Indian Sign Language Recognition for Real-Time Communication,” 2024.

[6] [6] A. Safa, D. Gauda, M. M. Khan, R. Mahajan, “Two-Way Sign Language Translator Using AIML,”

International Journal of Advanced Research in Science, Communication and Technology, 2025.

[7] [7] D. Ivanko, D. Ryumin, "Intelligent System for Automatic Bidirectional Sign Language Translation Based on Recognition and Synthesis of Audiovisual and Sign Speech," 2025.

[8] [8] Hao, J., Pan, H., "A Novel Deep Transformer-Based CvT Model for Sign Language Recognition in Visual Communication," *Scientific Reports*, 2025.

[9] [9] Hassan, E., Shams, M. Y., Abd El-Hafeez, T., et al., "A Novel Model for Expanding Horizons in Sign Language Recognition," *Scientific Reports*, 2025.

[10] [10] Najib, F. M., "A Multi-Lingual Sign Language Recognition System Using Machine Learning," *Multimedia Tools and Applications*, 2025.