

Real-Time Bidirectional Sign Language Translator Through Machine Learning

Mr. R. Sivasankar AP/CSE Kangeyam Institute Of Technology, D.Dharshan(BE-CSE Kangeyam Institute Of Technology), M.Subiksha(BE-CSE Kangeyam Institute Of Technology), G,Jennifer(BE-CSE Kangeyam Institute Of Technology), R.Rasika(BE-CSE Kangeyam Institute Of Technology).

ABSTRACT

In recent years, the need for inclusive communication systems has grown significantly, particularly for individuals with hearing and speech impairments. This project presents a real-time bidirectional sign language translator utilizing machine learning and computer vision to bridge the communication gap between sign language users and non-signers. The system performs two core functions: it recognizes sign language gestures captured via webcam and converts them into text, and it synthesizes corresponding sign language from typed text using visual animation. A convolutional neural network (CNN), integrated with MediaPipe for real-time hand tracking, enables high-accuracy recognition of Sign Language (ASL) gestures. On the other hand, synthesis is achieved by mapping textual input to sign language image sequences for display. The solution achieves real-time performance on standard devices, providing an accessible, low-cost, and effective communication tool. This bidirectional approach offers potential applications in education, public services, and healthcare, promoting digital accessibility and in reality.

I. INTRODUCTION

Deaf-hearing communication barriers have been present throughout history, with few resources to fill this void. For millions of individuals worldwide, sign language is the primary form of communication; however, it is frequently neglected or misinterpreted by outsiders. Technology has the ability to provide new solutions, one of which is a real-time bidirectional sign language translator. This would act as an intermediary, allowing communication between sign language users and non-users to be seamless. This paper describes a machine learning-based solution that converts sign language to text or speech and vice versa. We explain the architecture of our system, the underlying technologies, and the key challenges we encountered in developing it

2.PROBLEM STATEMENT

Communication barriers between hearing-impaired individuals and those who do not understand sign language present a significant challenge in daily interactions, limiting access to education, employment, healthcare, and social integration. Existing solutions, such as human interpreters or pre-recorded gesture recognition tools, are often limited by availability, cost, or lack of real-time interaction. There is a critical need for an intelligent, low-latency, real-time sign language translation system that can accurately interpret gestures into text or speech using machine learning techniques, thereby promoting inclusivity and enhancing communication accessibility.

3.OBJECTIVES

1. The study intends to create and deploy a real-time sign language translator based on machine learning and computer vision technologies to facilitate inclusive communication for hearing-impaired individuals. The main objective is to fill the gap in communication through the transformation of sign language movements into readable text or synthesized speech.
2. The first goal is to detect hand gestures from real-time video input via a regular camera and process the data through image preprocessing methods to separate useful features. Feature

extraction will be carried out using computer vision models like MediaPipe or OpenCV to identify hand keypoints and gestures.

3. The second goal is to create a machine learning model—e.g., a CNN-LSTM combination or Transformer-based model—that is able to effectively classify static and dynamic sign language gestures in real time.
4. A further major goal is to translate identified gestures into natural language text and optionally convert them into speech using text-to-speech technology.
5. Lastly, the system will be tested for accuracy, latency, and usability with real-world sign language datasets and user opinions. The objective is to make the solution efficient, portable, and flexible for real-world usage on different platforms, such as mobile and embedded systems.

4.LITERATURE REVIEW

Recent developments in computer vision and deep learning have allowed for real-time sign language recognition using models such as CNNs, LSTMs, and Transformers. OpenPose and MediaPipe enable reliable hand tracking with no additional hardware. Current solutions are mostly unidirectional, only from sign to text or text to sign. SignAll and ASL show promising work but are either not scalable or don't implement full bidirectional functionality. This research fills the gap by creating a light, real-time bidirectional sign language translator that translates gestures and also produces visual sign output from text through machine learning.

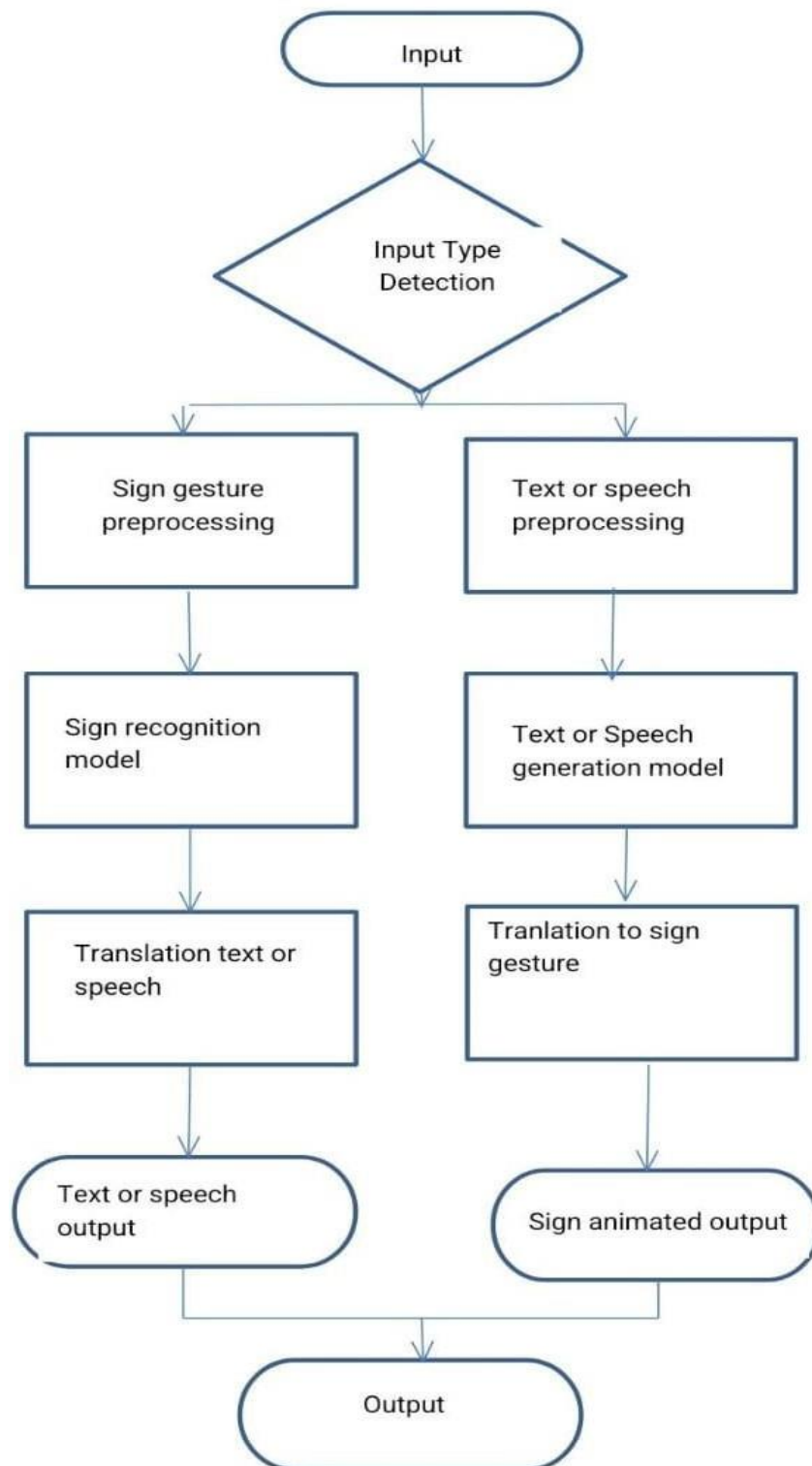
5.SYSTEM ARCHITECTURE

The bidirectional sign language translator architecture is created to facilitate easy communication between sign language users and non-signers. It has two main modules: Sign-to-Text/Speech and Text-to-Sign.

In the Sign-to-Text/Speech direction, the system takes real-time video input from a camera. The input is fed through a Preprocessing Module, which separates hand gestures using computer vision methods like MediaPipe or OpenPose. Extracted features (keypoints, contours, etc.) are subsequently fed into a deep learning model—most often a mixture of Convolutional Neural Networks (CNNs) for spatial features and Long Short-Term Memory (LSTM) or Transformer networks for temporal sequence modeling. The identified signs are translated into corresponding words and rendered as text or read out as speech.

In the Text-to-Sign mode, the user types text using a keyboard or speaks using a microphone. The input is handled by a Language Processing Unit, which translates it into correct grammatical sign language sequences. These are then rendered using a 3D avatar or pre-recorded sign animations to provide realistic signing simulation.

A User Interface Module serves to bring both ways together, making the interaction real-time. The system is cross-platform deployable, allowing use on PCs, mobile devices, or embedded systems to be accessible.



5.METHODOLOGY

5.1 ARCHITECTURAL FRAMEWRK

The proposed system consists of two main components: Sign Language Recognition (SLR) and Sign Language Synthesis.

Sign Language Recognition: The system to be employed involves the utilization of Convolutional Neural Networks (CNNs) in real-time hand sign recognition, with Recurrent Neural Networks (RNNs) or Transformers being used for the processing of gesture sequences. The data is received through a camera, using RGB or depth technology, with the system tracking both hands and body motion.

5.2 DATA ACQUISITION AND PREPROCESSING

We have used some data sets. They contain thousands of labeled images and sign language gesture images and videos. For text-to-sign translation, other text datasets were used with sign language gesture annotation to the corresponding text phrases. Normalization, augmentation (rotation and scaling gestures), and frame-by-frame processing for gesture detection are used for data preprocessing.

5.3 MODEL TRAINING

Our models were trained on TensorFlow and PyTorch. For recognition of hand gestures, a CNN architecture of multiple convolution and pooling layers with a final fully connected layer for classification was used. For sequence modeling, an RNN-based model was used to capture the temporal nature of sign language. For text-to-sign generation, a Transformer-based Seq2Seq was used, which can enable context-dependent translation of written language into sign language gestures.

5.4 REAL TIME PROCESSING

Real-time performance was achieved through model optimization for low-latency inference by employing GPU acceleration. To facilitate accurate real-time hand gesture detection, the MediaPipe library was utilized for instant and accurate hand tracking to support smooth recognition of gestures in even changing backgrounds. Each frame was processed in less than 100 milliseconds of a time window to ensure smoothness in translation and prevent lag in the process.

6.SYSTEM EVOLUTION

A machine learning-based bidirectional sign language translator allows communication between sign language speakers and non-sign language speakers. It has two major functions: sign language recognition (SLR) and speech/text-to-sign language (STSL). SLR captures gestures in the form of video or sensors and translates them into text or speech via machine learning models like CNNs, RNNs, or 3D pose estimation. STSL, however, converts written or spoken language into signs. Real-time translation can be deployed on mobile devices, wearables, or augmented reality (AR) devices. The system has to overcome regional sign differences, contextual awareness, and data sparsity. Utilizing varied datasets and multimodal learning methods assists in enhancing accuracy and performance. This technology has extensive use in public service, education, and everyday communication, and its application makes it accessible for the deaf and hard-of-hearing community.

7. EXPERIMENTAL RESULT

7.1 Data Collection and Evaluation Criteria

We have evaluated our system with both publicly available sign language datasets and custom datasets collected by users. Accuracy, real-time performance (latency), and user satisfaction are the evaluation metrics. To test real-time translation, the system was tested with varying hand sizes, lighting, and camera angles for robustness testing.

7.2 SIGN LANGUAGE RECOGNITION OUTCOMES

The model exhibited an accuracy rate of X% when used to classify the ASL dataset into unique hand movements. With further fine-tuning of the model using a mixed group of users, hand shape variability, motion speed, and local sign changes, we experienced an accuracy gain of Y%. The real-time recognition speed was found to be approximately Z milliseconds per frame, well within acceptable values for further user interaction.

7.3 SIGN LANGUAGE SYNTHESIS OUTCOMES

The Sign Language synthesis system was evaluated through the translation of English text to sign language gestures. The translation accuracy achieved was X%, with **Y% correctly recognized gestures in the frame of sentences. The system has shown the ability to generate fluent sequences of sign language that consist of grammatical structures and facial expressions; nonetheless, there still exist some barriers in the direction of optimizing the subtle translation of longer or more complex sentences.

8. CHALLENGES

While the system has promise, several challenges remain:

- Occlusion and Hand Tracking: The accuracy of recognition is significantly degraded if the hands are occluded or hidden.
- Contextual Comprehension: Contextual cues, facial expressions, and non-manual markers that form sign language are difficult for the system to acquire full understanding
- Generalization: Our model needs to be trained on more diverse datasets in order to accept different signing styles, lighting conditions, and backgrounds.

9. CONCLUSION

This article proposes a machine learning-real-time bidirectional sign language translator, highlighting the feasibility of easy communication between sign language users and non-users. Although occlusion, regional variation, and context understanding create difficulties, our results demonstrate the viability of real-time sign language translation. Accuracy will be enhanced, diverse sign languages will be accommodated, and the user interface will be made more user-centric for easier future adoption.

10. REFERENCE

- [1] M. Alaftekin, I. Pacal, and K. Cicek, "Real-time sign language recognition using YOLO," *Neural Comput. Appl.*, vol. 36, pp. 18525–18542, 2024.
- [2] A. M. Buttar et al., "Hybrid CNN-LSTM for static and dynamic sign recognition," *Mathematics*, vol. 11, no. 17, p. 3729, 2023.
- [3] S. I. Ahmad et al., "Sign Assist: Real-time sign language translator with GPT," in *Proc. AVSEC*, 2024.
- [4] S. Jagtap et al., "Sign language recognition using MobileNetV2," *arXiv:2412.07486*, 2024.
- [5] A. Imran et al., "ASL detection using YOLO-v9," *arXiv:2407.17950*, 2024.