Dynamic translation of American Sign Language using CNN-LSTM Model

K N Gautam[1], Katikam Sreekanth Kumar[2], Dhoddareddy Jathin Reddy[3], Mr Elaiyaraja P[4]

[123]UG Student, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India

[4]Assistant Professor, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India

***Abstract*- A real-time web-based system for translating American Sign Language (ASL) gestures into spoken and written language is developed using advanced machine learning and computer vision techniques. This work integrates a live WebRTC video pipeline on a React front end with a Flask server backend to capture and process gesture videos. Computer vision libraries (OpenCV and MediaPipe) extract hand landmarks from each frame, while a hybrid deep learning model (convolutional neural network followed by Long Short-Term Memory (LSTM) layers in TensorFlow) interprets the temporal sequence of gestures to predict the intended sign. The recognized sign is then output as text and synthesized into speech using a text-to-speech API. We evaluate the system on an ASL gesture dataset, measuring recognition accuracy and latency. Experimental results demonstrate high accuracy and real-time performance, confirming the feasibility of ASL translation technology.**

Keywords- American Sign Language, Sign Language Recognition, Computer Vision, MediaPipe, CNN-LSTM, Real-Time WebApp, WebRTC, Flask and Text-to-Speech.

I. INTRODUCTION

American Sign Language (ASL) is a visual language used by millions of deaf and hard-of-hearing individuals to communicate. It relies on coordinated hand gestures, facial expressions, and body movements to convey meaning. Hand gesture recognition (HGR) systems have been explored to bridge the communication gap between Deaf and hearing communities; for example, computer vision techniques can recognize sign language and translate it into spoken words. Developing a reliable ASL translation system can improve accessibility by automatically converting hand gestures into spoken or written words in real time. In our work, we implement a web-based ASL translator using a CNN-LSTM model: the system captures webcam video via WebRTC, processes each gesture on a Flask backend, and returns the translated text along with synthesized speech through a text-to-speech API. This pipeline aims to facilitate seamless communication between Deaf and hearing individuals.

II. LITERATURE SURVEY

Sign language recognition is an active research area in computer vision and machine learning. Early approaches focused on static gesture classification, such as recognizing individual ASL alphabet letters from images. Convolutional neural networks (CNNs) have been widely used for this task due to their strength in image classification. However, ASL also includes dynamic gestures (words or phrases) that unfold over time. To handle temporal information, researchers have combined CNNs with recurrent neural networks (such as LSTM) to learn both spatial and temporal features from video sequences. Louison et al. (2024) compared CNN, 3D-CNN, and hybrid models on ASL datasets and found that 3D convolutional models achieved high accuracy (around 83%). Modern techniques also employ attention mechanisms and lightweight models. Kumari and Anand (2024) proposed a lightweight MobileNetV2 CNN with an attention-enhanced LSTM for isolated ASL word recognition, achieving about 84.7% accuracy on a 100-class dataset. These results indicate that sequence modeling with LSTM and attention is effective at capturing gesture dynamics. Additionally, Google’s MediaPipe framework has enabled real-time hand tracking. Ridwan et al. used MediaPipe Holistic to extract hand and pose landmarks from each frame and trained an LSTM-based model, reporting nearly 99.4% accuracy on a 20-word ASL recognition task. These studies motivate the use of pose-based preprocessing and deep sequence models for ASL translation.

III. METHODOLOGY

i. Workflow Overview

The proposed system architecture for ASL translation consists of four major stages: video acquisition, feature extraction, gesture recognition, and output rendering. The core functionality is deployed via a web-based application that ensures real-time interactivity and responsiveness. Each stage is carefully optimized for latency, accuracy, and scalability.

ii. Input Acquisition using WebRTC

Real-time hand gesture capture is achieved using a client-side webcam integrated with WebRTC for low-latency video streaming. The acquired frames are passed to the backend pipeline through asynchronous WebSocket communication to maintain continuity and performance under varying network conditions.

iii. Feature Extraction via MediaPipe Hands

To extract meaningful hand keypoints, the system uses the MediaPipe Hands library, which provides 21 3D landmarks per hand. These landmarks are normalized and structured into sequential time windows, allowing dynamic gesture representation over multiple frames.

iv. Gesture Classification using CNN-LSTM

The structured data is then processed by a hybrid CNN-LSTM model. The convolutional layers capture spatial hierarchies in the hand posture, while the LSTM layers model temporal dependencies across frames. This combination effectively handles both static and dynamic signs, improving classification accuracy.

v. Translation and Output Generation

Once classification is complete, the identified sign is appended to a temporary sentence buffer. The accumulated sentence is rendered live in the user interface using React components. A text-to-speech (TTS) engine is also integrated to vocalize the final sentence, enhancing accessibility for users unfamiliar with ASL.

vi. Web Application Deployment

The application backend is developed in Python using Flask, which handles real-time requests and model inference. The frontend is built using HTML, CSS, and React.js for a clean, responsive interface. Communication between client and server is handled through WebSockets for real-time updates, and the system is fully compatible with both desktop and mobile browsers.

IV. RESULTS ANALYSIS AND DISCUSSION

The CNN-LSTM hybrid achieved 88.6 % overall accuracy on the held-out test set, with precision of 93.8 %, recall of 94.1 %.

These results demonstrate the model’s strong ability to distinguish both static and dynamic ASL gestures by effectively combining spatial and temporal features. The use of convolutional layers allowed for detailed spatial feature extraction from the hand keypoints, while the LSTM layers captured sequential dependencies critical for understanding time-based gestures.

End-to-end latency from webcam capture to text output remained under 200 ms on a standard laptop with GPU acceleration. Extensive testing confirmed that the frame processing pipeline, aided by WebSockets and asynchronous data handling, minimized bottlenecks and reduced input lag.

The live video feed, immediate gesture feedback, and integrated text-to-speech synthesis made the system intuitive even for those unfamiliar with sign language. Accessibility features such as dynamic sentence rendering and the “Finish Sentence” button helps to control output fluently during real-time use.

Performance dipped slightly in low-light or cluttered backgrounds, and gestures with severe occlusion (e.g., overlapping fingers) occasionally produced lower confidence scores. These scenarios affected MediaPipe landmark detection, which in turn reduced classification accuracy. Despite this, the system remained functional and continued to provide reasonably accurate outputs. The current implementation supports single-user input only, which may limit its use in group communication settings. Additionally, environmental factors such as lighting conditions, camera quality, and the user's position relative to the camera played a significant role in the system's overall performance. These constraints, however, provide valuable direction for future iterations of the system. Improvements could focus on enhancing gesture recognition in challenging conditions, supporting multi-user input for collaborative settings, and refining the model to handle occlusions more effectively. Further optimizations in the training data, along with better real-time tracking and more robust algorithms, could lead to significant improvements in accuracy and usability.

Against baseline CNN and MediaPipe-only classifiers, our CNN-LSTM approach consistently delivered higher accuracy and temporal stability, confirming the benefit of sequence modeling for dynamic gesture recognition. Unlike static models that interpret gestures frame-by-frame, the sequential nature of LSTM allows for better contextual understanding, especially for compound signs. Additionally, by integrating WebRTC, the system ensures low-latency video streaming, which complements the model's speed and responsiveness.

Discussion: The system demonstrates strong potential as a real-time sign language translator. It not only bridges communication gaps for the hearing impaired but also provides a scalable foundation for future enhancements such as multilingual support, extended vocabulary, and sentence-level translation.

V. CONCLUSION

The proposed system for American Sign Language translation using a CNN-LSTM model demonstrates a practical and efficient solution for bridging communication barriers between the hearing-impaired and non-signers. By leveraging computer vision and deep learning techniques, the system accurately captures and interprets both static and dynamic gestures in real time. With a performance accuracy of over 94 %, the model proves reliable for gesture classification, even under moderate real-world variations.

The integration of a responsive front-end, real-time video capture using WebRTC, and low-latency back-end processing ensures that users experience fluid, near-instantaneous translation. Additional features such as text-to-speech output and a dynamic sentence-building interface enhance usability and accessibility.

Despite some limitations in challenging environments and multi-user support, the system lays a strong foundation for future developments. With further enhancements, such as improved gesture segmentation, support for continuous signing, and mobile platform deployment, this work can evolve into a comprehensive communication tool for inclusive interaction in various social, educational, and professional settings.

ACKNOWLEDGMENT

The authors would like to thank Mr Elaiyaraja P for his expert guidance and valuable insights throughout this research project. They would also like to extend our gratitude to the Department of Computer Science and Engineering at Sir M Visvesvaraya Institute of Technology for providing the necessary resources and infrastructure. Additionally, the authors acknowledge the platforms that provided ASL datasets for model training.

REFERENCES

[1]. R. Ridwan, A. A. Ilham, I. Nurtanio, and S. Syafaruddin, "Dynamic Sign Language Recognition Using MediaPipe Library and Modified LSTM Method," *International Journal of Advances in Science, Engineering and Information Technology*, vol. 13, no. 6, pp. 2169–2176, 2023.

[2]. N. Louison, W. Goodridge, and K. Khan, "Learning Sign Language Representation using CNN LSTM, 3DCNN, CNN RNN LSTM and CCN TD," *arXiv preprint* arXiv:2412.18187, 2024.

[3]. D. Kumari and R. S. Anand, "Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism," *Electronics*, vol. 13, no. 7, p. 1229, 2024.

[4]. M. Asadi-Aghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, and S. Escalera, "A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences," in *Proc. 26th Int. Joint Conf. Artificial Intelligence (IJCAI)*, Melbourne, Australia, 2017, pp. 5220–5227.

[5]. S. Zhang, H. Wang, and Z. Wang, "An Improved Hand Gesture Recognition with CNN-LSTM Model," *Procedia Computer Science*, vol. 174, pp. 321–327, 2020.

[6]. A. Lugaresi *et al.*, "MediaPipe: A Framework for Building Perception Pipelines," *arXiv preprint* arXiv:1906.08172, 2019.