International Journal of Scientific Research and Engineering Development-– Volume 8 Issue X, 2026

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

A REAL-TIME BIDIRECTIONAL AMERICAN SIGN LANGUAGE COMMUNICATION SYSTEM USING MEDIAPIPE AND DEEP LEARNING

SUSHMA C K

Department of Computer Science & Engineering, East West Institute of Technology, Bangalore, India

Email: sushmack238@gmail.com

----------**********----------

Abstract:

Communication Between Individuals Who Use American Sign Language (Asl) And People Who Do Not Understand Sign Language Remains A Significant Accessibility Challenge. This Paper Presents Signfusion, A Real-Time, Bidirectional Asl Communication System Designed To Bridge This Communication Gap Using Computer Vision, Machine Learning, Speech Processing, And Web Technologies. The System Is Implemented Using Python, Opencv, Mediapipe, Tensorflow/Keras, Scikit-Learn, Speechrecognition, Pyttsx3/Gtts, Flask, And Socket.Io. Experimental Evaluation Reported 97.3% Recognition Accuracy On The Standard Asl Alphabet Test Dataset And Approximately 91.6% Recognition Accuracy Under Live Conditions. The Random Forest Pipeline Achieved An Average Frame-To-Dashboard Latency Of 56 Ms, While The System Maintained Real-Time Operation At Approximately 28–32 Frames Per Second On Standard Hardware. These Results Demonstrate The Feasibility Of A Software-Only, Low-Cost System For Real-Time Bidirectional Asl Communication.

Keywords— American Sign Language, Asl Recognition, Mediapipe, Computer Vision, Convolutional Neural Network, Random Forest, Speech Recognition, Text-To-Speech, Real-Time Communication.

----------**********----------

I. INTRODUCTION

Communication is fundamental to human interaction, allowing individuals to exchange information, express emotions, request assistance, and participate in social activities. However, communication can become challenging when individuals who use sign language interact with people who do not understand it. American Sign Language is a complete visual language that uses hand shapes, movements, and other visual cues. The communication barrier between ASL users and non-signers creates a need for accessible technological solutions.

Traditional sign-language recognition systems have explored wearable sensors, data gloves, depth cameras, and other specialized hardware. Although these approaches can provide useful recognition capabilities, they can increase cost, reduce portability, and make deployment less convenient. The SignFusion project therefore focuses on a software-based approach using commonly available hardware such as a webcam, microphone, and standard computer.

Recent advances in computer vision and machine learning make it possible to extract hand landmarks from ordinary camera streams and classify gestures in real time. MediaPipe Hands provides a mechanism for detecting hand landmarks, while machine learning models can use these features for gesture classification. The proposed system combines these technologies with speech

recognition, text-to-speech, and a web-based interface to support communication in both directions.

The primary objective of this work is to develop and evaluate a real-time bidirectional ASL communication system that operates using standard consumer hardware. The project specifically investigates real-time hand gesture capture, landmark normalization, CNN-based classification, Random Forest classification, sentence formation, speech-to-text conversion, text-to-speech conversion, and text-to-ASL visualization.

II. LITERATURE SURVEY

Sign language recognition has been studied using several technological approaches, including wearable sensors, depth cameras, computer vision, machine learning, and deep learning.

Hardware-Based Approaches: Early sign-language recognition systems commonly used data gloves and wearable sensor arrays to capture finger movements, joint angles, and wrist orientation. Fang et al. investigated large-vocabulary sign-language recognition using fuzzy decision trees. Neverova et al. investigated multi-scale deep learning for gesture detection and localization [4]. Although sensor-based approaches can capture useful physical information, their dependence on specialized equipment can reduce accessibility and portability.

Depth-Camera and Skeleton-Based Approaches: Depth-camera systems introduced another approach to gesture recognition. Zafrulla et al. investigated ASL recognition using the Microsoft Kinect sensor . Cooper and Bowden explored large-lexicon sign-language detection. Such systems can provide skeletal information, but the requirement for dedicated depth-sensing hardware can limit deployment in ordinary environments.

Vision-Based Deep Learning: The development of CNNs significantly expanded vision-based sign-language recognition. Pigou et al. investigated sign-language recognition using convolutional neural networks. Huang et al. explored 3D convolutional neural networks for sign-language recognition [8]. Koller et al. studied continuous sign-language recognition for multiple signers. Camgoz et al. subsequently explored neural sign-language translation.

MediaPipe-Based Recognition: MediaPipe Hands introduced an efficient method for real-time hand landmark detection. Zhang et al. described a system capable of providing 21 three-dimensional hand keypoints at interactive frame rates. Later approaches used MediaPipe landmarks with lightweight classifiers such as SVM and Random Forest for gesture recognition.

Text-to-Sign Conversion: Compared with sign-to-text recognition, text-to-sign conversion has received comparatively less attention. Karpouzis et al. investigated an architecture for Greek Sign Language synthesis , while Stoll et al. explored neural machine translation and GAN-based sign-language production .

Research Gap: The literature demonstrates significant progress in gesture recognition; however, many systems focus primarily on one-way recognition or depend on specialized hardware. SignFusion addresses this gap by combining real-time gesture recognition with speech processing and text-to-ASL visualization in a single bidirectional software-based platform.

III. PROPOSED SYSTEM

SignFusion is designed as a real-time communication system with two operating modes: ASL-to-Speech and Speech-to-ASL. In the first mode, the webcam captures hand gestures. MediaPipe extracts hand landmarks, after which the recognition model classifies the gesture. Recognized characters are processed by the sentence formation module and displayed as text. The resulting sentence can be converted into speech.

In the reverse mode, the microphone captures speech. A speech recognition engine converts the speech into text. The resulting text is then mapped to ASL alphabet images or phrase animations and displayed to the ASL user.

The system follows a three-layer architecture. The input and output layer interfaces with the physical environment. OpenCV captures webcam video at a minimum resolution of 640 × 480 pixels and processes captured frames. The processing and intelligence layer contains computer vision, feature engineering, machine learning, speech processing, and sentence formation components. The presentation layer provides a web-based Flask interface with real-time Socket.IO communication.

IV. METHODOLOGY

Dataset Preparation: Two datasets are used in the system. The first is the ASL Alphabet image dataset used for CNN training. It contains approximately 87,000 images across 29 classes, including 26 alphabet letters and SPACE, DELETE, and NOTHING gestures. The second is a custom landmark-based dataset containing 14,000 samples, with approximately 500 samples per letter class. MediaPipe extracts 21 landmarks from each hand and represents them as normalized 63-feature vectors. The data is divided into 70% training, 15% validation, and 15% testing subsets.

Image Preprocessing: For CNN training, images are resized to 64 × 64 pixels and pixel values are normalized between 0 and 1. Labels are one-hot encoded for categorical cross-entropy training. Data augmentation includes rotation, width and height shifts, brightness variation, horizontal flipping, Gaussian noise, and zoom variation.

MediaPipe Landmark Extraction: MediaPipe Hands detects 21 hand landmark points from the webcam stream. These landmarks are normalized to reduce sensitivity to differences in hand size, camera distance, and camera angle. The normalized coordinates form a 63-element feature vector used by the lightweight classifier.

CNN Model: A CNN is trained to classify ASL alphabet gestures from image data. Different network configurations, convolutional layers, filter sizes, dropout rates, and learning rates were experimentally evaluated. The final model was selected based on validation accuracy and inference speed. The CNN was trained for 30 epochs using the Adam optimizer, with the best-performing checkpoint retained for inference.

Random Forest Model: A Random Forest classifier is trained using normalized MediaPipe landmark features. This provides a lightweight alternative to CNN-based classification and is particularly useful for CPU-based real-time processing. Hyperparameter tuning was performed using five-fold cross-validation.

Sentence Formation: A sentence formation module maintains a character buffer and uses temporal smoothing to stabilize predictions. SPACE gestures are used for word separation, while sentence completion triggers the next stage of processing.

Speech Processing: The speech-to-text module uses the SpeechRecognition library with Google Web Speech API and Whisper offline model support. For text-to-speech conversion, pyttsx3 is used as the primary engine, with gTTS available as an alternative.

Web Dashboard: The Flask application provides the web dashboard, while Socket.IO enables real-time communication between the backend and frontend. HTML, CSS, and JavaScript are used to create the user interface.

V. IMPLEMENTATION

The implementation followed an iterative and incremental development methodology. Individual components were developed and tested before being integrated into the complete system. This approach allowed problems to be identified early and enabled continuous improvement of individual modules.

The implementation was divided into five

major phases:

1-problem analysis and planning.

2-dataset preparation and model training,

3-core pipeline development.

4-speech module and dashboard development.

5-integration testing and evaluation.

The core technology stack includes Python, OpenCV, MediaPipe, TensorFlow/Keras, Scikit-learn, NumPy, SpeechRecognition, pyttsx3, gTTS, Flask, and Socket.IO.

VI. RESULTS AND DISCUSSION

The system was evaluated using controlled indoor lighting, variable natural lighting, and low-light conditions. Three users with different hand sizes and signing styles participated in the testing.

The project report reports 97.3% recognition accuracy on the standard ASL Alphabet test dataset and approximately 91.6% under live real-world conditions. The real-world environmental evaluation reported an overall average of 90.2%, with 95.2% under controlled indoor lighting, 90.3% under variable natural lighting, and 85.3% under low-light conditions.

The Random Forest frame-to-dashboard pipeline recorded an average latency of 56 ms and a maximum latency of 119 ms. CNN classification averaged 71 ms, while MediaPipe landmark extraction averaged 18 ms. These measurements indicate that the Random Forest pipeline provides a lightweight processing path suitable for real-time interaction.

Functional testing covered individual letters, sequential alphabet recognition, SPACE gesture handling, speech recognition, webcam interruption, microphone disconnection, dashboard loading, mode switching, session-history storage, and continuous operation. The reported test cases passed successfully.

The results indicate that combining MediaPipe landmark extraction with machine learning can provide practical real-time ASL recognition using standard computing hardware. Recognition performance decreases under low-light conditions, demonstrating that environmental conditions remain an important factor affecting recognition performance. The bidirectional design additionally provides a speech/text-to-ASL path rather than limiting communication to ASL-to-text conversion.

VII. CONCLUSION

This paper presented SignFusion, a real-time bidirectional American Sign Language communication system based on computer vision, MediaPipe, machine learning, speech processing, and web technologies. The system supports ASL-to-text/speech and speech/text-to-ASL communication using a standard webcam, microphone, and computer.

MediaPipe is used to extract hand landmarks, while CNN and Random Forest models provide gesture classification capabilities. Sentence formation, speech recognition, text-to-speech, and ASL visual mapping extend the system beyond basic gesture classification. The Flask and Socket.IO dashboard provides a unified interface for real-time interaction.

The reported experimental results demonstrate 97.3% benchmark recognition accuracy, approximately 91.6% live recognition accuracy, and low-latency operation through the Random Forest pipeline. The system therefore demonstrates the feasibility of a software-only approach to real-time bidirectional ASL communication.

VIII. FUTURE ENHANCEMENT

Future development can extend the system in several directions. Short-term improvements include expanding the phrase vocabulary, supporting two-hand gestures, enabling custom gestures, improving low-light recognition, and optimizing the dashboard for mobile browsers.

Medium-term improvements include continuous sign-language recognition using LSTM or Transformer models, facial-expression analysis using MediaPipe Face Mesh, edge-device deployment, and support for additional sign languages such as Indian Sign Language and British Sign Language.

Long-term research could incorporate contextual language models for error correction, 3D animated avatars for text-to-ASL rendering, wearable sensors for hybrid recognition, and usability studies involving deaf and hard-of-hearing participants.

RESULTS TABLES

TABLE I. GESTURE RECOGNITION ACCURACY UNDER DIFFERENT CONDITIONS

Condition	User A	User B	User C	Average
Controlled Indoor Lighting	96.2%	94.8%	94.5%	95.2%
Variable Natural Lighting	91.0%	89.3%	90.7%	90.3%
Low-Light (80 lux)	85.7%	84.1%	86.1%	85.3%
Overall Average	90.9%	89.4%	90.4%	90.2%

TABLE II. SYSTEM LATENCY MEASUREMENTS

Processing Stage	Average	Maximum
Video frame capture	5 ms	12 ms
MediaPipe landmark extraction	18 ms	35 ms
Feature normalization	2 ms	4 ms
Random Forest classification	12 ms	20 ms
CNN classification	71 ms	95 ms
Sentence formation	1 ms	3 ms
Dashboard update (Socket.IO)	18 ms	45 ms
Text-to-speech (pyttsx3)	85 ms	140 ms
RF frame-to-dashboard pipeline	56 ms	119 ms

REFERENCES

[1] World Health Organization, "World Report on Hearing," WHO Press, Geneva, Switzerland, 2021.

[2] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. Chang, and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," in Proc. IEEE CVPR Workshops, Seattle, WA, USA, 2020.

[3] G. Fang, W. Gao, and D. Zhao, "Large vocabulary sign language recognition based on fuzzy decision trees," IEEE Trans. Syst., Man, Cybern. A, vol. 34, no. 3, pp. 305–314, May 2004.

[4] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in Proc. ECCV Workshop, Zurich, Switzerland, 2014.

[5] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti, "American sign language recognition with the Kinect," in Proc. ICMI, Alicante, Spain, 2011.

[6] H. Cooper and R. Bowden, "Large lexicon detection of sign language," in Proc. ICCV Workshop, 2007.

[7] L. Pigou, S. Dieleman, P.-J. Kindermans, and F. Schrauwen, "Sign language recognition using convolutional neural networks," in Proc. ECCV Workshop, Zurich, Switzerland, 2014.

[8] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3D convolutional neural networks," in Proc. IEEE ICME, Turin, Italy, 2015.

