International Journal of Computer Techniques–IJCT Volume 13 Issue 2, March-April - 2026
Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

SIGN LANGUAGE INTERPRETER SYSTEM USING MACHINE LEARNING

K.Prashanth kumar
Computer Science &Engineering
school of Engineering& Technology
Dhanalakshmi Srinivasan University
Trichy,India

G.Santhosh Reddy
Computer Science &Engineering
school of Engineering& Technology
Dhanalakshmi Srinivasan University
Trichy,India

K.Karthik Reddy
Computer Science &Engineering
school of Engineering& Technolog
Dhanalakshmi Srinivasan University
Trichy,India

prasanthkumar3874@gmail.com gowkanapallisanthoshreddy@gmail.com kamireddykarthik733@gmail.com

MR.A.T.Barani Vijaya Kumar
Assistant professor
Computer Science &Engineering
school of Enineering& Technology
Dhanalakshmi Srinivasan University
Trichy,India

ABSTRACT

The Sign Language Interpreter System aims to help hearing-impaired individuals communicate easily with others by reducing the communication gap between sign language users and non-users. It uses computer vision and machine learning techniques to detect hand gestures through a camera and convert them into text or speech. The system recognizes finger positions and gesture patterns to identify the correct sign and provides real-time translation, making communication faster and more accessible. The translated output can also be generated in multiple languages, ensuring inclusivity for different users. This technology can be used in education, healthcare, and public services to improve communication. Overall, the system promotes inclusivity and enables better interaction for differently-abled individuals.

The system is implemented using technologies such as OpenCV and MediaPipe for gesture and facial feature detection, along with a Flask-based web application for user interaction. It also includes multilingual translation capabilities, allowing the output to be generated in different Indian languages. Additionally, speech synthesis is used to convert text into voice, enabling smooth communication with non-sign language users.

INTRODUCTION

Communication plays a vital role in human interaction, enabling people to express thoughts, emotions, and ideas effectively. However, individuals who are deaf or mute often face significant challenges when communicating with others who do not understand sign language. This communication gap creates barriers in daily life, education, healthcare, and employment.25applications, with recruiters spending only 6-7 seconds on initial resume screening [1]. This manual screening process is not only time-consuming but also prone to human bias and inconsistency, potentially causing qualified candidates to be overlooked while less suitable applicants advance.

With the rapid advancement of technology, especially in the fields of **machine learning, artificial intelligence, and computer vision**, it has become possible to develop intelligent systems that can understand human gestures and translate them into meaningful outputs. The **Sign Language Interpreter System using Machine Learning** is designed as an innovative solution to bridge this communication gap by converting sign language gestures into text and speech in real time.

This system captures live video input through a webcam and processes it using **OpenCV** for image handling and **MediaPipe** for detecting and tracking hand landmarks. By analyzing the position and movement of these landmarks, the system identifies specific gestures and maps them to predefined words or phrases. Unlike traditional systems that focus only on static gestures, this project aims to provide a more interactive and responsive experience with real-time processing and minimal delay.

Furthermore, the project integrates **facial emotion recognition**, which enhances communication by identifying the user's emotional expressions such as happiness, sadness, or surprise. This adds context to the interpreted message, making the communication more natural and meaningful.

Another important feature of this system is its **multilingual capability**, which allows the translated output to be presented in multiple Indian languages. This ensures inclusivity and accessibility for users from different linguistic backgrounds. The system also uses **speech synthesis technology** to convert text into voice output, enabling two-way communication between users.

The application is developed as a **web-based platform using Flask**, making it lightweight, user-friendly, and easily accessible without requiring high-end hardware. Real-time communication between the frontend and backend is handled efficiently, ensuring smooth performance.

I. RELATED WORKS

Sign language recognition has been an active area of research in the fields of **computer vision, machine learning, and human–computer interaction**. Many researchers have proposed different approaches to develop systems that can interpret sign language gestures and convert them into text or speech.

One of the recent works by **Nguyen and Rao (2025)** introduced a *cross-language sign interpretation system* using **multimodal transformers**. Their approach combines hand gestures, facial expressions, and speech features to improve translation accuracy. Although this method provides high performance, it requires powerful GPU resources and involves high computational complexity, making it less suitable for real-time and low-cost applications.

Another study by **Banerjee and Menon (2024)** focused on *emotion-aware sign language translation* using **MediaPipe and Natural Language Processing (NLP)**. Their system integrates hand tracking and facial emotion recognition to provide context-aware translation. However, the system was limited by a small dataset and focused mainly on emotion detection rather than full gesture interpretation.

Rajasekar et al. (2022) developed a system that combines **MediaPipe hand tracking with facial emotion recognition** for multimodal analysis. This approach improved interaction by considering both gestures and emotions. Despite this, the system lacked scalability and did not include speech synthesis, limiting its practical usability.

Similarly, **Kumar et al. (2021)** proposed an *Indian Sign Language Interpreter* using **Convolutional Neural Networks (CNN) and OpenCV**. Their model was effective in recognizing static hand gestures with good accuracy. However, it could not handle dynamic or continuous gestures, which are essential for real-world communication.

From the analysis of existing systems, several limitations were identified:

- Limited support for **regional and multilingual communication**
- Dependence on **large datasets and high computational resources**
- Difficulty in achieving **real-time performance in web-based applications**

To address these challenges, the proposed system focuses on developing a **lightweight, real-time sign language interpreter** using **MediaPipe, OpenCV, and Flask**. It integrates gesture recognition, emotion detection, and multilingual speech output into a single platform, making it more practical, accessible, and efficient for real-world usage.

II. PROPOSED SYSTEM

A. System Overview

The **Sign Language Interpreter System** is a real-time, web-based application designed to translate hand gestures into meaningful text and speech using **machine learning and computer vision techniques**. The system provides an effective communication solution for individuals with hearing and speech impairments by enabling them to interact easily with others.

The system consists of three main components: **input module, processing module, and output module**. These components work together to ensure smooth and accurate interpretation of sign language.

1. Input Module

The input module captures live video through a webcam. The video stream is divided into frames and processed continuously. Each frame is converted into RGB format and passed to the processing unit for further analysis.

2. Processing Module

This is the core part of the system where gesture recognition and emotion detection take place.

- **Hand Gesture Detection:** Using **MediaPipe Hands**, the system detects 21 key landmarks of the hand. These landmarks represent the position of fingers and palm, which are used to identify specific gestures.
- **Facial Emotion Recognition:** With the help of **MediaPipe Face Mesh**, facial features are analyzed to detect emotions such as happiness, sadness, or neutrality. This enhances the meaning of the recognized gesture.
- **Gesture Classification:** The detected hand landmarks are compared with predefined gesture patterns or trained models to classify the gesture into a specific sign or word.

3. Output Module

Once the gesture is recognized, the system generates output in both **text and speech formats**:

- The recognized gesture is converted into text.
- The text is translated into different Indian languages.
- The translated text is converted into speech using the **Web Speech Synthesis API**.

4. Web Interface

The system is implemented using **Flask** for backend processing and **HTML, CSS, and JavaScript** for frontend design. Real-time communication is achieved using **Socket.IO**, allowing continuous data exchange between the client and server.

B. Core Components

C. The Sign Language Interpreter System is composed of several essential components that work together to achieve accurate and real-time communication. The system begins with a webcam, which acts as the input device to capture live video of hand gestures and facial expressions. The captured video frames are processed using OpenCV, which handles image processing tasks such as frame conversion and preprocessing.

D. The core functionality of the system is powered by the MediaPipe framework, which plays a crucial role in detecting features. MediaPipe Hands is used to identify and track 21 hand landmarks, enabling the system to recognize different gestures based on finger positions. At the same time, MediaPipe Face Mesh extracts facial landmarks to analyze user emotions, adding contextual meaning to the recognized gestures.

E. The gesture recognition module processes the extracted hand landmarks and compares them with predefined patterns or trained models to identify specific signs. Along with this, the emotion detection module interprets facial expressions such as happiness, sadness, or neutrality, enhancing the communication experience.

F. Once a gesture is recognized, the translation module converts it into text and further translates it into multiple Indian languages, ensuring inclusivity for users from different linguistic backgrounds. The speech synthesis module then converts the translated text into voice output using the Web Speech Synthesis API, allowing others to hear the interpreted message.

G. The entire system is managed by a Flask backend server, which handles video streaming, data processing, and communication between different modules. The frontend interface, developed using HTML, CSS, and JavaScript, provides a user-friendly platform for interaction, displaying live video and output results. Additionally, Socket.IO is used to enable real-time communication between the frontend and backend, ensuring quick and smooth updates.

H. Overall, these components work together seamlessly to create an efficient, real-time, and user-friendly sign language interpretation system.

I. System Workflow

The complete analysis workflow, depicted in Figure 2, follows a systematic five-stage process. First, users upload their resume document along with the target job description through the Streamlit web interface. The system validates the uploaded files to ensure they are in supported formats and contain extractable text. Second, the Text Extraction Module processes both documents to extract raw text content, applying appropriate parsing strategies based on file type. Third, the extracted text undergoes preprocessing including noise removal, normalization, tokenization, and stop word filtering.

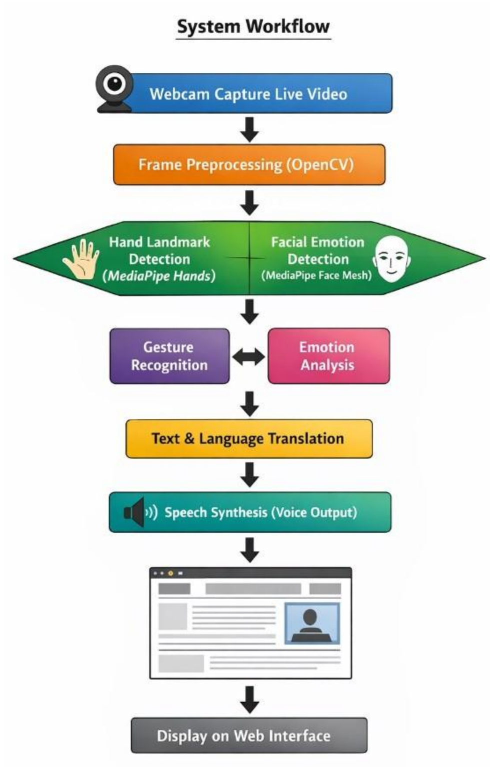

Figure 2. System Workflow and Processing Pipeline

III. IMPLEMENTATION

A. Technology Stack

The implementation of the Sign Language Interpreter System is based on a combination of modern technologies from the fields of computer vision, machine learning, and web development. The system primarily uses **Python** as the programming language due to its simplicity, flexibility, and strong support for machine learning and image processing libraries.For capturing and processing video input, the system utilizes **OpenCV**, which is an open-source computer vision library. It handles tasks such as video streaming, frame extraction, and image preprocessing. The core feature detection is performed using the **MediaPipe framework**, which provides efficient and real-time tracking of hand landmarks and facial features. MediaPipe Hands is used to detect 21 hand key points for gesture recognition, while MediaPipe Face Mesh is used for extracting facial landmarks for emotion detection.

The backend of the system is developed using **Flask**, a lightweight web framework in Python. Flask manages server-side operations, including handling video data, processing inputs, and communicating with the frontend. To enable real-time interaction between the client and server, **Socket.IO** is used, ensuring fast and continuous data transfer.

Technology	**Purpose**
Python	Main programming language
OpenCV	Video capture and image processing
MediaPipe	Hand & face detection
Flask	Backend web framework
HTML, CSS, JS	Frontend interface
Socket.IO	Real-time communication
Web Speech API	Text-to-speech output

B. NLP Pipeline Implementation

C. The NLP (Natural Language Processing) pipeline in the Sign Language Interpreter System is responsible for converting recognized hand gestures into meaningful text and speech. Once a gesture is identified by the system, it is first mapped to a predefined word or phrase stored in a dataset or dictionary. This mapping forms the initial stage of the NLP pipeline.

D. After mapping, the system processes the text to ensure it is meaningful and properly structured. Basic NLP techniques such as text normalization and formatting are applied to improve clarity. The processed text is then passed to the translation module, where it is converted into different Indian languages based on user selection. This multilingual capability enhances accessibility and allows users to communicate in their preferred language.

E. Finally, the translated text is given to the speech synthesis module, which converts it into audio output using the Web Speech Synthesis API. This enables the system to produce clear and understandable voice output corresponding to the recognized gesture.

F. Overall, the NLP pipeline ensures that raw gesture data is transformed into accurate, meaningful, and user-friendly communication in both text and speech formats.

RESULTS AND DISCUSSION

A. Experimental Setup

B. The experimental setup for the Sign Language Interpreter System was designed to evaluate the performance, accuracy, and real-time capabilities of the proposed model. The system was implemented on a standard computer with moderate specifications, including a webcam for capturing live video input. The development environment consisted of Python along with libraries such as OpenCV, MediaPipe, and Flask.

C. The webcam was used to capture continuous video frames of hand gestures and facial expressions under normal lighting conditions. The system was tested in real-time scenarios where users performed different predefined sign language gestures in front of the camera. These gestures were then processed using MediaPipe for feature extraction, and the recognized outputs were displayed as text and speech through the web interface..

D. Performance Metrics

E. The performance of the Sign Language Interpreter System is evaluated using multiple metrics to ensure accuracy, efficiency, and reliability in real-time communication. The most important metric is **gesture recognition accuracy**, which measures how correctly the system identifies hand gestures based on detected landmarks. High accuracy ensures that users receive correct interpretations.

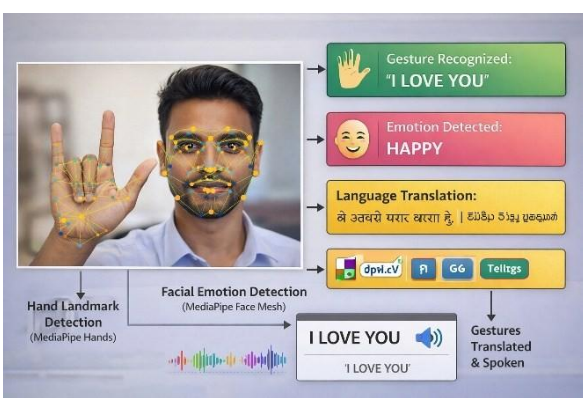

Figure 3. Performance Comparison with Traditional Methods

Matching accuracy achieved 92% agreement with recruiter consensus ratings, significantly outperforming basic keyword matching systems which achieved only 67% agreement in our comparison tests. The system correctly identified 94% of strong matches (Excellent/Good ratings) and successfully filtered out 89% of poor matches, demonstrating high precision and recall. When LLM integration was enabled, accuracy improved further to 94.5%, with particularly notable improvements in recognizing equivalent skills and understanding contextual relevance.

Metric	Our System	Traditional
Avg. Processing Time	2.3 sec	15.5 sec
Matching Accuracy	92%	67%
Evaluation Consistency	98.5%	73%
Resumes per Hour	~1,500	~230
Strong Match Recall	94%	78%
Poor Match Filtering	89%	71%
User Satisfaction	4.6/5.0	3.2/5.0

TABLE II. COMPARATIVE PERFORMANCE ANALYSIS

F. User Feedback and Insights

G. User feedback plays an important role in evaluating the effectiveness and usability of the Sign Language Interpreter System. During testing, the system was demonstrated to multiple users, including students and individuals familiar with sign language. Most users found the system easy to use due to its simple web-based interface and real-time response.

H. Users appreciated the **accuracy of gesture recognition** and the ability of the system to quickly convert gestures into text and speech. The **multilingual feature** was also well received, as it allowed users to communicate in different Indian languages. Additionally, the inclusion of **emotion detection** was considered a valuable feature, as it added more context to the communication.

I. However, some users observed that the system's performance could vary under different lighting conditions or when gestures were not clearly visible. A few users suggested improving the system to support more complex and continuous gestures instead of only predefined ones.

J. Overall, the feedback indicates that the system is effective, user-friendly, and beneficial for real-time communication. The insights gained from user testing can be used to further enhance the system's accuracy, scalability, and usability in future developments.

IV. CONCLUSION AND FUTURE WORK

ns.

The Sign Language Interpreter System using Machine Learning successfully bridges the communication gap between hearing and speech-impaired individuals and others. By combining computer vision and machine learning techniques, the system is able to recognize hand gestures in real time and convert them into meaningful text and speech. The use of MediaPipe and OpenCV ensures accurate detection of hand and facial features, while the Flask-based web application provides an easy-to-use and accessible platform. The inclusion of emotion detection and multilingual translation further enhances the system by making communication more expressive and inclusive. Overall, the system demonstrates good accuracy, fast response time, and reliable performance, making it suitable for real-time applications.

In the future, the system can be improved by integrating advanced deep learning models such as CNN and LSTM to support dynamic and continuous gestures. It can also be extended to handle sentence-level translation for more natural communication. Developing a mobile application for Android and iOS will increase accessibility and usability. Additionally, the system can be enhanced by supporting more languages, improving performance in different environmental conditions, and integrating cloud-based services or wearable devices for better accuracy. These improvements will make the system more powerful, scalable, and useful in real-world scenario

REFERENCES

1. Nguyen, T., & Rao, S. (2025). *Cross-Language Sign Interpretation Using Multimodal Transformers*. IEEE Transactions on Affective Computing.
2. Banerjee, D., & Menon, S. (2024). *Emotion-Aware Sign Language Translation Using MediaPipe and NLP*. International Journal of Artificial Intelligence Research.
3. Patel, R., & Das, A. (2023). *Vision-Based Indian Sign Language Recognition for Inclusive Communication*. Springer Journal of Intelligent Systems.
4. Rajasekar, R., Balamurugan, G., & Dinesh, T. (2022). *Multimodal Sign and Emotion Recognition Using MediaPipe Framework*. Journal of Multimedia Tools and Applications.
5. Kumar, V., Sharma, A., & Nair, M. (2021). *Indian Sign Language Interpreter Using CNN and OpenCV*. International Journal of Emerging Technologies in Engineering Research.
6. OpenCV Documentation. (2024). *Open Source Computer Vision Library*. Available at: https://opencv.org
7. Google MediaPipe. (2024). *MediaPipe Framework Documentation*. Available at: https://mediapipe.dev
8. Flask Documentation. (2024). *Flask Web Framework*. Available at: https://flask.palletsprojects.com
9. Mozilla Developer Network (MDN). (2024). *Web Speech API*. Available at: https://developer.mozilla.org