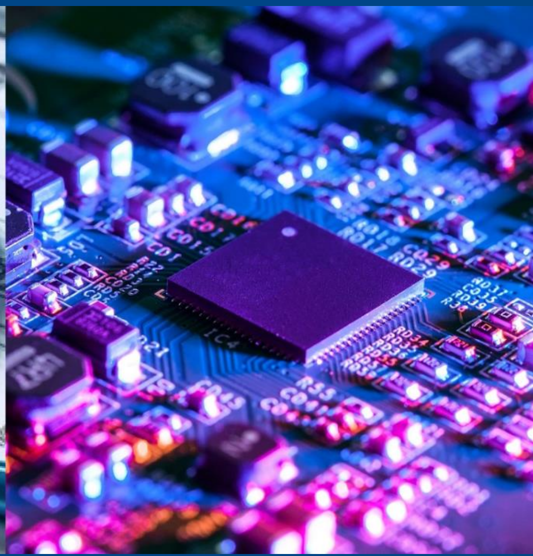
ISSN: 2582-7219

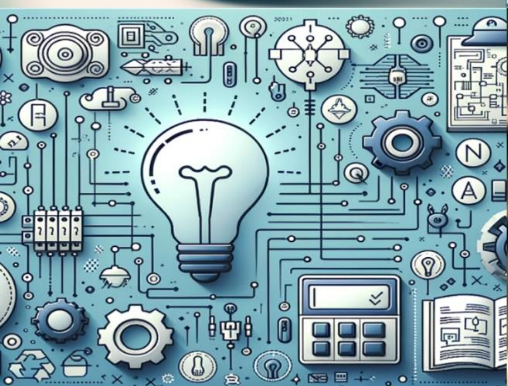




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 4, April 2025

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Virtual Sign Language Translator

Pathan M.N[1], Shaikh M.M[2], Ghodake P.S[3] Subhedar A.J[4]

Department of Computer Engineering, SVSMD's KKI Polytechnic College Akkalkot, India

Gaikwad S.T., Chougule V.V

Guide, Department of Computer Engineering, SVSMD's KKI Polytechnic College Akkalkot, India

ABSTRACT: The development of a Virtual Sign Language Translator represents a significant advancement in facilitating communication between hearing and non-hearing individuals, particularly in educational and business settings. This initiative aims to bridge the communication gap by providing real-time translation of sign language into text or speech, enhancing accessibility and inclusivity. The system utilizes advanced technologies such as computer vision, machine learning, and natural language processing to accurately interpret sign language gestures. Users can interact through intuitive interfaces, enabling seamless conversion of sign language into readable or audible formats. By integrating this translator into classrooms, universities, institutions, and business environments, we can promote equal participation and understanding, ensuring that information is accessible to all, regardless of hearing ability.

KEYWORDS - : HTML, CSS, JavaScript, Python.

I. INTRODUCTION

Sign language is an essential means of communication for millions of people worldwide, particularly those who are deaf or hard of hearing. However, the lack of widespread understanding of sign language among the general population creates significant communication barriers. These challenges highlight the importance of developing assistive technologies that can bridge the gap between sign language users and non-signers. With recent advancements in artificial intelligence and computer vision, it is now possible to create automated systems that can recognize and translate sign language gestures into text or speech in real-time.

Traditional methods of sign language recognition often involve human interpreters, sensor-based gloves, or specialized cameras, all of which have limitations. Human interpreters are not always readily available, making communication difficult in certain situations. Sensor-based gloves and specialized cameras, while effective, can be costly and impractical for everyday use. These challenges have driven researchers and engineers to explore AI-powered solutions that rely on standard camera inputs and machine learning models for sign language detection.

This system utilizes MediaPipe, a framework developed by Google that offers solutions for hand tracking and gesture recognition, and integrates it with a machine learning model trained to classify hand gestures. The video feed from a webcam is captured, processed, and analyzed for hand landmarks, and the gestures are predicted and translated into readable text. This is achieved by utilizing Flask, which serves as the backend web framework for serving the web interface and providing real-time video streaming capabilities.

How It Works:

Hand Gesture Detection: The core of the system relies on **MediaPipe's Hand module**, which tracks hand landmarks using a webcam feed. The hand landmarks represent specific points on the hand such as the wrist, fingers, and joints. These landmarks are captured in real-time, and based on their positions and relative distances, the system extracts features that are used for classification.

Machine Learning Model: A pre-trained machine learning model (stored in model.p) is used to classify the extracted features into one of the 26 letters (A-Z) and numbers (0-9) in ASL. The model uses the coordinates of the hand landmarks, as well as additional features like the distance between the wrist and index finger, to make predictions.

Video Stream: The system captures the webcam feed using **OpenCV** and sends it to the browser in real-time using Flask's Response object. The processed frames are streamed as a video, with detected gestures displayed along with the corresponding predicted letter or number.

© 2025 IJMRSET | Volume 8, Issue 4, April 2025 |DOI:10.15680/IJMRSET.2025.0804350

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

User Interaction: Users can start or stop the video feed, and the detected text is dynamically updated based on hand gestures. The predicted text is continuously displayed as a string, which simulates an ongoing translation of sign language.

System Components:

Flask: The web framework that serves the frontend and backend, handling routes like /video_feed, /get_detected_text, and /start_video.

MediaPipe: A computer vision library used for hand gesture detection. The Hand module provides real-time tracking of hand landmarks and provides the foundation for feature extraction.

OpenCV: A library used for video capture and image processing. It handles the webcam feed and is responsible for converting the frames into the appropriate format for display in the web interface.

Machine Learning Model: A pre-trained classifier that predicts the ASL letter or number based on the detected hand landmarks. This model is loaded from a pickle file.

HTML & JavaScript: The frontend displays the video feed and detected text. It also handles user interactions, like starting and stopping the video feed.

This Virtual Sign Language Translator is a significant step toward improving communication accessibility for people with hearing impairments. The combination of real-time hand gesture recognition, machine learning, and web technologies provides an interactive and dynamic way to translate sign language into text, making communication more inclusive and easier. Further advancements can expand the system's capabilities and enhance its usability in real-world applications.

II. FEATURES AND FUNCTIONALITY

Real-Time Translation: Users can perform ASL gestures, and the system translates them into text in real-time.

Start/Stop Video Feed: The video feed can be started or stopped with a simple route, allowing users to control the video stream.

Text Display: The detected sign language is displayed continuously as text on the webpage, which updates as new gestures are recognized.

Interactive Web Interface: The interface is simple and easy to use, with multiple pages for learning more about the project, how it works, and viewing the translated text.

Challenges and Future Work:

Gesture Recognition Accuracy: While the system works well for isolated gestures, continuous signing (such as sentences) may present challenges in context and accuracy. Future work could focus on improving the model's ability to handle sequences of gestures.

Training Data: The model relies on the quality and quantity of training data. Expanding the dataset to include more vriations of ASL could improve the model's robustness.

Mobile and Hardware Integration: The system could be further improved by optimizing it for mobile devices and integrating it with hardware solutions like smart glasses or AR devices for a more immersive experience

III. EXISTING SYSTEM:

Several existing systems have successfully combined machine learning, computer vision, and web technologies to create virtual sign language translators. These systems aim to bridge communication gaps for individuals with hearing impairments by translating sign language gestures into text or speech. Here are some notable examples:

BISINDO Sign Language Translator Web App

This web application employs object detection with YOLOv8, MediaPipe, and Flask to translate Indonesian Sign Language (BISINDO) gestures into text. It captures hand gestures using MediaPipe, processes them through a Convolutional Neural Network (CNN) for classification, and displays the translated text on the web interface. GitHub

Sign Language Recognition System with Flask and MediaPipe

This system uses computer vision to detect and interpret hand gestures, converting them into text and spoken words. It employs MediaPipe for hand landmark detection and integrates with Flask to provide a web-based interface for real-time gesture recognition. GitHub

Sign Language Detection Using MediaPipe and Machine Learning

This project focuses on simplifying sign language recognition by utilizing MediaPipe's open-source framework and machine learning algorithms. It processes webcam video to detect hand landmarks, classifies gestures using a Random Forest model, and integrates with Flask for user authentication and real-time predictions. IJRAR

- **Real Time Examples:**
- sign.mt
- SignSync AI
- SignAll
- slait.ai
- Signapse AI
- DeepSign
- SimulSLT
- SIGNEASE
- Sign-Language-Translator
- sign/translate

Need of this project:

This project is important because it helps deaf and hard-of-hearing people communicate easily with others by translating sign language in real-time. It makes everyday interactions, like in schools, hospitals, and workplaces, more accessible for them. The system also helps reduce isolation by allowing deaf individuals to communicate with non-signing people. It can be especially helpful in places where sign language interpreters are not available. Overall, this project ensures that deaf individuals can take part in society and have the same opportunities as everyone else.

1. Bridging Communication Gaps
2. Enhancing Accessibility
3. Promoting Social Inclusion
4. Supporting Independence
5. Addressing Interpreter Shortages
6. Facilitating Equal Opportunities
7. Improving Emergency Communication

IV. PROPOSED SYSTEM:

The proposed system is a real-time sign language translator that aims to bridge communication gaps between deaf or hard-of-hearing individuals and others who may not understand sign language. This system integrates advanced technologies such as computer vision, machine learning, and web frameworks to translate American Sign Language (ASL) into readable text or audible speech. The core of the system relies on **MediaPipe**, a framework developed by Google, which facilitates real-time detection and tracking of hand gestures. By capturing video from a webcam, the system can process the hand landmarks and gestures in real-time, enabling immediate translation.

Once the gestures are detected, the system utilizes machine learning models, particularly

© 2025 IJMRSET | Volume 8, Issue 4, April 2025 |DOI:10.15680/IJMRSET.2025.0804350

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

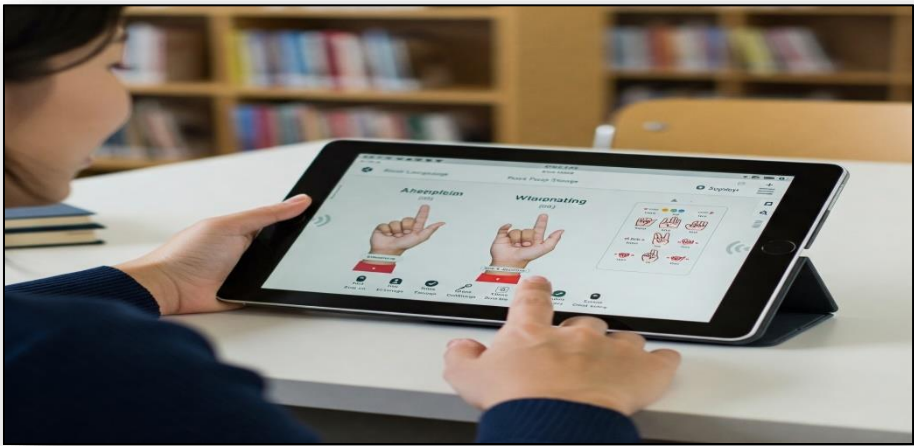
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Convolutional Neural Networks (CNNs), to classify the gestures and map them to the corresponding ASL signs. The machine learning model will be trained on a large dataset of ASL signs to ensure accurate recognition of gestures. This training is crucial to handle the diversity of hand shapes and motions involved in sign language. After identifying the gesture, the system translates it into text, which is then displayed on a web interface. **Text-to-Speech (TTS)** technology can be integrated to read out the translated text, ensuring that the system is accessible not only for sign language users but also for those who prefer auditory communication.

The system will be hosted through a **Flask**-based web interface, which will manage the video stream and display the translated text in real-time. Flask will allow for easy deployment of the application on any device with a web browser, making it highly accessible. The interface will be simple and user-friendly, catering to a wide range of users, from those familiar with technology to those who are less tech-savvy. The system's primary objective is to enhance accessibility, inclusivity, and communication between deaf individuals and the broader community.

In the future, the system can be expanded to support multiple sign languages, allowing users from different regions or cultural backgrounds to benefit from it. A mobile app version of the system could also be developed to increase accessibility, especially in situations where a webcam may not be available. Furthermore, the system could be enhanced with continuous learning capabilities, allowing it to adapt and improve over time based on user feedback and new sign language gestures.

Overall, the proposed real-time sign language translator will make significant strides toward inclusivity and accessibility. It will empower deaf and hard-of-hearing individuals by breaking down communication barriers and promoting social participation, independence, and equal access to information.



V. METHODOLOGY

Methodology for Virtual Sign Language Translator in Flask Web App

The methodology for developing the **Virtual Sign Language Translator** using Flask involves several stages, from data collection and model training to web development and integration. The goal is to create a functional, real-time sign language translator that accurately converts American Sign Language (ASL) gestures into text, leveraging machine learning, computer vision, and web technologies. Below is a step-by-step breakdown of the methodology:

Data Collection and Preprocessing

Objective: Gather and prepare a dataset of American Sign Language (ASL) gestures for training the machine learning model.

ASL Dataset: The first step is to gather a large and diverse dataset of ASL signs. There are publicly available datasets such as the **ASL Alphabet Dataset** or **American Sign Language Dataset** that contain images of different ASL signs.

Preprocessing: The collected images need to be preprocessed for model training:

- ❖ Resize images to a fixed resolution (e.g., 64x64 or 128x128 pixels).
- ❖ Normalize pixel values (scaling pixel values to a range of 0-1).
- ❖ Data augmentation (rotations, flips, and scaling) to improve the model's generalization capabilities.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- **Hand Landmark Extraction**: For real-time gesture recognition, use **MediaPipe** to extract hand landmarks from video input. These landmarks will help in detecting and understanding hand positions and movements in ASL gestures.

Model Development and Training
Objective: Develop and train a machine learning model to recognize ASL gestures.
Model Selection:
Use a **Convolutional Neural Network (CNN)**, a popular choice for image-based tasks. CNNs are well-suited for recognizing spatial patterns like hand shapes in images.
Alternatively, for more advanced and sequential gesture recognition, consider using **Recurrent Neural Networks (RNNs)** with **Long Short-Term Memory (LSTM)** layers for recognizing dynamic hand movements.
Training:
Split the dataset into training, validation, and test sets (e.g., 70% training, 20% validation, 10% testing).
Train the model on a GPU (if available) for faster training. Use **TensorFlow** or **Keras** to build the model.
Monitor the model's performance using metrics like accuracy and loss to prevent overfitting and ensure proper learning.
If needed, fine-tune the model by adjusting hyperparameters like the number of layers, learning rate, and batch size.

Model Evaluation:
Evaluate the model on the test dataset to check its accuracy in recognizing ASL gestures.
Implement methods like **confusion matrix** or **cross-validation** to measure performance in real-world conditions.
Hand Gesture Recognition in Real-Time
Objective: Enable the system to detect and recognize ASL gestures from a live video stream.
Video Stream Capture:
Use **OpenCV** to capture real-time video from the webcam. This will provide the live input for gesture recognition.
MediaPipe Hand Tracking:
MediaPipe will be used to detect hand landmarks and gestures in the video stream. It provides real-time hand tracking by identifying 21 key points on each hand, such as fingertips and palm positions.
Once the landmarks are detected, the system will extract and preprocess the data to feed into the trained machine learning model for gesture classification.

Gesture Classification:
For each detected hand gesture, the system will pass the landmark data (coordinates) into the trained CNN model to predict the corresponding ASL gesture.
The output of the model will be the predicted ASL sign, which will be translated into text.
Text Translation and Display
Objective: Display the translated ASL gesture as readable text on the web interface.
Text Output:
The recognized ASL gesture will be translated into its corresponding word or phrase in text form.
Text-to-Speech (Optional):
To make the system more accessible, integrate **Text-to-Speech (TTS)** technology using libraries like **pyttsx3** or APIs like **Google Text-to-Speech**. This will allow the system to read the translated text aloud, benefiting individuals with varying levels of literacy.
Web Development with Flask
Objective: Create the web interface to display the translation results and allow real-time interaction.
Flask Web App Setup:
Use **Flask**, a lightweight Python web framework, to create the web interface. Flask is ideal for rapid development and can handle both the backend logic and frontend display.
Set up routes to handle video stream input, gesture processing, and text output.

Real-Time Video Streaming:
Use **Flask-SocketIO** or **AJAX** to stream the video feed from the webcam to the frontend in real-time. This will ensure a smooth user experience, where the webcam input can be processed instantly.
Frontend Display:
Use **HTML**, **CSS**, and **JavaScript** to design the user interface.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Display the translated ASL signs in a simple, clear, and user-friendly format. The text will be dynamically updated as gestures are recognized.

Real-Time Interaction:

Integrate real-time feedback into the web interface, so users can see the translated ASL gestures instantly as they perform them. The system will continuously process the video stream and update the translation on the UI.

Testing and Debugging

Objective: Ensure the system works effectively and reliably across different scenarios.

Unit Testing:

Test individual components of the system, such as video capture, gesture recognition, and text output, to ensure they function as expected.

End-to-End Testing:

Conduct real-world testing by having users perform ASL gestures to verify the accuracy and speed of translation. Make adjustments to the model and system based on feedback.

Performance Optimization:

Optimize the system for low latency, ensuring the system processes gestures quickly enough for fluid communication.

Deployment

Objective: Deploy the web app so it is accessible to users online.

Web Hosting:

Host the Flask web application on a cloud platform like **Heroku**, **AWS**, or **Google Cloud Platform** to make it publicly accessible.

Cross-Device Compatibility:

Ensure the web app is compatible with various devices, including desktops, laptops, and mobile phones, by making the frontend responsive using **Bootstrap** or **CSS Grid**.

Future Enhancements

Multilingual Support: Add support for different sign languages or incorporate automatic translation between sign language and multiple spoken languages.

Mobile App: Develop a mobile app version of the system for greater accessibility on smartphones and tablets.

Machine Learning Enhancements: Continuously improve the model by retraining with more diverse datasets to increase recognition accuracy and add more signs.

VI. CONCLUSION

The **Virtual Sign Language Translator** project provides a promising solution for real-time translation of American Sign Language (ASL) into text using machine learning, computer vision, and web technologies. By integrating **MediaPipe** for gesture detection, a machine learning model for gesture classification, and **Flask** for the web interface, the system allows for seamless communication between deaf or hard-of-hearing individuals and those who do not know sign language.

Future work:

The system can be significantly enhanced by incorporating several key features. First, expanding support to multiple sign languages, such as British Sign Language or Spanish Sign Language, would make the translator accessible to a broader global audience. Developing a mobile app version would allow users to communicate on the go, increasing accessibility and convenience. Additionally, improving gesture recognition through training with a wider range of signs and diverse data would enhance accuracy and adaptability to various signing styles. Integrating real-time speech recognition would further elevate the system's versatility, enabling two-way communication between spoken and signed languages. Offering personalization options would allow users to teach the system their unique signs or regional variations, increasing flexibility. Finally, implementing continuous learning would enable the system to evolve over time by adapting to new gestures or feedback, improving its performance and relevance with regular use.

- **Support for Multiple Sign Languages**:

The system could be expanded to support different sign languages, like British Sign Language or Spanish Sign Language. This would make the translator useful to a wider range of people around the world.

- **Mobile App Development**:

Creating a mobile version of the system would allow users to use it on smartphones and tablets. This would make it more accessible for people to communicate on the go.

- **Improving Gesture Recognition**:

To make the system more accurate, it can be trained with more gestures and more varied data. This would help it recognize signs better and work well with different users' signing styles.

REFERENCES

1. https://www.signapse.ai/
2. https://www.handtalk.me/en/blog/meet-the-hand-talk-sign-language-translator-app/
3. https://ieeexplore.ieee.org/document/8951663
4. Marella, B.C.C., & Kodi, D. (2025). Fraud Resilience: Innovating Enterprise Models for Risk Mitigation. Journal of Information Systems Engineering and Management, 10(12s), 683–695
5. https://github.com/sign-language-translator/sign-language-translator
6. https://en.wikipedia.org/wiki/Machine_translation_of_sign_languages
7. https://www.mdpi.com/2227-7080/11/4/83
8. https://dl.acm.org/doi/10.1145/3448823.3448830
9. https://www.researchgate.net/publication/261460857_Vision-based_sign_language_translation_device
10. https://www.signfordeaf.com/about-us

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com