An Intelligent Computer Vision Framework for Hand Gesture Recognition in Sign Language

P.Suresh
Computer Science & Engineering
School of Engineering & Technology
Dhanalakshmi Srinivasan University
Trichy,India
sn320398@gmail.com

P.Prathap
Computer Science & Engineering
School of Engineering & Technology
Dhanalakshmi Srinivasan University
Trichy,India
ponnagantiprathap@gmail.com

P.Rohith
Computer Science & Engineering
School of Engineering &Technology
Dhanalakshmi Srinivasan University
Trichy,India
pokurirohith40054005@gmail.com

Mr.P.Jaya Chandran
Associate Professor
Computer Science & Engineering
School of Engineering & Technology
Dhanalakshmi Srinivasan University
Trichy,India

Email:jayachandran.set@dsuniversity.ac.in

***Abstract*—** **Sign language serves as a vital communication medium for individuals with hearing and speech impairments; however, limited awareness among the general population often leads to communication barriers and social exclusion. This project presents an intelligent computer vision–based framework for real-time hand gesture recognition .in sign language. The proposed system employs a vision-based approach using a standard webcam to capture hand gestures and utilizes MediaPipe for accurate hand landmark detection and feature extraction. These landmarks are processed to generate skeleton-based representations that minimize the impact of background variations and lighting conditions. Support Vector Machine is trained to classify sign language gestures, focusing on alphabetlevel recognition. To improve classification accuracy, gestures are strategically grouped into multiple classes and further distinguished using geometric relationships between landmarks. The system is implemented with an interactive graphical user interface, enabling real-time gesture recognition with high accuracy and robustness. This framework demonstrates an efficient, cost-effective, and userfriendly solution for automated sign language recognition, contributing to more inclusive human–computer interaction and improved accessibility for the hearing- and speechimpaired community.**

***Keywords*—** *: Sign Language Recognition , Hand Gesture Recognition , Computer Vision, Real-Time Gesture Classification , Support Vector machine*

I INTRODUCTION :

Sign language is one of the most powerful and expressive forms of communication for individuals with hearing and speech impairments. It allows people to convey thoughts, emotions, and ideas through structured hand gestures, facial expressions, and body movements. However, despite its importance, a large section of society is still unfamiliar with sign language. This lack of awareness often creates communication gaps, limits opportunities, and contributes to social isolation for members of the deaf and hard-of-hearing community.

With the rapid advancement of computer vision and artificial intelligence, technology now offers promising solutions to bridge this gap. In recent years, vision-based gesture recognition systems have gained significant attention because they provide a natural and contactless way to interpret human gestures. Unlike sensor-based approaches that require gloves or wearable devices, webcam-based systems are more practical, affordable, and accessible to everyday users.

This project focuses on developing an intelligent real-time sign language recognition system using computer vision techniques. The system captures hand gestures through a standard webcam and processes them using the powerful hand-tracking framework provided by MediaPipe. By extracting precise hand landmarks, the system converts raw visual input into structured skeletal representations. This varying lighting conditions, and camera noise, making the recognition process more stable and reliable.

To accurately classify gestures, Support Vector machine is trained on alphabet-level sign language data. The model learns distinctive spatial patterns from hand landmark representations rather than relying solely on raw images.

Additionally, grouping similar gestures and analyzing geometric relationships between landmarks further improves classification accuracy. This structured learning strategy enhances the system's ability to differentiate between visually similar signs.

An interactive graphical user interface (GUI) integrates all components into a user-friendly platform, enabling real-time prediction and display of recognized gestures. The overall framework is designed to be efficient, cost-effective, and practical for real-world applications.

By combining computer vision, machine learning, and intuitive interface design, this project aims to contribute toward more inclusive human–computer interaction.

II. LITERATURE SURVEY

This section reviews the research and technological developments related to sign language recognition systems and gesture-based human–computer interaction. It explores different approaches used in gesture detection, including sensor-based methods, vision-based systems, machine learning models, and natural language processing techniques used for improving communication. The discussion also highlights the limitations of existing solutions and establishes the motivation for developing an intelligent computer vision framework for sign language recognition.

2.1 Sensor-Based Sign Language Recognition Systems

Early research in sign language recognition relied heavily on sensor-based technologies, where users were required to wear special gloves embedded with sensors. These gloves were designed to capture finger movements, hand orientation, and motion patterns. The collected sensor data was then processed using algorithms to recognize the corresponding gestures.Although these systems offered relatively accurate tracking of hand movements, they introduced several practical limitations. The hardware equipment used in such systems was often expensive, bulky, and uncomfortable for users. Because of these drawbacks, sensor-based solutions have not gained widespread adoption outside research environments.

2.2 Vision-Based Gesture Recognition Systems

With advancements in computer vision technologies, researchers began exploring camera-based approaches for sign language recognition. In these systems, cameras capture images or video frames of hand gestures, which are then analyzed using image processing techniques. Vision-based systems eliminate the need for wearable hardware and provide a more convenient and user-friendly solution.Several computer vision frameworks have been introduced to detect hand structures and extract gesture-related features. However, many vision-based systems still face challenges related to lighting conditions, background noise, camera angles, and variations in hand size or skin tone, which can affect system performance in real-world environments.

2.3 Machine Learning Approaches for Gesture Classification

Machine learning algorithms have been widely used for classifying sign language gestures based on extracted features. Algorithms such as Support Vector Machines (SVM), Random Forest, K-Nearest Neighbors (KNN), and Neural Networks have shown promising results in gesture recognition tasks. These models are trained using datasets containing images or extracted gesture features representing different sign language alphabets. However, many existing machine learning systems are limited to recognizing a small set of predefined gestures, and their performance often depends on the quality and diversity of the training dataset.

2.4 Natural Language Processing in Sign Language Systems

While gesture recognition converts hand movements into text, the generated text may sometimes contain spelling mistakes or incomplete sentence structures. To improve the quality of communication, modern systems integrate Natural Language Processing (NLP) techniques. NLP modules can perform tasks such as spell correction, grammar refinement, and sentence enhancement. Despite these improvements, many existing systems focus primarily on gesture recognition and do not effectively integrate NLP-based enhancements or multilingual support within the same framework.

2.5 Limitations of Existing Systems and Research Gap

Although significant progress has been made in the field of gesture recognition, several challenges still remain. Many systems either rely on specialized hardware or struggle with real-time performance when deployed in practical environments. In addition, several solutions focus only on gesture detection without providing advanced features such as sentence refinement, speech output, or multilingual translation.Another limitation is that many research prototypes are designed for controlled laboratory conditions and do not perform consistently under real-world variations. These limitations highlight the need for an integrated system that combines computer vision, machine learning, and natural language processing to provide a more practical and accessible communication solution.

Positioning of the Proposed Work

The proposed project, “An Intelligent Computer Vision Framework for Hand Gesture Recognition in Sign Language,” aims to address the limitations of existing approaches by developing a real-time vision-based gesture recognition system. The system utilizes MediaPipe for accurate hand landmark detection and an SVM classifier to recognize sign language alphabets efficiently.In addition to gesture recognition, the system integrates NLP-based sentence enhancement, text-to-speech functionality, and multilingual translation, enabling more natural and effective communication. By combining these technologies into a single platform, the proposed framework provides a costeffective, accessible, and scalable solution for bridging the communication gap between sign language users and the general population.

III . METHODOLOGY :

3.1.System Architecture :

The architecture of the Sign Language Communication System is designed as a multi-layer intelligent framework that enables real-time hand gesture recognition and translation into meaningful text and speech. The system integrates computer vision, machine learning, natural language processing, and web technologies to create an efficient communication platform between sign language users and individuals unfamiliar with sign language. By utilizing a standard webcam and intelligent algorithms, the system ensures accessibility, accuracy, and real-time performance without requiring specialized hardware.

1. **Video Capture Module:** This module is responsible for capturing live hand gestures through a webcam connected to the user's computer or laptop. The webcam continuously records video frames of the user's hand movements. These frames act as the primary input for the system and are sent to the processing module for further analysis. The system supports real-time video streaming, allowing continuous monitoring of hand gestures.

2. **Image Preprocessing Module:**In this stage, the captured video frames are processed to improve image quality and prepare them for gesture recognition. Operations such as frame extraction, noise reduction, normalization, and background filtering are performed to enhance the visibility of hand movements. This module ensures that only relevant gesture information is extracted while minimizing interference from environmental noise or background objects.

3. **Hand Landmark Detection Module:**The system uses **MediaPipe**, an advanced computer vision framework, to detect hand landmarks within each frame. MediaPipe identifies multiple key points on the hand, including finger joints and palm coordinates. These landmarks represent the structural positions of the fingers and allow the system to understand the orientation and shape of the hand gesture accurately. This module converts raw image data into structured landmark data that can be used for feature extraction.

4. **Feature Extraction Module:**Once hand landmarks are detected, the system calculates finger joint angles and geometric relationships between landmarks. These extracted features represent the unique characteristics of each gesture. By converting hand movements into numerical feature vectors, the system simplifies the gesture recognition process and prepares the data for machine learning classification.

5. **Gesture Recognition Module:**This module serves as the core intelligence of the system. A Support Vector Machine (SVM) classifier is trained using the extracted gesture features from the dataset containing sign language alphabets (A–Z). During execution, the trained model analyzes incoming feature vectors and predicts the corresponding gesture. The predicted gestures are then converted into letters that form words or sentences.

6. **Natural Language Processing Module**:After gestures are converted into text, the Natural Language Processing (NLP) module enhances the generated sentence. This module performs tasks such as spell correction, grammar refinement, and sentence improvement using language processing tools. These enhancements help produce clear and meaningful sentences, improving the readability of the recognized text.

7. **Translation and Speech Output Module:**The system also supports multilingual translation and speech generation. Using translation APIs, the recognized text can be converted into different languages such as Hindi, Tamil, Telugu, Spanish, or French. Additionally, the Text-to-Speech engine converts the final sentence into audible speech, enabling more natural communication between users.

8. **User Interface Module:**The final module provides an interactive graphical user interface (GUI) where users can view recognized gestures, generated sentences, translated outputs, and system controls. The interface allows users to start or stop the camera, insert spaces between words, correct sentences, and convert text into sign images. This module ensures smooth interaction between the user and the system.

3.2 : Hardware requirements :

S.NO	COMPONENT	REQUIREMENT
1	Operating System	Windows 10 or above
2	Processor	Intel core i5 or above
3	RAM	Minimum 8 gb
4	Hard Disk	Atleast 25gb free space
5	Webcamera	Standard Webcam

Table 1: Hardware requirements

3.3.Software requirements:

S.NO	COMPONENT	REQUIREMENT
1	Programming Language	Python 3.9 or above
2	Frame Work	Flask
3	Libraries	Open CV, Scikit Learn, MediaPipe, Numpy
4	Development Environment	VS code
5	Web Technologies	HTML, CSS, JS

Table 2: Software requirements

3.4 :Working Principle:

1. **Gesture Input Capture**
   The system captures live hand gestures using a webcam connected to the computer. The webcam continuously records video frames of the user's hand movements.
2. **Frame Processing**
   The captured video stream is divided into individual frames. Each frame is processed to prepare it for gesture detection.
3. **Hand Landmark Detection**
   MediaPipe is used to detect hand landmarks such as finger joints and palm coordinates from each frame. These landmarks represent the structure of the hand.
4. **Feature Extraction**
   Finger joint angles and positional relationships between landmarks are calculated. These extracted features represent the gesture in numerical form.
5. **Gesture Classification**
   The extracted features are sent to the trained Support Vector Machine (SVM) model, which predicts the corresponding sign language alphabet.
6. **Text Generation**
   The predicted alphabets are combined to form words and sentences, which are displayed on the user interface.
7. **Sentence Enhancement**
   Natural Language Processing techniques are used to perform spell correction and grammar refinement to improve the generated sentence.
8. **Translation and Speech Output**
   The final sentence can be translated into multiple languages and converted into speech using text-tospeech functionality.
9. **User Interface Display**
   The processed text, translated output, and speech feedback are presented to the user through the system interface, enabling effective communication.

3.5 . Algorithm:

Step 1: Start the system and initialize the webcam to capture hand gestures.
Step 2: Continuously capture video frames from the webcam in real time.
Step 3: Convert each captured frame into RGB format for processing.
Step 4: Detect hand landmarks using the MediaPipe framework.
Step 5: Extract finger joint angles and positional features from the detected landmarks.
Step 6: Load the trained SVM model used for gesture classification.
Step 7: Provide the extracted features as input to the SVM classifier.
Step 8: Predict the corresponding sign language alphabet for the detected gesture.
Step 9: Append the predicted alphabet to form words or sentences.
Step 10: Apply NLP techniques to perform spell correction and grammar refinement.
Step 11: Display the enhanced sentence on the user interface and optionally convert it to speech.
Step 12: Repeat the process for continuous gesture recognition until the system is stopped**.**

IV. BLOCK DIAGRAM :

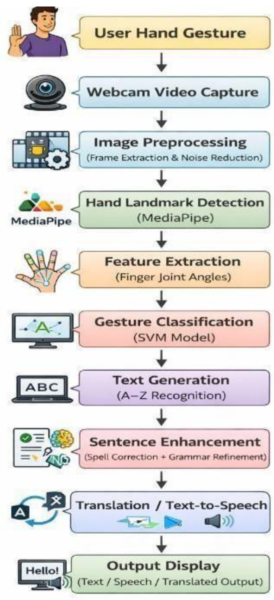


Fig 1: Block Diagram

1. **User Hand Gesture:**The process begins when the user performs a hand gesture in front of the camera. These gestures represent sign language alphabets or words that the system needs to recognize.
2. **Webcam Video Capture:**A webcam is used to capture the live video of the user's hand movements. The captured video provides the visual data required for the gesture recognition process.
3. **Image Preprocessing :**The captured video stream is divided into individual frames so that each frame can be analyzed separately. During this stage, noise and unnecessary visual disturbances are reduced to improve the overall clarity of the image.
4. **Hand Landmark Detection :**The system uses the MediaPipe framework to identify key points on the hand, such as finger joints and palm positions. These landmarks help the system understand the structure and movement of the hand.
5. **Feature Extraction :**After detecting the hand landmarks, the system calculates finger joint angles and spatial relationships between different points of the hand. These values are used as important features that describe each gesture.
6. **Gesture Classification :**The extracted features are provided to a trained Support Vector Machine (SVM) model. The model analyzes these features and determines which sign language alphabet the gesture represents.

7. **Text Generation :**Once the gesture is recognized, the predicted alphabet is converted into text. Multiple recognized letters can be combined to form meaningful words or sentences.
8. **Sentence Enhancement :**To improve the quality of the generated text, Natural Language Processing techniques are applied. These techniques help correct spelling errors and refine sentence grammar for better readability.
9. **Translation / Text-to-Speech:** The final sentence can be translated into different languages if required. The system can also convert the text into speech using a textto-speech module, making communication easier.
10. **Output Display** :The final result is presented to the user through the system interface. The output can appear as readable text, translated language, or spoken audio depending on the selected option**.**

V. Proposed Work

The proposed work focuses on developing an intelligent **sign language communication system** that can recognize hand gestures and convert them into meaningful text and speech in real time. The system captures hand movements through a webcam and processes the captured frames using computer vision techniques. MediaPipe is used to detect important hand landmarks such as finger joints and palm positions, which help represent the structure of the gesture. From these landmarks, features like finger joint angles are extracted and used as input for a machine learning model. A **Support Vector Machine (SVM)** classifier is trained using these features so that it can accurately recognize sign language alphabets. Once the gesture is recognized, the predicted letters are combined to form words and sentences. The generated sentence is further improved using natural language processing techniques such as spell correction and grammar refinement. The final output can also be translated into different languages or converted into speech, making the system more accessible and useful for communication between sign language users and others.

Algorithm:

A Support Vector Machine (SVM) is a machine learning algorithm that is mainly used for classification tasks. In this project, SVM is used to identify which sign language alphabet corresponds to a particular hand gesture. After the system extracts numerical features such as finger joint angles from hand landmarks, these values are given as input to the SVM model. The SVM then compares the input features with the patterns it learned during training and predicts the most suitable gesture class.

The working principle of SVM is based on finding an optimal boundary that separates different gesture classes in a feature space. This boundary, called a hyperplane, divides the data into different categories so that each gesture can be clearly distinguished. During training, the model learns the best position of this boundary using labeled gesture data. When new gesture features are provided during real-time execution, the model determines on which side of the boundary the input lies and classifies the gesture accordingly.

SVM is chosen for this project because it performs very well for classification problems involving structured feature data and relatively smaller datasets. It is efficient, reliable, and capable of producing accurate predictions even when the input features have complex relationships. These advantages make SVM a suitable choice for recognizing sign language gestures based on extracted hand landmark features.

A Support Vector Machine (SVM) is a machine learning algorithm that is mainly used for identifying patterns and classifying data into different categories. In this project, the SVM model is responsible for recognizing which sign language alphabet corresponds to a specific hand gesture. Before the model makes a prediction, the system performs calculations using the hand landmark coordinates detected by MediaPipe. These calculations include determining finger joint angles and measuring the positional relationships between different landmarks on the hand. These computed values act as feature inputs for the SVM model.

The SVM works by analyzing these features and separating different gesture classes using a decision boundary known as a hyperplane. During training, the model learns how different gestures appear based on the extracted features. When a new gesture is shown to the system, the calculated features are compared with the learned patterns, and the model predicts the most likely gesture category. SVM is used in this project because it is well suited for classification tasks involving structured feature data, and it performs efficiently even with moderate-sized datasets. Its ability to provide accurate and reliable predictions makes it a suitable choice for recognizing sign language gestures based on calculated hand landmark features.

VI. RESULTS & DISCUSSION

After executing the program, the flask development server has been activated to indicate that the application is running, and it has generated a URL to access the interface of the application

In Fig 2 The image shows the main user interface of the Sign Language Communication System, which is designed to enable two-way communication between sign language users and others. The interface is divided into two main sections: Sign to Text (Live Recognition) and Text to Sign (Human to Sign). On the left side, the system allows users to perform hand gestures in front of the camera, which are then recognized in real time and converted into text. The interface provides control buttons such as Start Camera, Stop Camera, Space, Finish Sentence, Backspace, and Translate Page, allowing the user to manage the gesture recognition process easily. The recognized sentences are displayed in the Completed Sentences section, where users can also clear the history if needed.

On the right side, the system provides a Text to Sign conversion feature, where users can type a sentence and convert it into corresponding sign language images.

In fig 3 displays the Sentence Translation interface of the Sign Language Communication System. This section of the system allows users to translate the recognized sentence into different languages for better understanding and accessibility. The interface first shows the recognized sentence, which is generated from the hand gesture recognition process. Users can then select a preferred language from the dropdown menu, which includes options such as Hindi, Tamil, Telugu, Spanish, and French. After selecting the desired language, the user can click the Translate button to convert the sentence into the chosen language. The translated result is then displayed in the Output section below. This feature improves the usability of the system by allowing communication across multiple languages, making the application more inclusive and helpful for users from different linguistic backgrounds.

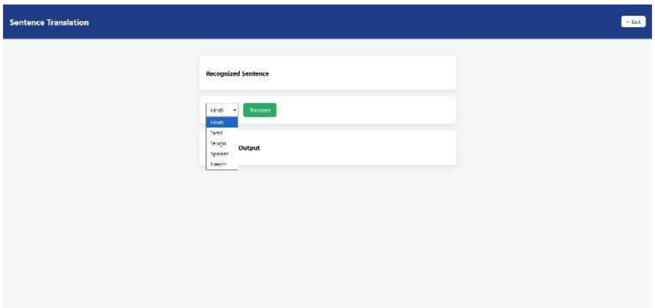


Fig 3 : Sentence Translation interface

In fig4 shows a section of the Sign -to-Text live recognition interface of the Sign Language Communication System. In this interface, users can perform hand gestures in front of the webcam, and the system recognizes these gestures in real time. The panel includes control buttons such as Start Camera, Stop Camera, Space, Finish Sentence, Backspace, and Translate Page, which help the user manage the gesture recognition process while forming sentences. Once the user completes a gesture-based input, the system converts the recognized alphabets into text and displays them in the Completed Sentences section. The system also improves the generated sentence using language processing techniques, which is shown through the Original and Enhanced sentence outputs. This feature allows the system to correct spelling and improve sentence clarity, making the final output easier to understand. Overall, this interface demonstrates how the system processes hand gestures, refines the generated text, and presents meaningful sentences to support effective communication.

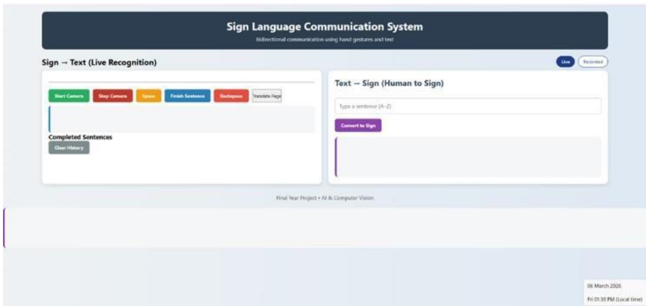
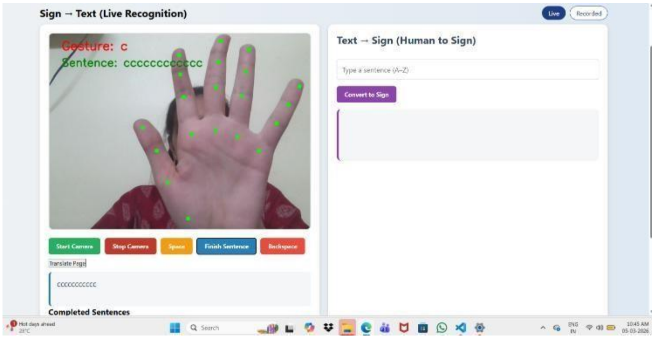


Fig2 :User interface of sign language communication system

Fig 4 : Demonstrating the web interface

In fig 5 illustrates the live gesture recognition functionality of the Sign Language Communication System. In this interface, the webcam captures the user's hand gesture in real time, and the system detects important hand landmarks using the MediaPipe framework. These landmarks are displayed as green dots on the hand, showing how the system tracks finger joints and palm positions to understand the gesture structure.



Based on these detected landmarks, the system extracts gesture features and predicts the corresponding sign language alphabet, which is displayed on the screen as the recognized gesture and sentence.. On the right side, the system also includes a Text-to-Sign module, where users can type a sentence and convert it into sign language representations. This interface demonstrates how the system performs realtime gesture detection, recognition, and text generation to support effective communication between sign language users and others.

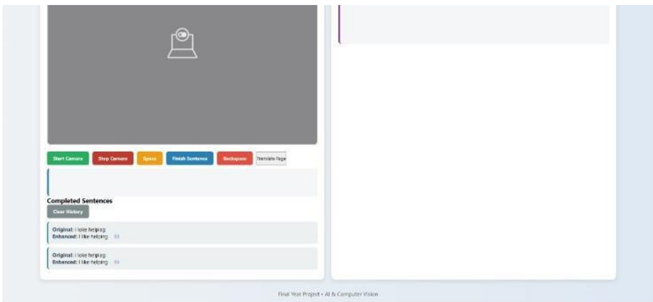


Fig 5 : 21 Landmarks in hand

In fig 6 On the right side, the Text → Sign (Human to Sign) section allows users to type a sentence and convert it into sign language representations. After entering text and clicking the Convert to Sign button, the system displays corresponding sign language images for each alphabet of the sentence. This feature helps users understand how a sentence can be expressed through sign language gestures. Overall, the interface demonstrates how the system enables two-way communication by recognizing hand gestures as text and also translating text into sign language visuals, making the application useful for both sign language users and people who want to learn or communicate using sign language.

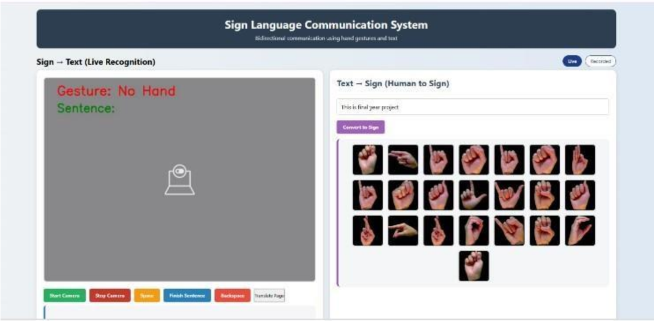


Fig 6 : Text to sign conversion Interface

In fig 7 shows the Sentence Translation interface of the Sign Language Communication System, which allows users to translate the recognized sentence into different languages. In this interface, the sentence generated from the gesture recognition process is displayed under the Recognized Sentence section. The user can then choose a desired language from the dropdown menu, such as French, Hindi, Tamil, Telugu, or Spanish. After selecting the language and clicking the Translate button, the system processes the request and converts the sentence into the selected language. The translated result is then shown in the Translated Output section. This feature improves the usability of the system by allowing users to communicate across different languages, making the application more accessible and helpful for people from diverse linguistic backgrounds.

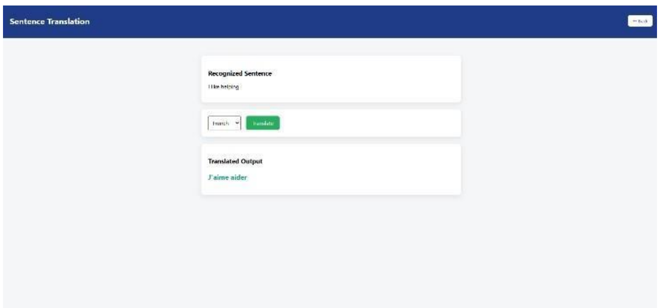


Fig 7 : Sentence Translation Output Showing MultiLanguage Conversion

VII CONCLUSION:

This project presents an intelligent Sign Language Communication System that helps reduce the communication gap between individuals who use sign language and those who do not understand it. By using computer vision and machine learning techniques, the system is able to recognize hand gestures and convert them into meaningful text in real time. The use of MediaPipe for hand landmark detection and an SVM model for gesture classification allows the system to identify sign language alphabets effectively. This approach makes the system practical and accessible since it only requires a standard webcam rather than specialized hardware. Another important aspect of the project is the integration of Natural Language Processing techniques that enhance the generated sentences through spell correction and grammar refinement. This ensures that the output is not only recognized correctly but also presented in a more readable and understandable form. In addition to gesture recognition, the system also provides useful features such as text-tospeech conversion and multilingual translation, allowing users to communicate in different languages and formats. These additional functionalities make the system more flexible and useful for real-world communication scenarios. Overall, the developed system demonstrates how modern technologies such as computer vision, machine learning, and natural language processing can be combined to create an inclusive communication tool. The project provides a simple and cost-effective solution that supports both sign-to-text and text-to-sign interaction, making communication easier for people with hearing or speech impairments. With further improvements and expanded gesture datasets, the system has the potential to become a more advanced assistive technology that can be widely used in educational, social, and professional environments

VIII FUTURE WORK:

Although the developed system successfully recognizes sign language gestures and converts them into text and speech, there are several opportunities to further improve and expand the project in the future. One possible enhancement is to extend the system beyond alphabet-level recognition to support complete words, phrases, and dynamic gestures. Sign language often involves continuous hand movements and facial expressions, so incorporating these elements would make the system more natural and closer to real human communication.

Another area for future improvement is the use of advanced deep learning models and larger gesture datasets. By training the system with more diverse gesture samples collected from different users and environments, the recognition accuracy can be further improved. The system could also be optimizedto work on mobile devices or embedded platforms, allowing users to access the application through smartphones or portable devices without requiring a computer setup. In addition, the project can be enhanced by integrating voice recognition and real-time conversation features, enabling full two-way communication between sign language users and speakers. Features such as automatic sentence prediction, emotion recognition, and support for more regional and international sign languages could also be included. With these improvements, the system could evolve into a more comprehensive assistive communication tool that can be used in educational institutions, workplaces, healthcare environments, and everyday social interactions.

REFERENCES:

1. Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition.* International Conference on Learning Representations (ICLR).
2. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
3. Zhang, F., Bazarevsky, V., Vakunov, A., et al. (2020). *MediaPipe Hands: On-device Real-time Hand Tracking.* arXiv preprint arXiv:2006.10214.
4. Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). *ImageNet Classification with Deep Convolutional Neural Networks.* Advances in Neural Information Processing Systems (NeurIPS).
5. Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). *Hand Gesture Recognition with 3D Convolutional Neural Networks.* IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
6. Neverova, N., Wolf, C., Taylor, G., & Nebout, F. (2014). *Multi-scale Deep Learning for Gesture Detection and Localization.* European Conference on Computer Vision Workshops (ECCVW).
7. Freeman, W. T., & Roth, M. (1995). *Orientation Histograms for Hand Gesture Recognition.* International Workshop on Automatic Face and Gesture Recognition.
8. Wang, R. Y., & Popović, J. (2009). *Real-Time HandTracking with a Color Glove.* ACM Transactions on Graphics.
9. Pigou, L., Dieleman, S., Kindermans, P., & Schrauwen, B. (2015). *Sign Language Recognition Using Convolutional Neural Networks.* European Conference on Computer Vision Workshops (ECCVW).
10. Ohn-Bar, E., & Trivedi, M. (2014). *Hand Gesture Recognition in Real Time for Automotive Interfaces.* IEEE Transactions on Intelligent Transportation Systems.
11. Dalal, N., & Triggs, B. (2005). *Histograms of Oriented Gradients for Human Detection.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
12. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.
13. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.
14. Shotton, J., Fitzgibbon, A., Cook, M., et al. (2011). *RealTime Human Pose Recognition in Parts from Single Depth Images.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
15. Mitra, S., & Acharya, T. (2007). *Gesture Recognition: A Survey.* IEEE Transactions on Systems, Man, and Cybernetics.
16. Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2019). *OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields.* IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).
17. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention Is All You Need.* Advances in Neural Information Processing Systems (NeurIPS).
18. Hochreiter, S., & Schmidhuber, J. (1997). *Long ShortTerm Memory.* Neural Computation.
19. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). *Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D CNNs.* IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
20. Oyedotun, O. K., & Khashman, A. (2017). *Deep Learning in Vision-Based Static Hand Gesture Recognition.* Neural Computing and Applications.
21. Huang, J., Zhou, W., & Li, H. (2018). *Sign Language Recognition Using 3D Convolutional Neural Networks.* IEEE International Conference on Multimedia and Expo (ICME).
22. Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2018). *Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition.* International Journal of Computer Vision.
23. Neverova, N., Wolf, C., Taylor, G. W., & Nebout, F. (2016). *ModDrop: Adaptive Multi-Modal Gesture Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence.
24. Zhang, X., Liu, Z., & Sun, L. (2020). *Hand Gesture Recognition Based on Deep Learning: A Review.* Journal of Computer Science and Technology.
25. Chen, X., Wang, G., Guo, H., & Zhang, C. (2019). *PoseGuided Hand Gesture Recognition Using CNN.* Pattern Recognition Letters.
26. Koller, O., Ney, H., & Bowden, R. (2015). *Deep Hand: How to Train a CNN on 1 Million Hand Images for Gesture Recognition.* IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
27. Wan, J., Escalera, S., Guyon, I., et al. (2016). *Results and Analysis of ChaLearn LAP Multi-Modal Isolated and Continuous Gesture Recognition.* IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).
28. Mittal, A., Zisserman, A., & Torr, P. H. S. (2011). *Hand Detection Using Multiple Proposals.* British Machine Vision Conference (BMVC).
29. Liu, J., Shahroudy, A., Xu, D., et al. (2017). *SkeletonBased Action Recognition Using Spatio-Temporal LSTM Networks.* IEEE International Conference on Computer Vision (ICCV).
30. Yang, W., Zhang, C., & Tian, Y. (2019). *Deep Learning for Vision-Based Hand Gesture Recognition: A Survey.* IEEE Acces