International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AI-Driven Gesture Recognition and Multilingual Translation

[1]Prof. Ravindra Patil

Assistant Professor Department of CSE,
KLS Vishwanathrao Deshpande Institute of Technology Haliyal
Uttara-Kannada District, Karnataka, India

rtp@klsvdit.edu.in

[2]Nikhil Suresh Yallaraddi, Prasannakumar Pattar, Prateek Hadli, Pritam A Shetty
Final Year Students Department of CSE
KLS Vishwanathrao Deshpande Institute of Technology Haliyal
Uttara-Kannada District, Karnataka, India
2vd21cs031@klsvdit.edu.in , 2vd21cs034@klsvdit.edu.in
2vd21cs035@klsvdit.edu.in , 2vd21cs036@klsvdit.edu.in

Abstract **– This paper presents a real-time system designed to improve communication effectively for everyone with speech and hearing impairments through gesture-based language translation. This approach uses machine learning algorithms to interpret American Sign Language (ASL) hand gestures which is a universal sign language and convert them into both text and speech outputs. By integrating Mediapipe for landmark detection with a Convolutional Neural Network (CNN) for gesture classification, the system effectively identifies static hand signs and ensures robustness in diverse surrounding environment with proper lighting conditions. The recognized gestures are further translated into various languages using natural language processing techniques, followed by speech synthesis through text-to-speech tools. The implementation includes a user-friendly graphical interface, enabling live gesture capture and multilingual feedback, making the system both accessible and adaptable for real-world scenarios.**

I. INTRODUCTION

Communication is an important aspect of daily life, yet person with deaf and dumb challenges often encounter significant barriers when interacting with others. While the standard sign languages such as American Sign Language (ASL) offer an effective visual communication method, the lack of widespread understanding among the general population limits its accessibility. This disconnect requires a system that can interpret sign language in a way that is universally understood text and speech.

This project covers the gap by developing an AI-powered platform capable of interpreting ASL gestures and converting them into readable and audible formats. The system utilizes a vision-based approach, employing Mediapipe to extract hand landmarks from real-time video input. These landmarks are drawn onto a clean background and processed through CNN model, which classifies the gestures with maximum accuracy. Once recognized, the system translates the result into regional languages using translation APIs and outputs audio via text-to-speech engines.

The system is further enhanced with a GUI that enables the users to build words and sentences from individual gestures, facilitating smoother and more effective communication.

Despite the advancements in gesture recognition technologies, most existing systems are constrained by limited functionality, often focusing solely on gesture classification without providing contextual language translation or speech output. Moreover, these systems typically rely on controlled environments with clean backgrounds and consistent lighting, making them less effective in practical settings. By combining deep learning with robust image processing techniques, this project aims to overcome those limitations and offer a comprehensive communication aid. The integration of multilingual support and real-time text-to-speech conversion in this project ensures that it is not only useful for everyone with speech and hearing impairments but also can be used as versatile tool in multiple contexts.

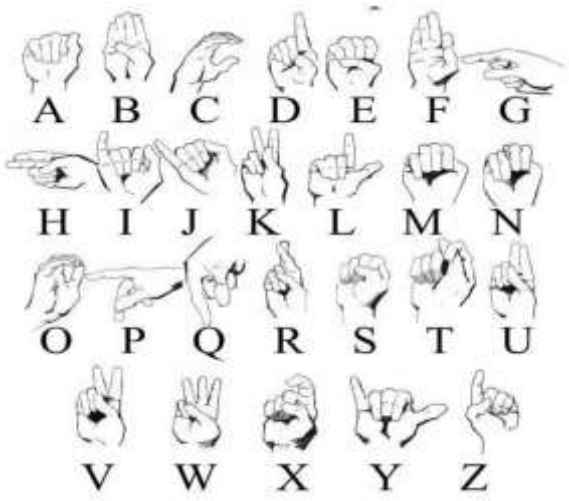

Fig 1 – Gesture chart

II. LITERATURE SURVEY

From many years, range of studies have explored devoted to improving the accuracy of sign language recognition systems. Many techniques have been proposed, each offering unique advantages and facing specific limitations with respect to performance, complexity, and practical deployment.

Mahesh Kumar (2018) introduced a MATLAB-based identification system for 26 alphabetical ISL hand gestures. The framework included image pre-processing, segmentation, feature extraction using eigenvectors, and differentiating via Linear Discriminant Analysis (LDA) [1]. While efficient in processing, it struggled with variability in hand orientation and lighting. Additionally, the dependence on MATLAB limited its portability to mobile or real-time systems, reducing its practical deployment potential

Rakesh Kumar (2021) proposed a lightweight contour-based recognition algorithm for ASL characters and symbols. By analyzing image contours and convexity features, the system avoided complex computations and hardware requirements [6]. Though accurate (86%), it could not generalize well to overlapping or occluded gestures. The model's effectiveness was highest in controlled environments, indicating limited robustness in dynamic or cluttered real-world scenarios.

Ankit Ojha developed a CNN-based desktop application that recognized ASL gestures and converted them into text and speech. Utilizing webcam input and a deep neural network, the system reached approximately 95% accuracy. This work highlighted the power of convolutional layers in learning spatial hierarchies within hand shapes. However, the system was primarily tested on static gestures and did not incorporate continuous gesture streams or sentence-level translation

Victorial Adebimpe Akano (2018) designed a system combining supervised and unsupervised learning to change sign language into text and audio. The system utilized the Kinect sensor which is used for data acquisition and used KNN for classification, supported by FAST and SURF for feature extraction. Despite achieving 92% accuracy in supervised scenarios, performance dropped in unsupervised learning. The use of depth sensors improved detection, but reliance on specialized hardware made it less accessible for widespread use.

Krishna Modi (2013) investigated the use of Blob Analysis, a method focused on identifying continuous pixel regions within binary images [2]. This approach achieved an accuracy rate of 93%, demonstrating strong performance in isolating hand gestures from simple backgrounds.

However, its effectiveness diminished in scenarios involving visual clutter or irregular lighting, as it heavily relied on precise segmentation. Moreover, its application was limited to environments with consistent visual contrast, reducing its adaptability to real-world variability.

Bikash K. Yadav (2020) implemented gesture identification using CNN, reaching a commendable accuracy of 95.8%. By leveraging CNN's multi-layered structure, the system can instantly extract important characteristic derived from hand gesture images [3]. This work highlights the suitability of CNNs for visual recognition tasks involving complex spatial patterns. It also reflected the increasing reliance on deep learning to enhance the working ability of human-computer interaction interfaces.

Ayush Pandey (2020) also adopted a CNN-based strategy and achieved accuracy with 95%, closely matches with contemporary deep learning benchmarks. His system showcased strong generalization across hand gesture datasets with varying shapes and orientations, emphasizing CNN's robustness in static image classification [4]. Additionally, his findings supported the notion that well-trained CNNs can distinguish between fine-grained differences in gestures, even when the input conditions were not ideal.

This literature review demonstrates a clear evolution in gesture recognition—from basic statistical techniques to advanced deep learning models. CNN-based systems are chosen for its capability to learn complex features and maintain high accuracy across various conditions. Nevertheless, traditional approaches still offer value in specific use cases, especially where computational resources are limited.

Author Name	*Algorithm*	*Accuracy*	*Year*
Krishna Modi	*Blob Analysis*	*93%*	*2013*
Victorial Adebimpe Akano	*KNN*	*92%*	*2018*
Mahesh Kumar	*LDA*	*80%*	*2018*
Bikash K. Yadav	*CNN*	*95.8%*	*2020*
Ayush Pandey	*CNN*	*95%*	*2020*
Rakesh Kumar	*Contour Measurement*	*86%*	*2021*

III. SYSTEM ARCHITECTURE

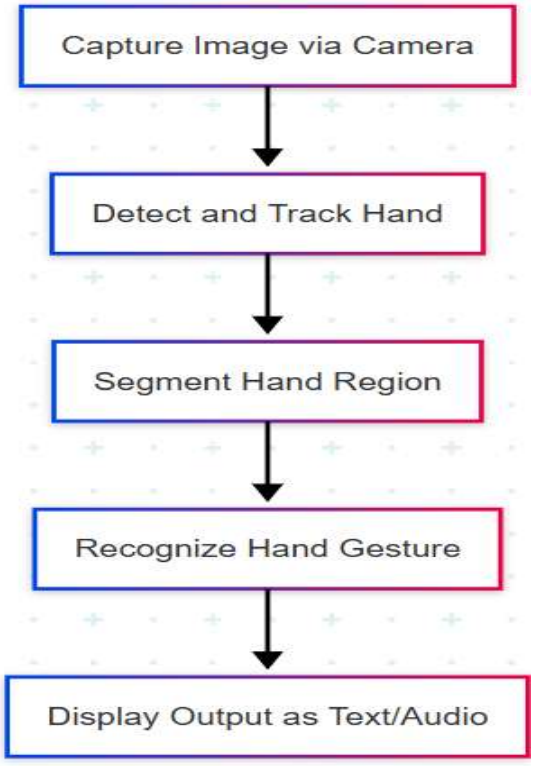

Fig 2 – System Architecture

Step 1: Data Acquisition:

The initial phase of the system involves capturing hand gesture data through image acquisition, primarily using a standard webcam. Unlike glove-based methods that require specialized and costly hardware, this vision-based approach offers a more accessible and natural interface by leveraging common camera devices. The camera continuously captures real-time frames containing hand movements, which serve as the raw data for the subsequent processing stages. The primary challenge in this step is managing the variability in hand appearance due to diverse skin tones, different surroundings, various lighting conditions, and varying hand orientations. Despite these challenges, the usage of a webcam as the

acquisition device ensures the system remains cost-effective and user-friendly, making it suitable for everyday applications without requiring additional equipment.

Step 2: Data Pre-processing and Feature Extraction:

Once the hand images are captured, they undergo several pre-processing stages to prepare the data for accurate gesture recognition. Initially, the hand region is detected within the frame using the Mediapipe library, which efficiently identifies key hand landmarks regardless of background complexity or lighting variations. The detected region of interest is then cropped and converted into a grayscale image using OpenCV, simplifying the image information while ensuring essential features are kept.

To reduce noise and smooth the image, a Gaussian blur filter is applied, enhancing the quality of the input for further analysis. Subsequently, the grayscale image is transformed into a binary format using thresholding techniques, including adaptive thresholding, to clearly differentiate the hand from the white background. This sequence of pre-processing ensures that the features extracted are consistent and robust, providing a reliable foundation for the classification model to accurately interpret diverse hand gestures.hand.

To improve accuracy, the detected hand landmarks are drawn on a plain white background. This helps eliminate the effects of varying lighting and complex backgrounds, allowing the system to focus on the shape and position of the hand. Using these key points simplifies the input and enhances the model's ability to recognize subtle differences between gestures, improving overall recognition performance.

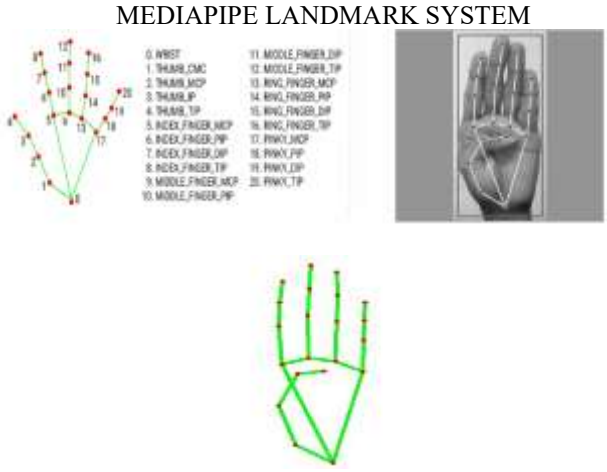

Fig 3 – Media Pipe Landmark System

The obtained landmark points are then set on a plain white canvas utilizing the OpenCV library. This approach effectively minimizes the impact of varying backgrounds and lighting conditions, as the Mediapipe library consistently provides accurate landmark detection regardless of these factors. Using this method, we have compiled a dataset of 180 skeletal hand images representing the alphabets from A to Z.

Step 3: Gesture Classification:

Convolution based Neural Networks are a specialized category of hierarchical learning models more often used for image-related tasks because of their exceptional ability to automatically identify spatial features. Inspired by the human visual cortex, CNNs operate by applying multiple filters or kernels that scan across the pixel data of an input image, extracting essential features such as edges, textures, and patterns [4]. This process is accomplished through a series of layers designed to progressively detect more complex and abstract representations, starting from simple lines to detailed object parts.

Unlike previously used neural networks in which each neuron is connected to all neurons in the prior layer, CNNs organize neurons in three dimensions: width, height, and depth. Each neuron connects only to a tiny, localized area of the input—defined by the receptive field—enabling the network to capture local spatial relationships efficiently. This localized connectivity significantly decreases the number of parameters and enhances computational efficiency.

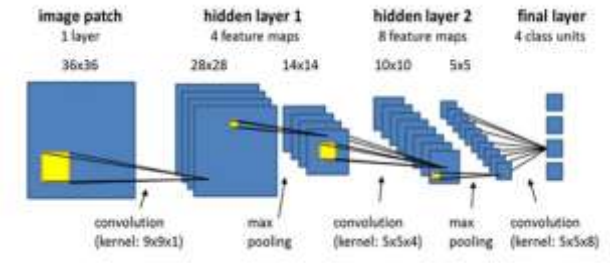

Fig 4 – CNN Architecture layers

The CNN architecture typically includes the following layers:

Convolutional Layer:

This layer applies a group of learnable filters with small spatial dimensions (for example, 5x5) across the entire image. As each filter overlaps the input with a mentioned stride, it computes mathematical operation that combines the filter weights and the input pixels in the current region. The output is a two-dimensional activation map that highlights the presence of specific features such as edges or color patches at different spatial locations.

Pooling Layer:

To overcome the physical area occupied by the feature maps and control overfitting, pooling layers are introduced. The two common pooling operations are:

Max Pooling: Selects the maximum value within a window (e.g., 2x2), preserving the most prominent feature.
Average Pooling: Computes the arithmetic mean of the values within the window, providing a smooth down sampling effect. These operations shrink the feature maps, making the network more tolerant to small translations in the input.

Fully Connected Layer:

After multiple feature retrieving and down sampling layers, feature maps are reshaped into one-dimensional vector. This vector feeds into at least one completely connected layers, where every neuron collects input from the earlier layer. These layers interpret the extracted features and perform classification by producing output probabilities for each gesture class.

Fig 5 – Gesture Datasets

For this project, the CNN model was instructed on a dataset of 180 pre-processed images listing letters from A to Z. Given the challenge of accurately classifying all 26 alphabets individually, the classes were initially grouped into 8 clusters of related hand gestures to enhance model performance. For example, consider one cluster includes the letters Y and J, another includes C and O, and so on. After initial cluster classification, a secondary process uses mathematical operations on the detected hand landmarks to further distinguish the exact alphabet within the cluster, such as differentiating between A, E, M, N, S and T in their group.
The model assigns probability scores to each possible gesture label, selecting the label with the highest probability as the recognized gesture. This hierarchical classification strategy optimizes accuracy by reducing misclassifications that could arise from visually similar hand signs.

Step 4: Text To Speech Translation:

Following the identification and translation of hand gestures into text, the system integrates a Text-to-Speech (TTS) component to produce spoken language output. This functionality allows the interpreted gestures to be vocalized, facilitating easier communication, especially for users who benefit from auditory feedback [10].

The system employs two distinct TTS frameworks to cover both offline and online requirements:

pyttsx3 acts as the offline speech synthesis engine. It operates locally without the need for internet connectivity, providing flexibility for users in environments with limited or no network access. This engine offers customization options such as voice selection and speech rate adjustments to cater to user preferences.

Google Text-to-Speech (gTTS) is utilized for high-quality, multilingual speech generation. By connecting to Google's API, the system supports multiple languages, including several regional dialects, enhancing accessibility for diverse populations. The cloud-based nature of gTTS ensures natural-sounding voice output and continuous updates to speech quality.

Users can easily choose the desired language and initiate the speech output, enabling seamless interaction through both visual and auditory channels. This combined TTS approach enhances system versatility, making it effective in a variety of practical scenarios and user environments.

IV. IMPLEMENTATION AND RESULTS

The system uses Media pipe for precise hand landmark detection, OpenCV for image processing, and a CNN built with TensorFlow and Keras to classify gestures. Real-time hand movements captured by a webcam are mapped onto a plain background to reduce noise. The CNN, trained on 180 skeleton images of ASL alphabets grouped into clusters, achieves reliable gesture recognition.

Testing shows about 97% accuracy in typical indoor settings and up to 99% in controlled environments. The trained model works with variety of languages for speech output using offline (pyttsx3) and online (gTTS) text-to-speech engines, including English, Hindi, Kannada, Marathi, and Tamil. A simple interface displays recognized gestures and plays audio, making it a practical tool for assisting interaction for individual with speech and hearing impairments [21].

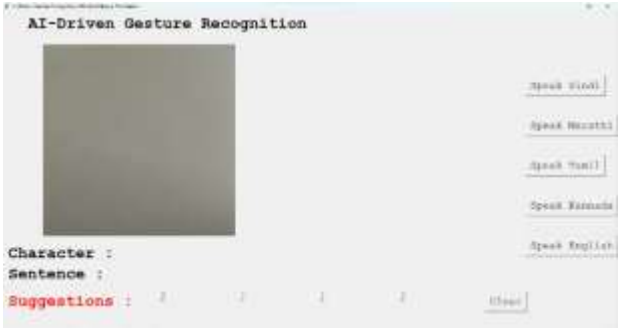

Fig 6 - User Interface of AI-Driven Gesture Recognition with Multilingual Translation

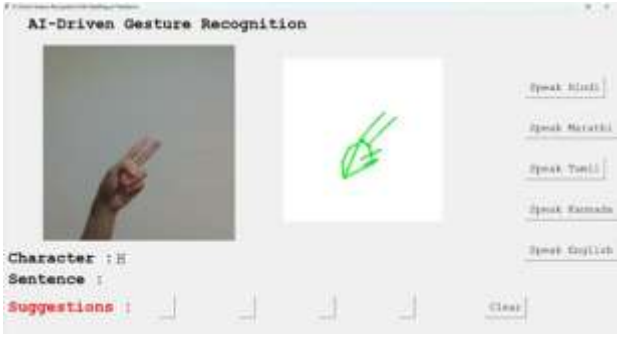

Fig 7 - Recognition of H alphabet from ASL sign language

Fig 8 - Space gesture with hello written

Fig 9 - Example of Building a Sentence Using Hand Gestures

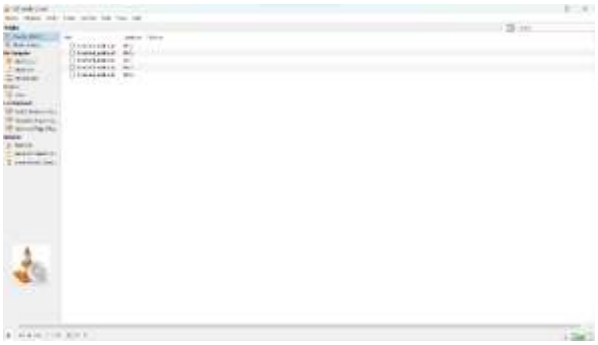

Fig 10 - Translated voice output stored in mp3

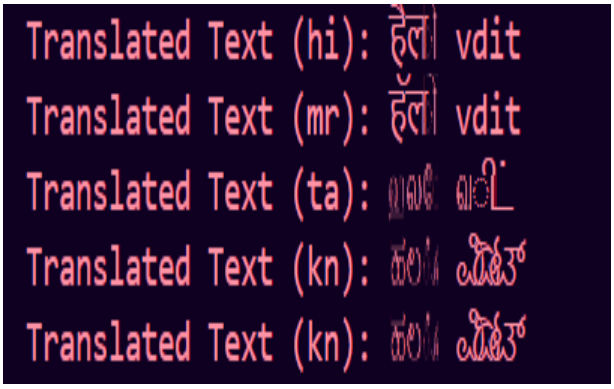

Fig 11 - Recognized sentence converted into different languages

CONCLUSION

Our approach successfully achieves a recognition accuracy of approximately 97% for all alphabets (A–Z), even in less-than-ideal lighting and background conditions. Under optimal circumstances with clear backgrounds and proper lighting, the accuracy improves to nearly 99%. Looking ahead, we plan to extend this work by developing an Android application that integrates this gesture recognition algorithm, enhancing accessibility and enabling real-time gesture prediction on mobile devices. This advancement will further support effective communication for users with speech and hearing challenges in everyday environments.

REFERENCES

[1] M. Kumar, "Hand Gesture Recognition Using LDA Algorithm," *International Journal of Computer Applications*, vol. 179, no. 1, 2018.
[2] K. Modi, "Hand Gesture Recognition Using Blob Analysis," *International Journal of Research in Computer Science*, vol. 4, no. 2, 2013.
[3] B. K. Yadav, "American Sign Language Identification Using CNN," *Procedia Computer Science*, vol. 167, pp. 1234-1241, 2020.
[4] A. Pandey, "Static Hand Gesture Recognition Using Deep CNN," *IEEE Access*, vol. 8, pp. 150000-150010, 2020.
[5] V. A. Akano, "Sign Language Alphabet Recognition Using KNN," *Journal of Artificial Intelligence Research*, vol. 61, pp. 237-246, 2018.
[6] R. Kumar, "Contour-Based Hand Gesture Recognition," *International Journal of Research in Science, Engineering and Technology*, vol. 10, no. 5, 2021.
[7] W. Kadous, "Machine Recognition of Auslan Signs Using PowerGloves," *Ph.D. dissertation, University of New South Wales*, 1995.
[8] M. A. Hossain and P. Chetty, "Deep Learning-Based Hand Gesture Recognition," *International Journal of Computer Applications*, vol. 182, no. 1, 2018.
[9] R. Sau et al., "Vision-Based Hand Gesture Recognition Techniques: A Review," *IEEE Access*, vol. 8, pp. 192459-192474, 2020.
[10] S. Antad et al., "Sign Language Translation Across Multiple Languages," *IEEE ESIC Conference*, 2024.
[11] Y. Wu and T. S. Huang, "Vision-Based Gesture Recognition: A Review," *Proceedings of the International Gesture Workshop*, pp. 103–115, 1999.
[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations (ICLR)*, 2015.
[13] A. Graves et al., "Speech Recognition with Deep Recurrent Neural Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.
[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[15] H. Cooper et al., "Sign Language Recognition," *Machine Learning for Gesture Recognition*, Springer, 2012.
[16] M. Wöfl et al., "Real-time Hand Gesture Recognition using CNN," *Journal of Imaging*, vol. 6, no. 8, 2020.