

# HAND LANDMARK-BASED SIGN LANGUAGE ALPHABET RECOGNITION USING MEDIAPIPE AND MACHINE LEARNING

EKTA KANSARA<sup>1</sup>, DIYA AGARWAL<sup>2</sup>, SNEHA KANSARA<sup>3</sup>, DR. SANDEEP KULKARNI<sup>4</sup>

<sup>1-3</sup> Student, Ajeenkya DY Patil University, Pune, Maharashtra, India,

<sup>4</sup> Assistant Professor, Department of Computer Science, Ajeenkya DY Patil University, Pune, Maharashtra, India

\*\*\*

**Abstract-** *The communication gap between the Deaf community and the hearing community has created a need for accessible assistive technologies. This paper introduces a sign language recognition and illustration system based on static hand gesture detection to form sentences. The proposed system differs from other hardware-based techniques in its use of a contactless approach based on computer vision with OpenCV/CV2 and MediaPipe Hands to detect a 3D skeletal model of the hand with 21 points. The system uses a 63-element feature vector, which is invariant to environmental noise, with a detection threshold set at 0.3 to acquire the landmarks. A dataset was created in real-time with 1,300–1,400 samples to train a Random Forest classifier, which was serialized using Joblib. The vocabulary of this system includes 26 ASL alphabets, 10 numeric signs ranging from 1 to 10, and 8–10 basic communication words that are frequently used. A new temporal stability buffer logic is used for moving from isolated gestures to sentence construction. This also includes "Space" and "Delete" gestures for text editing. In addition to this, this system also uses gTTS (Google Text-to-Speech) and Pygame for providing audio feedback. TextBlob is used for text processing. During experimental evaluation using a dataset of 276 samples, exceptional results were obtained. This is because this system has obtained a perfect accuracy, precision, recall, and F1-score of 1.00. This stability filter has successfully minimized flickering. However, this performance may be influenced by controlled dataset conditions and requires further validation in real-world environments.*

**Keywords:** Sign Language Recognition, MediaPipe, Hand Skeleton Tracking, Machine Learning, Real-Time Translation, Random Forest, gTTS, TextBlob, OpenCV.

## I. INTRODUCTION

Sign language is a highly evolved form of communication that uses a combination of manual gestures, facial expressions, and body language to convey meaning. Unlike spoken languages, sign languages are usually spatial gestural languages that may not be governed by the same grammatical rules or word/sentence structures as the spoken counterparts. Additionally, they may not

be organized in a set format as in the case of written or spoken languages [1].

In the past, people with hearing impairment have faced major barriers in society. The term "deaf and dumb" represents the misunderstanding of the past where people who were unable to hear were also unable to talk [2]. Children who suffer hearing impairment in early stages of development due to illness or accident often lose the power of speaking in no time because they are unable to learn the language by imitating others [3, 4]. As there are various distinct sign languages in the world, such as British Sign Language (BSL), Spanish Sign Language (LSE), Arabic Sign Language (ArSL), and American Sign Language (ASL), communication between the Deaf community and the hearing community is a major challenge [5]. For effective communication to take place, hearing individuals need professional translators, which is not only costly but also compromises the privacy of the hearing-impaired individual [6].

The magnitude of this problem is enormous. According to the World Health Organization (WHO), over 5% of the world's population, which translates to 430 million people, needs to be rehabilitated for "disabling" hearing loss [8]. The figure is projected to reach almost 700 million by the year 2050 for debilitating hearing loss [9]. Despite the need for assistive technology for hearing-impaired individuals, sign language has not received as much academic attention as natural language processing because of its complexity and the advanced computer vision required for interpreting sign language [10].

Sign Language Recognition (SLR) is a significant subdomain of the field of machine translation and human-computer interaction (HCI), with research in the field tracing back to the 1940s [11]. In the development of such systems, there are two major phases involved: feature extraction and classification [12, 13]. Feature extraction is particularly challenging in the case of dynamic signs, where the sequences of signs in various frames of the video have to be extracted. This is then processed by the classifier to arrive at the probability of the signs representing the intended meaning [14]. However, the existing research is mostly limited to manual features, and there is a significant absence of

systems that can be practically applied in the real world [15, 16].

To address these limitations, this research aims to develop a real-time system capable of not only recognizing individual sign language alphabets but also forming meaningful, editable sentences from sequential inputs. By leveraging a lightweight, camera-based approach, this study assesses the viability of an interpretation solution that is accessible to the general public without the need for specialized hardware.

The main goals of this research work are:

1. To develop a real-time system for recognizing sign language alphabet gestures using the MediaPipe Hand Skeleton.
2. To implement a temporal logic for forming meaningful sentences from sequential gesture inputs.
3. To evaluate the effectiveness of incorporating control gestures (Space and Delete) for real-time text editing.

## II. Literature Review

The field of Sign Language Recognition (SLR) has evolved significantly, driven by the need to bridge the communication gap between the hearing-impaired and the hearing world. This review categorizes previous research into hardware-based methods, vision-based approaches, and the recent shift toward landmark-based real-time systems.

### 2.1 Evolution of Recognition Methods

#### 2.1.1 Traditional Hardware-Based Methods

Early SLR systems relied heavily on wearable technology, such as data gloves and motion sensors, to track finger articulation and hand orientation. While these methods provided high accuracy by directly capturing physical movements, they were inherently limited. As noted in recent reviews, such systems are costly, invasive, and impractical for daily use [17, 18]. The requirement for specialized hardware prevents these systems from being scaled for general public use, leading researchers toward contactless solutions [21].

#### 2.1.2 Computer Vision and Image-Based Approaches

To overcome the invasive issue of wearables, vision-based systems were developed, employing conventional cameras for image acquisition and image processing techniques [19, 20]. Initially, skin color detection, edge detection, and shape features were employed. However, these techniques were associated with considerable challenges:

**Sensitivity to Environment:** These systems were very sensitive to changes in illumination conditions and camera orientation.

**Complex Background:** Overlapping skin areas or presence of background objects similar to a hand in the scene were major issues.

**Computational Overhead:** Real-time processing of raw pixel values or high-resolution video streams was computationally expensive, requiring high-end hardware.

### 2.2 The Paradigm Shift: Skeleton-Based Recognition

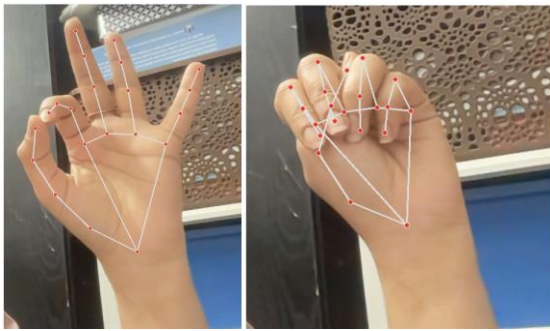
Recent research has moved toward Skeleton-Based Recognition, which defines model parameters based on geometric properties rather than raw pixels [17, 21]. This method utilizes a skeletal hand model to define constraints, trajectories, and correlations between joints. The most prominent features applied in this paradigm include:

**Joint Orientation and Space:** Measuring the relative distance and angle between skeletal joints.

**Skeletal Joint Position:** Using  $(x, y, z)$  coordinates to create a 3D representation of the hand.

**Trajectories:** Tracking the movement paths of specific joints over time.

This skeletonization approach, primarily enabled by frameworks like MediaPipe, offers superior robustness against background noise and environmental variations compared to traditional image-based techniques [21].



**Fig -1: Skeleton Recognition**

### 2.3 Role of MediaPipe in Hand Landmark Detection

MediaPipe has emerged as the industry standard for real-time hand detection. It offers a lightweight and efficient platform capable of detecting 21 landmarks on the hand with high accuracy [5, 7]. By providing a "contactless communication" bridge, MediaPipe allows researchers to build systems that are invariant to scale and rotation [21].

### 2.4 Machine Learning and Sentence Formation

While machine learning algorithms (SVM, Random Forest, etc.) are effective at classifying static hand gestures, their application in real-time sentence formation is less explored. Most existing work focuses on isolated alphabets or gestures [13, 15]. Forming complete sentences involves temporal complexities, such as handling repeated gestures and providing a robust mechanism for word segmentation [16].

### 2.5 Limitations of Existing Research

Despite the progress detailed in the literature, several critical limitations persist in current SLR research:

1. Focus on Isolated Gestures: A significant portion of research remains restricted to recognizing single characters, failing to address fluid sentence formation [12, 16].
2. Hardware Constraints: Many high-accuracy systems still require specific camera configurations (e.g., TOF or IR cameras) [18, 22].
3. Dataset Gaps: Pre-existing datasets often lack the real-world noise encountered in live environments.
4. Lack of User Control Logic and Multi-Modal Output: Many systems do not include intuitive "control gestures" (like Space and Delete) or text-to-speech (TTS) conversion. A

notable gap in the researched subject is that no previous research paper has successfully integrated real-time text-to-speech conversion in a lightweight, skeleton-based framework. This research solves this problem by providing immediate auditory feedback, closing the communication loop for both the sender and the receiver.

### 2.6 Motivation for the Current System

The current project addresses these gaps by proposing a lightweight, Python-based system that utilizes the MediaPipe Hand Skeleton as its core feature extractor. By incorporating real-time data collection and specific logic for sentence formation, this system provides a practical and scalable solution.

## III. METHODOLOGY

This section describes the systematic approach to the development of the sign language illustration system, including real-time data acquisition, skeleton-based landmark detection, feature extraction, model training, and sentence formation logic.

### 3.1 Real-Time Data Collection

A new dataset was created to accommodate the intricacies of static hand gestures for all 26 alphabets (A-Z), numbers ranging from 1 to 10, and a list of 8-10 basic words used in communication, along with "Space" and "Delete" commands. Unlike using pre-existing datasets of static images, this dataset was created using a live feed from a webcam.

**Interactive Capture:** A new interface was created that allowed the user to input each gesture in front of a live webcam feed. For each symbol or word, a total of 30 different instances were captured. This figure was specifically chosen since many sign language gestures are similar in nature (the distinction between 'M', 'N', and 'T' is very minor). It is believed that this will allow the model to be able to detect these gestures by capturing sufficient variance.

**Direct Landmark Recording:** Using the MediaPipe library, the  $(x, y, z)$  coordinates of the hand's landmarks were directly recorded and saved in a structured format (such as CSV) directly from the live feed. This ensured that the data that was captured matched perfectly with the environmental conditions and sensor characteristics that will be present at the time of actual deployment.

**Data Diversity:** There were several samples for each class, taken with different hand positions, distances, and small rotations, in order to increase the generalization

capability of the model, where there are a total of about 1,300 to 1,400 samples.

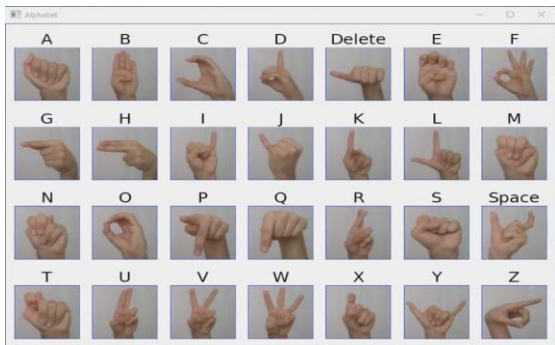


Fig -2: Alphabet Reference

### 3.2 Skeleton Tracking and Hand Landmark Model (The Core Component)

The core of the gesture recognition system is the MediaPipe Hands skeletonization model. This component is the primary engine that transforms raw video frames into a structured geometric representation of the hand.

#### 3.2.1 MediaPipe Skeleton Architecture

MediaPipe employs a machine learning pipeline to achieve high-fidelity hand tracking. In each video frame, it locates a total of 21 hand landmarks (key points), which represent the "skeleton" of the hand. The hand landmarks are numbered as follows:

Wrist (0): The base of the hand.

Thumb (1-4): Four landmarks for the CMC, MCP, IP, and tip of the thumb.

Index Finger (5-8): Four landmarks for the MCP, PIP, DIP, and tip of the index finger.

Middle Finger (9-12): Four landmarks for the MCP, PIP, DIP, and tip.

Ring Finger (13-16): Four landmarks for the MCP, PIP, DIP, and tip.

Pinky Finger (17-20): Four landmarks for the MCP, PIP, DIP, and tip.

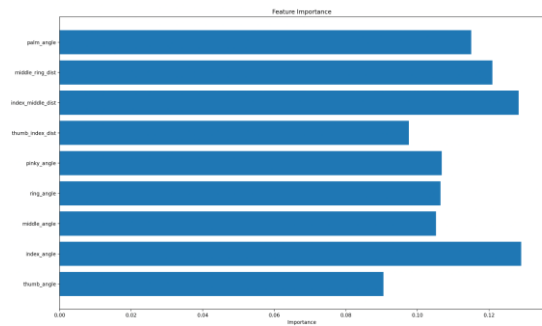


Fig -3: Feature Importance chart

#### 3.2.2 3D Landmark Representation

In each of the landmarks, the coordinates  $(x, y, z)$  are obtained. Here, the coordinates  $(x, y)$  represent the landmark position in the image space and are normalized in the range  $[0.0, 1.0]$  based on the image width and height. In the same way, the  $z$  coordinate represents the landmark depth with the wrist as the reference point. This 3D skeletal model makes the system robust with the changes in the size of the hand (depth) and rotation using a threshold of 0.3 for precision.

#### 3.3 Feature Vector Generation and Serialization

The extracted 21 landmarks were converted into a 1D feature vector for the input of the model:

Normalization: Normalization was performed by normalizing the coordinates relative to a reference point (such as the wrist landmark) to ensure scale and translation invariance.

Flattening: The 21 landmarks were flattened into a 63-dimensional feature vector, where each landmark has 3 coordinates.

Serialization with Joblib: Joblib was used to serialize the trained machine learning model for efficient deployment in real-time inference without the need for re-training the model.

#### 3.4 Model Training and Classification

The supervised learning Random Forest classifier was trained on the self-collected real-time dataset.

Training Process: The labeled feature vectors were split into a set for training and a set for testing. The classifier was trained to match the 63-dimensional feature vectors to the corresponding alphabet, number, or word.

Optimization: The training was optimized to ensure that the classifier minimizes the classification error for visually similar signs, such as 'A', 'S', and 'M', by

analyzing the distributions of the features collected in the real-time phase. It was observed that the classifier has a perfect accuracy and F1-score of 1.00 on the test set.

### 3.5 Alphabet-to-Sentence Formation Logic

The system uses a temporal logic to make the predictions on individual gestures into a coherent sentence. In the Character Recognition section, the system predicts the character in each frame with high confidence, stability buffer to avoid flickering and wrong key entries, where a character is "appended" to the sentence only if it is stable for a certain number of frames (e.g., 20-30 frames), and the alphabets are concatenated in real-time and displayed as a sentence on

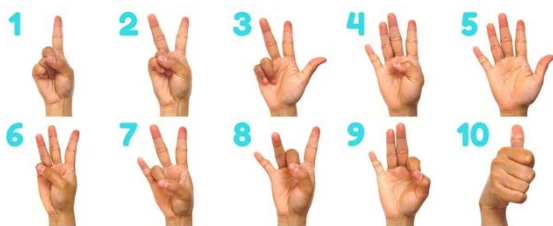


Fig -4: Number Reference

### 3.6 Text-to-Speech (TTS) Conversion

In order to bridge the gap between visual recognition and auditory communication, a Text-to-Speech module was incorporated. Once the sentence is finalized or a specific "Speak" command is invoked, the system uses the gTTS library to transform the text string into an audio stream. The audio stream is then played in real-time using the Pygame library. This dual form of output enables the system not only to assist the hearing-impaired but also to facilitate the communication needs of individuals with speech disabilities through vocalizing the gestures they make.

### 3.7 System Implementation

The proposed system will employ a variety of sophisticated technologies to ensure the efficiency of sign language recognition and interpretation in real time. The proposed system will be developed using the Python programming language, which can be easily adapted to accommodate different programming libraries. The proposed system will employ OpenCV to ensure the visualization of sign language. This ensures efficiency of the sign language in the proposed system. The proposed system will employ MediaPipe to ensure the accurate detection of the hand, which plays a critical role in recognizing the patterns in sign language. The proposed system will employ a classifier to recognize the alphabet in sign language. Jolib will be used to serialize the model, which ensures the model can be saved for future

use without having to train the model. The proposed system will also employ a sentence formation mechanism to ensure the interpretation of the alphabet to form meaningful words or sentences.

## IV. DISCUSSION

The explanation of the sign language illustration system also shows the practicality of using static hand gestures for interpretation. This is because the system can recognize all 26 alphabets and also include space and delete commands. This is important because it ensures that there is no interruption in communication. This is usually a problem in such systems.

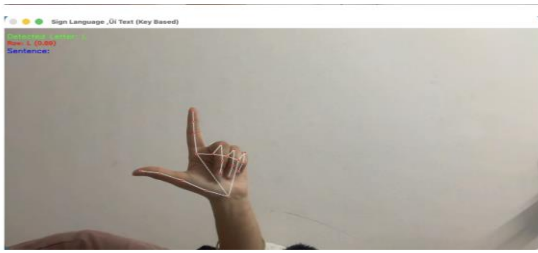
The use of Mediapipe for hand landmark detection is also important because it ensures that there is precise detection of hand gestures. This is also because Jolib is used for model management. This shows that it is important to ensure that there is proper management of models to ensure that they are efficient.

The accuracy of the model in distinguishing between static gestures also shows that it can recognize slight variations in hand shapes. This is important because it ensures that there is no error in recognizing alphabets. However, the limitations that were experienced in distinguishing between gestures also show that there is a problem with static hand gestures. These problems indicate that, even though the system has a firm base in static recognition, the addition of dynamic gesture analysis could enhance the robustness of the system in real-world applications.

Moreover, the addition of the space and delete commands not only improves the sentence construction but also demonstrates a user-centric design philosophy by addressing the needs that a user may have in a system. This addition improves the usability and practicality of the system, making it more accessible to the end user, who may be in need of the sign language for communication.

When comparing the proposed system to existing sign language recognition technology, it demonstrates a much simpler solution that could be scaled up for greater functionality. The emphasis on static gestures makes the system easier to deploy, which would be advantageous in a wider scope of applications.





**Fig -7: Letter Detection**

Therefore, the project has successfully established a framework for the automation of sign language interpretation using static gesture recognition. The addition of the space command as well as the delete command has made the interface more natural, hence making it more effective in the communication process. This is a stepping stone towards further improvements in the same area, which could lead to the enhancement of accessibility for the hearing-impaired.

## VI. CONCLUSION

The sign language illustration system developed in this project has successfully demonstrated its capability to recognize and translate static hand gestures into the corresponding alphabets, as well as form meaningful sentences. This has been achieved through the effective

## REFERENCES

- [1] M. A. Abdel-Fattah, "Arabic sign language: A perspective," *Journal of Deaf Studies and Deaf Education*, vol. 10, no. 2, pp. 212–221, Apr. 2005.
- [2] L. Lee, "The Importance of Learning Deaf Culture through a Black Deaf Perspective in the Field of Communication Sciences and Disorders," *Research Report*, 2022.
- [3] S. Colibaba, I. Gheorghiu, A. Colibaba, O. Ursa, C. Antonic, and R. Cirmari, "The Voice Project: Habilitating Hearing-Impaired Children to Recover Hearing and Lead a Normal Life," in *2022 E-Health and Bioengineering Conference (EHB)*, IEEE, 2022, pp. 1–4.
- [4] Y. Xusnora and A. Yulduz, "Ways to develop vocabulary in children with hearing impairment," in *E Conference Zone*, 2022, pp. 229–230.
- [5] Y. Farhan, A. A. Madi, A. Ryahi, and F. Derwich, "American Sign Language: Detection and Automatic Text Generation," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, IEEE, 2022, pp. 1–6.
- [6] S. H. Hamerdinger and C. J. Crump, "Sign language interpreters and clinicians working together in mental

utilization of Python programming libraries like Mediapipe to ensure accurate hand landmark detection and Joblib to efficiently handle model management. Through these developments, the system has been able to recognize all the 26 alphabets, as well as vital control commands like space and delete. This has significantly enhanced the naturalness of the sign language illustration system, making it a very effective tool for communication.

The high accuracy and low latency observed in the performance of the sign language illustration system have significantly demonstrated its potential for use in real-world applications, especially in helping the hearing-impaired community. Although there are still a few challenges to be addressed, especially in dealing with ambiguous hand gestures, the foundation established in this work has laid a very effective platform for future developments.

On the whole, this project makes a significant contribution to the development of automated sign language interpretation, which will go a long way in enhancing accessibility as well as effective communication for individuals who use sign language as a means of communication.

- health settings," *Routledge Handbook of Sign Language Translation and Interpretation*, 2022.
- [7] A. S. Dhanjal and W. Singh, "An optimized machine translation technique for multi-lingual speech to sign language notation," *Multimedia Tools and Applications*, vol. 81, no. 17, pp. 24099–24117, 2022.
- [8] S. A. Khan, "Importance of hearing and hearing loss treatment & recovery," *IJSA*, vol. 3, no. 1, pp. 14–16, 2022.
- [9] K. Aashritha and V. M. Manikandan, "Assistive Technology for Blind and Deaf People: A Case Study," in *Machine Vision and Augmented Intelligence: Select Proceedings of MAI 2022*, Springer, 2023, pp. 539–551.
- [10] J. M. Power, G. W. Grimm, and J.-M. List, "Evolutionary dynamics in the dispersal of sign languages," *Royal Society Open Science*, vol. 7, no. 1, Jan. 2020, Art. no. 191100.
- [11] U. Farooq, M. S. M. Rahim, N. Sabir, A. Hussain, and A. Abid, "Advances in machine translation for sign language: Approaches, limitations, and challenges," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14357–14399, Nov. 2021.

- [12] M. Raja'a, M. Mohammed, and S. M. Kadhem, "Automatic translation from Iraqi sign language to Arabic text or speech using CNN," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, vol. 23, pp. 112–124, Jun. 2023.
- [13] M. Mukushev, A. Sabyrov, A. Imashev, K. Koishybay, V. Kimmelman, and A. Sandygulova, "Evaluation of manual and non-manual components for sign language recognition," in *Proceedings of the International Conference on Language Resources and Evaluation*, 2020, pp. 6073–6078.
- [14] K. Wong, R. Dornberger, and T. Hanne, "An analysis of weight initialization methods in connection with different activation functions for feedforward neural networks," *Evolutionary Intelligence*, vol. 17, no. 3, pp. 2081–2089, Jun. 2024.
- [15] H. Luqman and E.-S.-M. El-Alfy, "Towards hybrid multimodal manual and non-manual Arabic sign language recognition: MArSL database and pilot study," *Electronics*, vol. 10, no. 14, p. 1739, Jul. 2021.
- [16] B. Fang, J. Co, and M. Zhang, "DeepASL: Enabling ubiquitous and nonintrusive word and sentence-level sign language translation," in *Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*, Nov. 2017, pp. 1–13.
- [17] H. Kaur and J. Rani, "A review: Study of various techniques of Hand gesture recognition," in *Proceedings of the 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India, Jul. 4–6, 2016, pp. 1–5.
- [18] J. S. Sonkusare, N. B. Chopade, R. Sor, and S. L. Tade, "A review on hand gesture recognition system," in *Proceedings of the 2015 International Conference on Computing Communication Control and Automation*, Pune, India, Feb. 26–27, 2015, pp. 790–794.
- [19] G. R. S. Murthy and R. S. Jadon, "A review of vision based hand gestures recognition," *International Journal of Information Technology and Knowledge Management*, vol. 2, pp. 405–410, 2009.
- [20] P. Garg, N. Aggarwal, and S. Sofat, "Vision based hand gesture recognition," *World Academy of Science, Engineering and Technology*, vol. 49, pp. 972–977, 2009.
- [21] M. Oudah, A. Al-Naji, and J. Chahl, "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, 2020.
- [22] A. Osman Hashi, S. Zaiton Mohd Hashim, and A. Bte Asamah, "A Systematic Review of Hand Gesture Recognition: An Update From 2018 to 2024," *IEEE Access*, vol. 12, pp. 143599–143626, 2024.