Sign Language Detection Using Python, OpenCV and Deep Learning

[1] P.TULASI, [2] P.SHARVANI, [3] P.ALPHONES ABHILASH, [4] V.NANDINI,[5] G.RAHUL
[1]Assistant Professor, Department of CS, Sri Indu College Of Engineering & Technology, Hyderabad.
[2,3,4,5] U.G. Scholar, Department of CS, Sri Indu College Of Engineering & Technology, Hyderabad.

***Abstract*- An effective Indian Sign Language (ISL) translation system has been designed to assist individuals with communication challenges, particularly those who are deaf and mute. The system utilizes OpenCV along with advanced computer vision techniques and neural networks to enable real-time conversion of ISL gestures into both textual and spoken outputs, thereby improving communication and social interaction.**

The proposed solution adopts a two-stage architecture that integrates a Convolutional Neural Network (CNN) for accurate gesture recognition. Extensive analysis of ISL gestures, combined with advanced deep learning methods, enhances the precision of the system. A custom dataset is created using OpenCV and used to train the CNN model, which incorporates layers such as ReLU activation, max pooling, dropout, and optimization through the Adam optimizer. One of the key features of this system is its ability to construct meaningful sentences from finger-spelling inputs. Additionally, it includes an automatic correction mechanism that refines the generated text, improving both accuracy and usability. These enhancements contribute to a more efficient and user-friendly experience.

Overall, this project represents a significant step toward reducing communication barriers for the deaf community. By leveraging modern technologies, it promotes inclusivity and provides a practical solution for real-time sign language interpretation.

***Keywords:* Indian Sign Language, Deaf and Mute, Convolutional Neural Networks, OpenCV, Computer Vision, Deep learning, Sign to text and speech, fingerspelling, Hand Gesture Recognition.**

I. INTRODUCTION

Indian Sign Language (ISL) is the dominant form of communication for people with communication disabilities, often referred to as deaf and mute (DandM). Since their primary disability is related to communication and spoken languages may not be available to them, ISL becomes an important tool to communicate thoughts and messages. DandM's human communication is facilitated by a rich visual language that includes hand gestures, body movements, facial expressions, and lip patterns. Contrary to popular belief, sign languages are not universal, but unique to certain communities and regions. In response to the communication barriers faced by DandM individuals, efforts have been directed towards the development of sign language interpretation systems. These systems aim to facilitate seamless communication and provide a natural mode of interaction for the hearing impaired.

The components of sign language include fingerspelling, word-level sign vocabulary, and non-manual signals like facial expressions. The main objective of is to develop a user-friendly interface that can detect and translate these elements of ISL with high accuracy. This initiative not only aims to bridge existing communication gaps but also aspires to establish a new standard in efficient ISL translation solutions, promoting greater inclusivity. Creating a Sign-to-Text Language Translator using OpenCV technology, aimed at converting ISL gestures into readable text and audible speech. This development is particularly significant due to the current scarcity of accessible tools for interpreting ISL, which creates barriers for those unfamiliar with sign language. By leveraging advanced techniques in computer vision and deep learning, the system will provide real-time interpretation of ISL, enhancing communication accessibility for the hearing impaired.

II. LITERATURE SURVEY

As an initial step, a comprehensive review of the existing literature was conducted.

Rachana Patil1,,Vivek Patil1, Abhishek Bahuguna[1]Indian Sign Language Recognition using Convolutional Neural Network This paper introduces a cutting-edge system designed for Indian Sign Language gesture recognition, utilizing a standard web camera and sophisticated image processing techniques like grayscale conversion and dilation to enhance detection. At its core, a Convolutional Neural Network (CNN) trained on a robust dataset, achieves approximately 95% accuracy in gesture recognition. Future enhancements aim to enable bidirectional communication, translating both sign language and spoken language. This innovation promises to significantly improve inclusivity and break down communication barriers, marking a significant step forward in human-computer interaction and universal accessibility.

Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen[2] Sign Language Recognition Using Convolutional Neural Networks: Sign language recognition (SLR) systems, crucial for bridging communication gaps between deaf and hearing communities, have advanced through technology such as Convolutional Neural Networks (CNNs) and Hidden Markov Models (HMMs). CNNs are highly effective for SLR due to their ability to automatically extract and learn spatial features from video, making them robust against variations like lighting and background. In contrast, HMMs excel in modeling sequential data through temporal sequences but rely on hand-crafted features, which can be limiting. Both methods have distinct advantages: CNNs for spatial feature learning and automatic extraction, and HMMs for efficient sequential data modeling.

Muhammad Usman,Hira Hameed, Ahsen Tahir[3] Recognizing British Sign Language Using Deep Learning: Contactless and Privacy-Preserving Approach .The proposed British Sign Language (BSL) recognition framework includes three stages: data curation, signal processing, and deep learning classification. Initially, an extensive dataset of annotated BSL signs was developed, providing essential training data. Next, signal processing techniques transformed these signs into spectrograms, capturing key frequency features crucial for analysis. Finally, deep learning models like GoogleNet, SqueezeNet, and VGGNet, combined with WB radar sensors, classified the BSL gestures. Evaluations showed a high classification accuracy of 90.07%, with VGGNet performing best, demonstrating the framework's capability to accurately recognize diverse BSL gestures across various user demographics.

III. PROBLEM STATEMENT

Sign-to-text and speech translation technology is crucial for enhancing communication accessibility for the deaf-mute community, who rely on sign language. This technology acts as a bridge, allowing them to effectively communicate with non-sign language users, thereby overcoming significant barriers in social, educational, and professional settings. Real-time interpretation of sign language into text and speech facilitates smoother interactions in various real-world scenarios, proving essential for effective communication.

Despite technological advancements, communication challenges persist for those using Indian Sign Language (ISL). Current translation technologies often suffer from inefficiencies and inaccuracies, leading to misunderstandings. There is a pressing need for an efficient and accurate Sign-to-Text Language Translator that utilizes OpenCV, Computer Vision, and Neural Networks for real-time, precise ISL gesture interpretation. This project aims to develop such a system, incorporating advanced features like finger spelling and AutoCorrect mechanisms, enhancing functionality, and promoting broader societal inclusivity and accessibility.

3.1 Existing Systems

The Existing systems face significant challenges in recognizing abnormal sign language images effectively, particularly when the images vary in angles and distances. This difficulty arises from inherent complexities and dataset limitations. Another challenge is extending the recognition capabilities to include English alphabets and numeric alongside traditional sign language gestures, which demands considerable computational resources. Additionally, these systems have a limited ability to effectively recognize and interpret complex signs and lengthy messages, constrained by the model architecture and the size of the datasets used. Furthermore, the effectiveness of Hidden Markov Models (HMMs) used in some systems is heavily dependent on the quality of hand-crafted features. This feature engineering process is time-consuming and highly specific to the domain, limiting the flexibility and scalability of the system. These limitations highlight the need for ongoing research and development to enhance the functionality and inclusivity of sign language recognition systems.

3.2 Proposed System

The proposed system, which utilizes a sign language to text and speech conversion approach based on OpenCV and Convolutional Neural Networks (CNN), offers a groundbreaking solution to overcome communication barriers between the deaf or hard of hearing and those who do not use sign language. By integrating OpenCV, the system gains access to a suite of advanced image processing tools that enhance the accuracy of gesture capture through refined feature extraction and image manipulation techniques. Concurrently, CNNs, with their ability to learn complex hierarchical data representations, are particularly adept at recognizing and interpreting the nuanced patterns of sign language gestures. This combination not only facilitates precise gesture recognition but also ensures the translation into text and speech is executed in real-time, significantly improving communication flow. The system's capacity for real-time processing is pivotal in real-world interactions, enabling spontaneous and natural communication that can greatly benefit educational settings, professional environments, and casual social exchanges. Through this technology, we can bridge the gap in communication, fostering inclusivity and understanding across different communities.

Real-time processing is crucial for facilitating seamless communication in diverse settings by providing instant sign language interpretation and feedback. This immediacy is critical in educational, professional, and social contexts, preventing misunderstandings and fostering smoother exchanges. Moreover, the flexibility of OpenCV and CNNs enhances the system's effectiveness. These technologies allow for tailored adjustments in image processing and neural network structures to suit various sign language dialects and individual user needs, significantly boosting the system's adaptability and overall functionality.

Furthermore, this technology not only facilitates more effective communication for deaf individuals but also fosters inclusivity, allowing non-sign language users to engage more fully with the deaf community. The application of Gaussian Blur and Media Pipe in this approach enhances feature extraction, aiding in the development of a more refined and accessible communication tool. This approach demonstrates significant potential in creating a more inclusive society by addressing critical communication barriers.

IV. SPECIFICATIONS

The specific requirements and specifications for the project, detailing both functional and non-functional aspects. It aims to clarify the expectations and criteria for the proposed system, ensuring it meets the necessary standards for performance, usability, and reliability.

The proposed model is designed to meet specific functional and non-functional requirements, enhancing its efficacy and user satisfaction. Functionally, it requires robust data pre-processing, training on diverse internet-sourced images, and the capability to transform old, damaged images into high-quality digital outputs. Interoperability across different platforms is also essential. Non-functionally, the model must ensure high performance and scalability, user-friendly interfaces, and high portability and compatibility. Additionally, it needs to be reliable, easy to maintain, and uphold strong data integrity to preserve the authenticity and quality of the digital images processed, ensuring a system that is both efficient and effective.

Software Specifications

- Python (3.7.4)
- IDE (VSCode)
- OpenCV (CV2 Version 3.4.2)
- Mediapipe

Hardware Specifications

- Processor: Intel core i3
- RAM : 4 GB (min)
- Hard Disk: 100 GB

V. SYSTEM DESIGN AND ANALYSIS

Sign language to text and speech conversion using OpenCV and Convolutional Neural Networks (CNN) presents a promising approach to bridge communication barriers between individuals who are deaf or hard of hearing and those who are not fluent in sign language. The appropriateness of this approach stems from its ability to leverage advanced computer vision techniques and deep learning algorithms to accurately interpret sign language gestures and convert them into text and speech in real-time.

The utilization of OpenCV and CNNs represents a significant advancement in the field of sign language recognition and translation. OpenCV provides robust tools and libraries for image processing and computer vision, enabling efficient extraction of features from sign language gestures. CNNs, on the other hand, excel at learning hierarchical representations of data, making them well-suited for tasks like gesture recognition. By harnessing these cutting-edge technologies, the sign language to text and speech conversion approach achieves high levels of accuracy and reliability, thereby enhancing communication accessibility for individuals who rely on sign language.

One of the key strengths of the proposed approach is its ability to perform sign language recognition and translation in real-time. This capability is essential for facilitating seamless and natural communication between sign language users and non-sign language users in various settings, such as educational institutions, workplaces, and public spaces. Real-time processing ensures timely interpretation of sign language gestures, enabling immediate feedback and responses, which is crucial for effective communication.

The architecture is such a way , the input live capture of the hand is passed to OpenCV's mediapipe landmark function giving a Skelton hand . OpenCV, coupled with MediaPipe's Hand Landmark function, offers a robust solution for real-time hand gesture recognition. MediaPipe employs a sophisticated machine learning model that identifies 21 specific landmarks across the hand, including joints and fingertips. The process begins with video input, either from a live feed or a recording. MediaPipe first detects the hand using a pre-trained palm detection model. Once detected, it focuses on a cropped region to pinpoint these landmarks with high precision. OpenCV then takes over to process this landmark data for various applications, such as translating sign language into text or controlling interfaces through gestures. The integration of these technologies allows for the visualization of these landmarks on the video itself, facilitating immediate feedback and interaction. This synergy between OpenCV and MediaPipe's capabilities significantly enhances the accessibility and usability of gesture-based communication and control systems.

A python library Hunspell suggest is used to suggest correct alternatives for each (incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence. This helps in reducing mistakes committed in spellings and assists in predicting complex words. The speak fun method is a part of a class, designed to utilize a text-to-speech engine to audibly render a string. Within the method, the self.speak_engine.say() function is called with the string self.str as its argument. This action queues the string for playback by the text-to-speech engine. Following this, the method invokes self.speak_engine.runAndWait(), which executes the queued speech, ensuring that the string is spoken synchronously. Incorporating the speak fun method into sign language applications broadens the accessibility and usability, ensuring that users with varying needs and preferences can effectively engage with the content and interface. By offering both visual and auditory feedback, these applications promote inclusivity and facilitate effective communication for all users.

Perhaps the most significant aspect of the sign language to text and speech conversion approach is its potential to enhance accessibility and inclusivity for individuals who are deaf or hard of hearing. By providing a means to translate sign language gestures into text and speech, the approach empowers deaf individuals to communicate more effectively with a broader range of people. It also promotes inclusivity by enabling non-sign language users to understand and interact with deaf individuals more easily, fostering a more inclusive and understanding society overall.

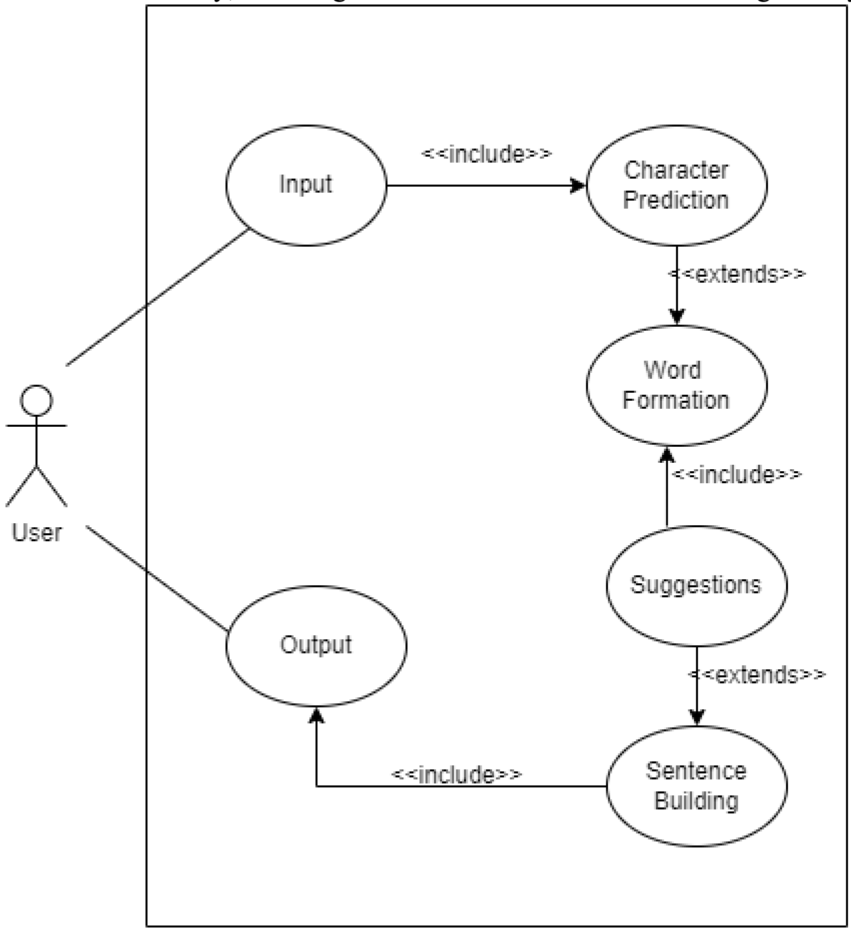

Fig.5.1 system design

The development of a sign language recognition system involves several critical procedures to ensure it functions accurately and efficiently. The first step is Data Collection and Preprocessing, where a diverse dataset of sign language gestures is gathered. This dataset must include a variety of hand shapes, movements, and orientations. To ensure consistency and quality, preprocessing steps such as standardizing lighting conditions, removing backgrounds, and normalizing hand sizes and positions are applied.

The second step involves Hand Gesture Recognition using OpenCV. This phase utilizes OpenCV for real-time hand detection and tracking. Techniques such as background subtraction, contour detection, and the use of convex hulls are employed to detect and track hand movements effectively. Relevant hand gestures are then recognized and extracted from the video stream using specialized algorithms.

Feature Extraction is the third step, where critical features from the hand region are identified. This includes pinpointing key hand landmarks like fingertips and the palm center using hand pose estimation libraries. Additionally, hand posture features such as orientation, the distance between fingertips, and angles between fingers are calculated. Visual features are also extracted using methods like histograms of oriented gradients (HOG) or local binary patterns (LBP).

The fourth step is the creation of a Deep Learning Model for Gesture Recognition. A deep learning model, typically a Convolutional Neural Network (CNN), is designed to recognize hand gestures from the extracted frames. The model is trained on the preprocessed dataset. Fine-tuning the model to improve its gesture recognition accuracy is a critical phase in this step.

Lastly, the Text Translation step converts recognized gestures into corresponding text and speech. A lookup table or a mapping system between gestures and their textual representations is used. To refine the translation process, natural language processing (NLP) techniques are explored to consider the context of gestures, thereby improving the accuracy and relevance of the translation. This comprehensive approach from data collection to translation ensures the system is robust and effective in
facilitating communication for the deaf and hard-of-hearing community.

VI. IMPLEMENTATION

6.1 Introduction to the Technologies Used

This Project leverages the combined strengths of Deep learning (DL) and OpenCV to create a sign-language translator. While both DL and OpenCV offer valuable functionalities, deep learning presents several key advantages in this specific context:

Deep Learning:

A subfield of Machine Learning takes a more advanced approach. It utilizes artificial neural networks with complex architectures that can automatically learn intricate patterns within data. This eliminates the need for extensive manual feature engineering required by traditional ML methods. Deep learning architectures like Convolutional Neural Networks (CNNs) excel at analyzing website screenshots, identifying suspicious design elements or low-quality visuals often associated with phishing attempts. Recurrent Neural Networks (RNNs), on the other hand, are adept at processing website text. They can analyze the context of surrounding words and identify suspicious language patterns often used in phishing attempts, such as urgency, unrealistic promises, or grammatical mistakes.

Convolutional Neural Networks:

Convolutional Neural Networks (CNN) play a crucial role in interpreting sign language to text and speech. The proposed method for interpreting sign language to text and speech uses a multi-headed CNN with two input data channels, one for processed images and the other for hand landmarks data . The processed images and hand landmarks data were trained separately using two different models before being combined in the CNN . The CNN model has two input data channels and one output channel. To improve the accuracy of the CNN model, several techniques were used, such as MaxPooling 2D, batch normalization, and dropout layers in both training sides. Two-dimensional Convolutional layers with specific filter size, kernel, and activation functions were also used in both image and hand landmarks training. The output dense layer of the CNN has 24 units with Softmax activation function . The classification layer of the CNN can complement a false result of one layer with the weight of the other layer, which can provide a positive outcome in interpreting sign language to text and speech . In addition, the proposed method is suitable for wild situations as it is not entirely dependent on hand position in an image frame. However, the effectiveness of the proposed method is dependent on the hand landmark extraction model used, and other hand landmark models can produce different results. It is important to note that the CNN model requires a good number of images for training, and raw image processing can be used to detect hand portions, which may increase recognition chance and reduce model training time. Despite this, the proposed method successfully improved the final validation and test results in interpreting sign language to text and speech.

OpenCV:

OpenCV has emerged as a powerful tool for hand gesture recognition in sign language. Hand gestures are an integral part of communication through sign language, and several methodologies for motion discovery can be used with OpenCV, such as the dynamic vision sensor (DVS) . There are several approaches proposed in the literature for hand gesture recognition in sign language using OpenCV. For example, Jun Haeng Lee et al. proposed a motion classification method with two DVSs to get a stereo-vision system for hand gesture recognition in sign language. The proposed methodology involves processing hand images using two processing techniques and creating two data channels. Similarly, Arnon et al. presented an event-based gesture recognition system that uses a temporal filter cascade to create spatio-temporal frames that CNN executes in the event-based processor. OpenCV can help achieve high accuracy in recognizing hand gestures in sign language, such as achieving a validation accuracy of 98.98% and test accuracy of 98.981% in detecting Indian Sign Language (ISL) gestures using the Finger Spelling dataset. OpenCV can identify image districts compared to human skin by binarizing the input image with a proper threshold value, which can be used for hand detection and landmark extraction. Additionally, OpenCV can be utilized for keyframe choice in sign language recognition by analyzing hand gesture image sequences using OpenCV. However, it is not always certain that OpenCV works with hand gestures as it detected face first and then body movement . Regularizing the yields with high-level features can improve the performance of the models for hand gesture recognition in sign language. Several deep learning models have also been proposed for hand gesture recognition in sign language, such as 3D CNNs and Faster Region-based Convolutional Neural Network (Faster-RCNN) models, which perform 3D convolution in the convolutional layers and use a model for hand recognition in the data image, respectively.

Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It is high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python plays a pivotal role in developing a Sign Language Detection Using Python, OpenCV and Deep Learning. It serves as the primary language for implementing deep learning models, leveraging frameworks like TensorFlow, Keras, or PyTorch for designing complex architectures. Python handles data preprocessing tasks using libraries like Pandas and NumPy. With Python, one can train and evaluate deep neural network models.

6.2 Packages:

NumPy:

NumPy is a fundamental library in Python for numerical and scientific computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy is indispensable for handling numerical operations which are critical in processing and transforming images of sign language gestures into a format suitable for machine learning models. NumPy's robust handling of large, multi-dimensional arrays and matrices allows for efficient manipulation of pixel data from images, essential for preprocessing steps like normalization and dimensionality reduction before they are fed into neural networks.

Keras:

Keras is a high-level deep learning library for Python. It's built on top of lower- level deep learning frameworks like TensorFlow and Theano. Keras simplifies the process of creating and training artificial neural networks, making it accessible for both beginners and experts. Keras serves as a high-level interface for neural network architecture design and training. It abstracts many complexities associated with building neural networks, making it simpler to experiment with different architectures, optimize performance, and achieve faster prototyping. In a sign language project, Keras can be used to build and train deep learning models that recognize and classify different hand gestures based on the input data processed by NumPy.

Tensor Flow:

Integrate TensorFlow, a leading deep learning framework, to empower the project with state-of-the-art machine learning capabilities, facilitating tasks such as neural network modeling, training, and inference. Leveraging TensorFlow within the software requirements ensures scalability, efficiency, and compatibility with cutting-edge machine learning algorithms, advancing the project's capabilities in data analysis, pattern recognition, and predictive modeling. TensorFlow is used for its extensive machine learning capabilities, particularly in areas requiring high-performance numerical computation. TensorFlow's flexibility and scalability in neural network training and inference make it ideal for developing sophisticated models that can accurately translate sign language gestures into text and speech. Its robust ecosystem supports detailed performance analysis and optimization, ensuring the model runs efficiently even on large datasets.

Mediapipe:

MediaPipe, a powerful framework for building cross-platform AI solutions, into the project's software requirements. By utilizing MediaPipe, the project gains access to pre-trained models and pipelines for various tasks such as pose estimation, hand tracking, and facial recognition. Leveraging MediaPipe's modular and efficient architecture ensures seamless integration and robust performance, enhancing the project's capabilities in real-time perception and understanding of visual data. MediaPipe offers ready-to-use tools for complex tasks such as hand tracking and gesture recognition that are central to a sign language recognition system. By integrating MediaPipe, the project benefits from advanced computer vision techniques that can detect and interpret hand movements and positions in real-time, which is crucial for translating sign language accurately and promptly.

Hunspell:

Hunspell, a widely-used spellchecking and morphological analysis library, into the project's software requirements. By leveraging Hunspell, the project gains access to comprehensive linguistic analysis tools, including spell checking, hyphenation, and stemming. Utilizing Hunspell ensures accurate language processing and improves the overall quality of textual content within the application. Its open-source nature and extensive language support make it an ideal choice for multilingual applications. Hunspell provides tools for spell checking, hyphenation, and stemming which are vital for ensuring the accuracy of the text generated from sign language gestures.

PyEnchant:

PyEnchant, a powerful spellchecking library, into the project's software requirements. By utilizing PyEnchant, the project gains access to robust spell checking capabilities with support for multiple languages and dictionaries. Leveraging PyEnchant ensures accurate detection and correction of spelling errors within textual content, enhancing the overall quality and readability of the application. Its ease of use and compatibility with various Python environments make it an ideal choice for implementing spell checking functionality. PyEnchant enhances the text output by checking and correcting spelling errors, thus improving the readability and quality of the translated text. These tools are especially useful in maintaining linguistic integrity when the system is used in educational or professional settings.

Operating System:

Provides functions for interacting with operating systems like file paths. Used for handling audio file directory paths. Operating System integration is crucial for managing system-level tasks such as file handling, which is necessary for loading the datasets, accessing pre-trained models, and storing outputs. Efficient handling of these tasks ensures that the system operates smoothly across different operating environments, supporting the portability and usability of the application.

VII. RESULTS

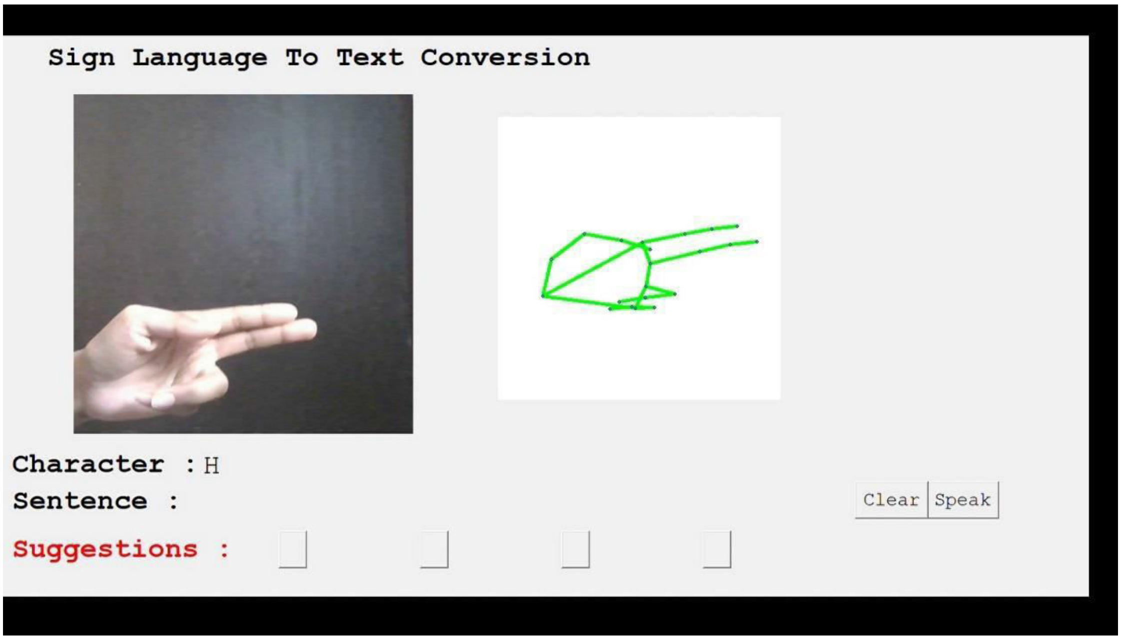

Fig 7.1 Hand gesture representing H

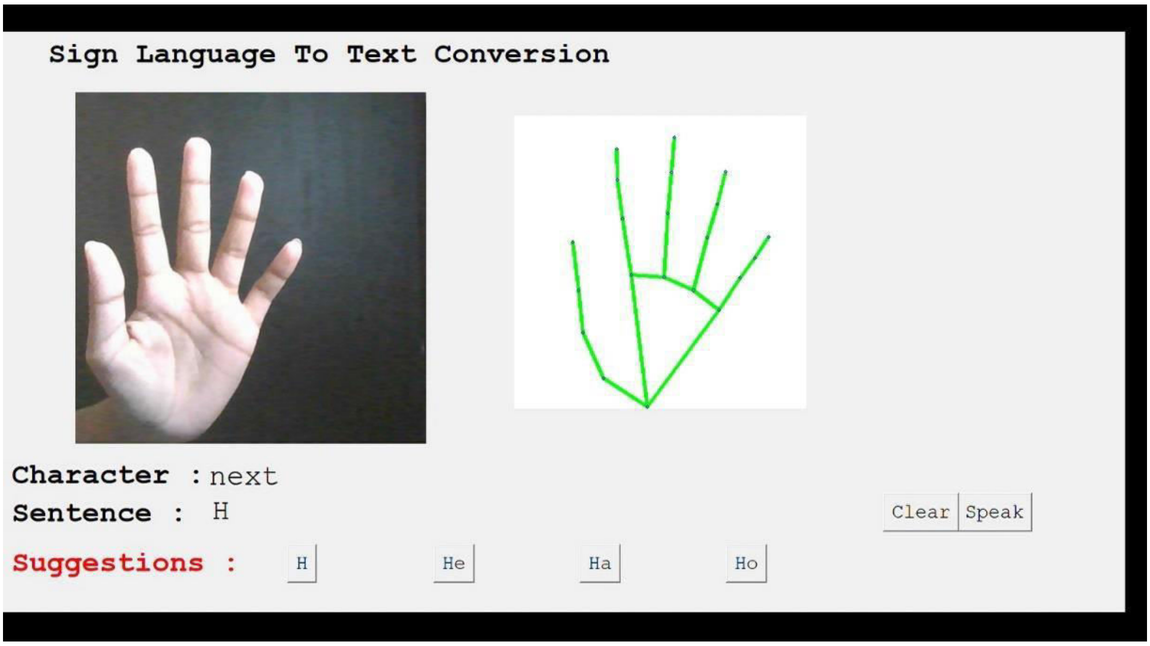

Fig 7.2 Hand gesture to take Next letter

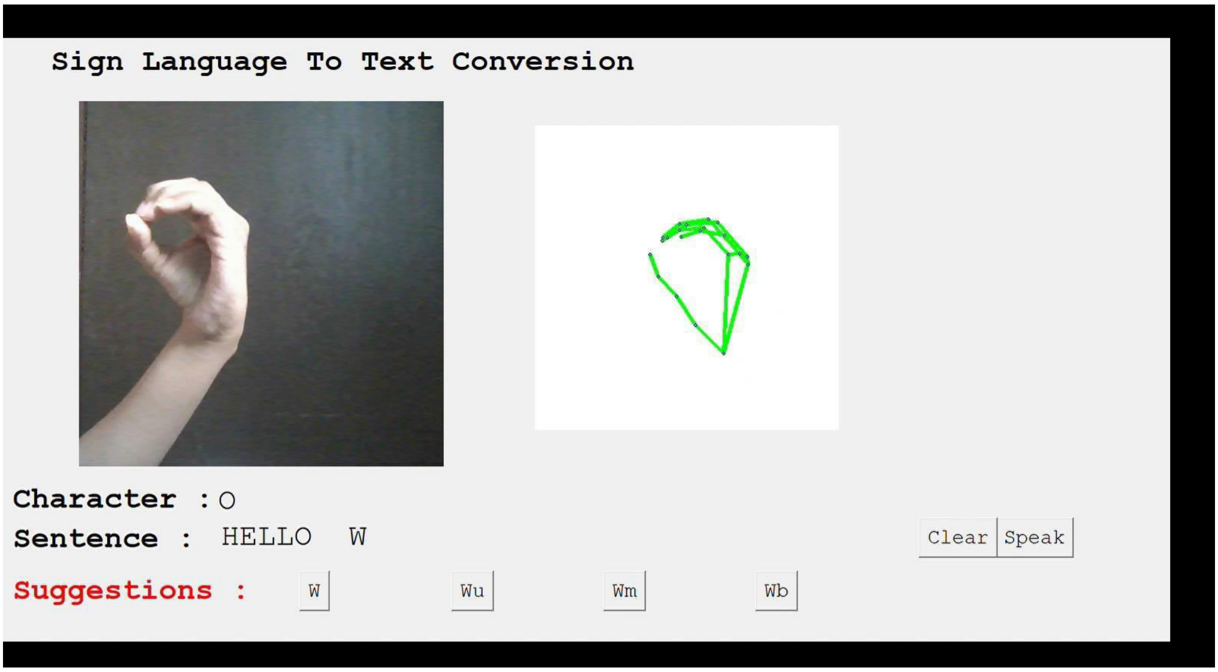

Fig 7.3 sentence forming

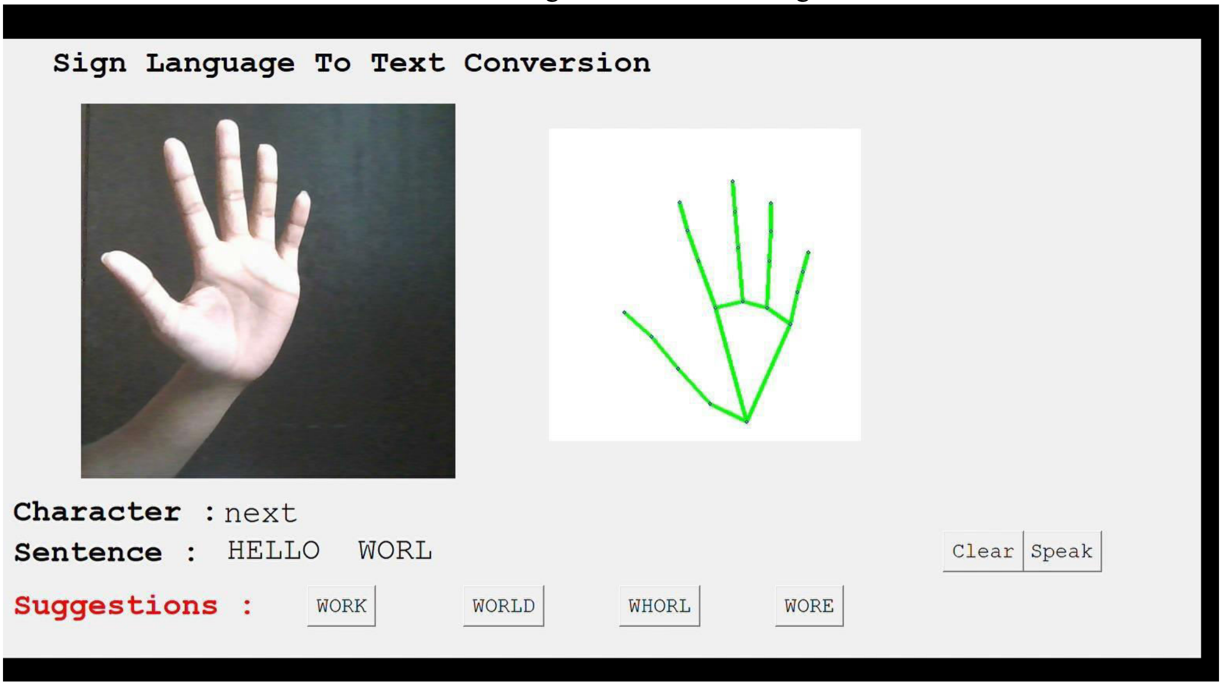

Fig 7.4-word suggestions

VIII. CONCLUSION

In conclusion, our project has successfully developed a real-time vision-based Indian Sign Language (ISL) recognition system for the Deaf and Mute (D&M) community, achieving a final accuracy of **98.0%** on our dataset. The incorporation of two layers of algorithms significantly improved predictions, particularly for symbols that exhibit similarities. This enhancement empowers the system to detect nearly all symbols, contingent upon proper display, absence of background noise, and adequate lighting conditions.

The gesture classification process involved leveraging OpenCV for capturing frames, applying a Gaussian blur filter, and thresholding. The processed images were then fed into a Convolutional Neural Network (CNN) model, trained on our custom dataset, facilitating effective feature extraction and accurate gesture recognition.

In essence, the project's design and implementation align seamlessly with our stated objective, yielding a promising accuracy rate. Beyond its technical achievements, our work underscores the potential of vision-based sign language recognition systems in overcoming communication barriers between the hearing-impaired and hearing individuals within the context of Indian Sign Language. The successful fusion of innovation, user-centric design, and cultural adaptability positions our system as a valuable tool in fostering inclusivity and communication accessibility for the Indian Sign Language community.

REFERENCES:

[1] H. Rachana Patil1,Vivek Patil1,Abhishek Bahuguna1,and Mr. Gaurav Datkhile," Indian Sign Language Recognition using Convolutional Neural Network" (2023) itmconf/20214003004

[2] Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen Sign Language Recognition Using Convolutional Neural Networks ECCV Workshops, Part I, LNCS 8925, pp. 572–578, 2015. DOI: 10.1007/978-3-319-16178-5 4

[3] Hameed et al.: recognizing bsl using dl: a contactless and privacy-preserving approach(2023) ieee ep/t021020/1.

[4] Srinivasan R, R Kavita & M Kavitha, Basetty Mallikarjuna, Sandeep Bhatia "Python And Opencv For Sign Language Recognition",(2023) International Conference on Device Intelligence | 978-1-6654-7491-7/23

[5] S. B. Abdullahi, K. Chamnongthai: IDF-Sign: "Addressing Inconsistent Depth Features for Dynamic Sign Word Recognition"(2023).IEEEAccess 023.3305255

[6] Russo, Michele. "AR in the Architecture Domain: State of the Art." Applied Sciences 11.15 (2021): 6800.

[7] Gao, L., Li, H., Liu, Z., Liu, Z., Wan, L., & Feng, W. (2021). RNN-transducer based chinese sign language recognition. Neurocomputing, 434, 45–54. https://doi.org/ 10.1016/j.neucom.2020.12.006.

[8] Sharma, A., Mittal, A., Singh, S. & Awatramani, V. 'Hand gesture recognition using image processing and feature extraction techniques'. Proc. Comput. Sci. 173, 181–190. https://doi.org/10.1016/j.procs.2020.06.022 (2020).

[9] Abiyev, R. H., Arslan, M., & Idoko, J. B. (2020). 'Sign language translation using deep convolutional neural networks'. KSII Transactions on Internet and Information Systems, 14(2), 631–653. https://doi.org/10.3837/tiis.2020.02.009.