Advanced Bilateral Communication System for Deaf and Mute Using Machine Learning

Aditi Kini[1], Akshata Gudekar[1], Bhushan Jadhav[1], Shruti Gurav[1], Dr. Manoj Patil[2]

[1] Student, Department of Computer Engineering, Rajiv Gandhi Institute of Technology, Mumbai, Mumbai University(India)
[2] Associate Professor, Department of Computer Engineering, Rajiv Gandhi Institute of Technology, Mumbai, Mumbai University(India)

ABSTRACT
Deaf and mute individuals are indispensable members of our society, deserving access to seamless communication platforms without requiring extensive training. While sign language serves as their primary means of communication, effective interaction often relies on others' proficiency in understanding sign language, which can be challenging for those unacquainted with it. To tackle this challenge, we propose a system designed to facilitate interaction between individuals who are Deaf and Mute (DnM) and the general population. This system equips non-DnM individuals with an audio interface, enabling the conversion of speech or text into videos. Through microphone input, speech from non-DnM individuals is transcribed into text, while the hand signals of Deaf and mute individuals are detected and processed using deep learning methods. In addition, supervised machine learning enables support for multiple languages, making the system adaptable for future language expansions. The system's modularity allows for continuous improvement, enabling the incorporation of additional data to support more languages over time. We anticipate that this system will empower Deaf and mute individuals to communicate effectively with others, thereby reinstating a sense of regularity in their daily interactions and promoting inclusivity within our society.

Keywords: Deaf and Mute(DnM), Indian Sign Language (ISL), American Sign Language (ASL), Machine Learning, Natural language processing, Convolutional neural network, Web speech API, Tokenization, Lemmatization, OPENCV, Stop words

1. INTRODUCTION

Sign language provides the predominant mode of communication for Deaf and mute individuals who utilize hand gestures, arm and body movements, and facial expressions as integral components of their communication. Worldwide, there exist 135 unique sign languages, encompassing American Sign Language (ASL), Indian Sign Language (ISL), British Sign Language (BSL), and Australian Sign Language (Auslan). In India, the Deaf and Mute community primarily uses Indian Sign Language (ISL) for communication. ISL possesses its phonology, syntax, and grammar, yet research on its philology and phonetic studies remains limited due to the scarcity of well-documented and annotated data. The adoption of ISL extends beyond the Deaf and Mute community, encompassing individuals with hearing impairments and educators involved in deaf education. Therefore, there is a pressing need to develop intelligent systems that facilitate seamless communication between Deaf and Mute (DnM) individuals and those who are not Deaf or Mute (NDnM). The overarching objective of such a system is to dismantle communication barriers by enabling the conversion of text or speech into sign language, and vice versa. Here, to accomplish this, Machine Learning (ML) and Deep Learning (DL) methodologies are employed for recognition tasks. ML algorithms utilize extracted features to categorize videos using supervised and unsupervised learning methodologies, while DL techniques serve for hand gesture detection and data processing. Developing an intelligent communication model capable of bridging the gap between DnM individuals and the broader society holds immense promise for societal integration and

inclusion, fostering greater participation and contributions from all individuals within the community. Data, predominantly focusing on American Sign Language (ASL) and Chinese Sign Language (CSL), pose a challenge for further advancements.

2. LITERATURE SURVEY

Title: ML-Based Sign Language Recognition System
Author: Amrutha K, Prabu P

The paper presents an ML-based automated Sign Language Recognition (SLR) system, emphasizing isolated hand gesture detection and recognition. Utilizing a convex hull for feature extraction and K-nearest neighbors (KNN) for classification, the system achieves a 65% accuracy rate, offering valuable assistance for the speech and hearing impaired. Challenges such as sudden movements and lighting variations undergo mitigation through preprocessing, segmentation, and feature extraction stages. Assessment of a dataset representing numbers 1-5 indicates potential enhancements through larger datasets and alternative classifiers to improve performance.

Title: A Review on Sign Language Recognition System Using ML & DL
Author: Soumen Das,Saroj Kr. Biswas, Manomita Chakraborty, Biswajit Purkayastha

The paper discusses the development of Sign Language Recognition Systems (SLRS) designed to assist individuals with hearing or speech impairments, tackling challenges such as accessibility and precision. Researchers have explored advanced machine learning (ML) and deep learning (DL) techniques for SLRS, encompassing data collection, clean-up, feature extraction, and classification. Deep learning methods like CNN and LSTM prove effective for handling complex visual challenges, yet concerns persist regarding variations in sign language and system reliability. Traditional methods (SVM, KNN, DT, RF) and modern approaches (CNN, Faster R-CNN, LSTM) undergo comparison, each presenting strengths and limitations. Limited diverse data, predominantly focusing on the American Sign Language (ASL) and Chinese Sign Language (CSL) pose a challenge for further advancements.

Title: Sign language conversion to text & speech
Author: Medhini Prabhakar, Prasad Hundekar, Sai Deepthi, Shivam Tiwari, Vinutha.

A novel system facilitates communication between individuals with hearing impairments and those without by translating sign actions into English text descriptions and speech. Utilizing Indian Sign Language alphabet samples for training, hand gestures are logged and classified using CNN, FRCNN, YOLO, and Media Pipe models. The system achieves satisfactory recognition accuracy, particularly with FRCNN, even in the face of longer computation times owing to hardware constraints. Media Pipe is the most efficient choice, offering real-time conversion capabilities without delays. Standardizing Indian Sign Language is crucial, given its diversity, with the system enhancing accessibility for deaf individuals and the general public. Future enhancements may focus on server-based systems and improved sign detection using high-range cameras.

Title: Audio to Sign Language Translation for Deaf People
Author: Ankita Harkude, Sarika Namade, Shefali Patil, Anita Morey

The paper introduces a system aimed at aiding communication for deaf individuals by translating audio messages into Indian Sign Language (ISL). It employs speech recognition and natural language processing (NLP) to convert audio input into text, then translates it into ISL using predefined gestures. The system's front end is designed with EasyGui, while PyAudio and Google Speech API process audio input. Implementation includes a block diagram illustrating audio processing and sign language conversion flow. The output comprises video clips displaying ISL words based on data generated from a predefined database. The system's versatility extends to government websites, form filling, and normal-deaf individual communication. Future enhancements may incorporate facial expressions and expand applications to include news channels for sign language interpretations.

3. PROBLEM STATEMENT

Communication challenges often arise when individuals with hearing and speaking impairments attempt to interact with those possessing full

hearing and speech capabilities. The primary obstacle in this scenario lies in the use of sign language by individuals with hearing impairments, which often remains unfamiliar to individuals without such impairments. To tackle this pervasive issue, a proposed system endeavors to bridge the communication gap and offer an innovative solution. This solution harnesses the power of video clips, machine learning algorithms, and natural language processing techniques to construct a bridge between these two distinct worlds, thereby facilitating more effective and inclusive communication channels. Through the integration of advanced technologies and methodologies, this system seeks to empower individuals with hearing and speaking impairments to communicate seamlessly with those who do not share their impairments, fostering greater understanding, empathy, and inclusivity within society. By using cutting-edge techniques, the proposed system aims to revolutionize communication dynamics and pave the way for a more inclusive and accessible future for all individuals, regardless of their hearing or speaking abilities.*3.1 Objectives*

a. To develop a communication system that enables effective two-way interactions between deaf and mute individuals and non-deaf and mute individuals.

b. To convert Speech/Audio to Sign language using Machine Learning and Animations to enhance the communication capabilities of people with hearing or Speaking disabilities.

c. To provide a user-friendly tool that minimizes the effort required for communication.

d. To address variations in sign language gestures among different DnM individuals and improve the systems adaptability and precision.

4. PROPOSED SYSTEM

To use our system for two-way communication, users must first create an account with a valid username and appropriate password on the website. Subsequently, they must log in. Upon successful login, they will be redirected to the home page automatically.

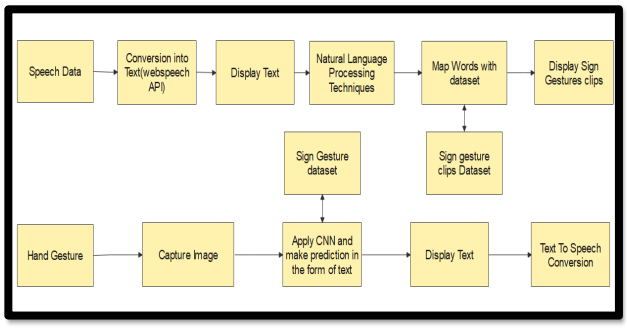

Fig.1: System Architecture

4.1 Convert speech to sign language

a. Users can input spoken words into a search bar, where the Web Speech Recognition API detects and transcribes the audio into text.
b. To process this text, we utilize the NLTK library, a valuable tool in natural language processing. NLTK performs word tokenization, which breaks the text into individual tokens.
c. These tokens then appear on the web page.
d. When users click the play button, the system displays a corresponding sign or action for each token in response.

4.2 Sign language to text

This system employs a vision-based approach where all signs are represented with bare hands, eliminating the need for any artificial devices for interaction.

a. Users perform sign language gestures in front of a camera.
b. Captured visual data undergoes pre-processing (resize, normalize, denoise).

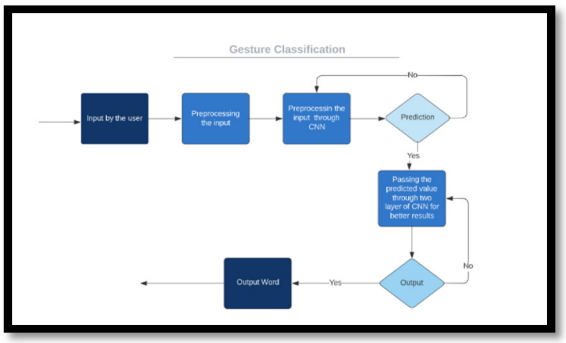

Fig 2: Sign language to text conversion

c. A trained CNN recognizes gestures from images, extracting essential features from gestures.
d. The CNN predicts gesture meaning in the text.
e. Conversion from image to text enhances accessibility for both deaf and hearing users.

4.3 Data Set Generation

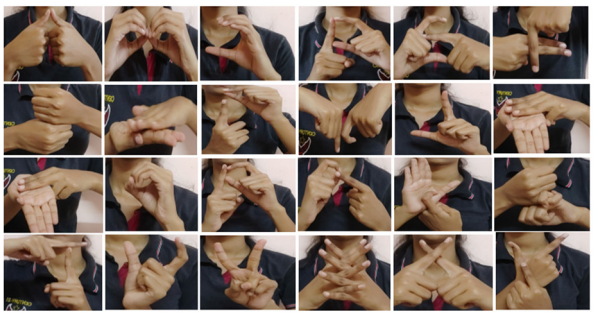

Fig 3: Dataset

4.3.1 We significantly enhanced the text-to-sign conversation module by creating a custom dataset. This dataset contains video clips demonstrating sign language gestures for letters, everyday words, and numerical figures. The inclusion of this diverse dataset has led to notable improvements in accuracy.

4.3.2 We attempted to find pre-existing datasets for the sign-to-text conversation module but couldn't locate raw image datasets that met our requirements. We encountered datasets in the form of RGB values only. Consequently, we opted to create our dataset for experimentation purpose.

Here are the steps we followed:

a. Utilizing OpenCV Library: We employed the OpenCV library to create our dataset.

b. Data Collection: Initially, we captured approximately images of each symbol in ISL (Indian Sign Language) for training purposes and around 30 images per symbol for testing purposes.

c. Frame Capture: We began by capturing each frame displayed by the webcam of our machine.

d. Region of Interest (ROI) Definition: In each frame, we defined a Region of Interest (ROI) represented by a blue bounded square. This ROI specifies the area where the sign language gesture is performed.

5. METHODOLOGY

5.1 Convert speech to sign language

Natural Language Processing (NLP) empowers computers to comprehend and produce human language. By leveraging computational algorithms and linguistic principles, NLP analyses text data for tasks like classification, sentiment analysis, and language translation. It revolutionizes industries by automating tasks, improving customer experiences, and facilitating efficient communication in areas like healthcare, finance, education, and entertainment.

5.1.1 Tokenization

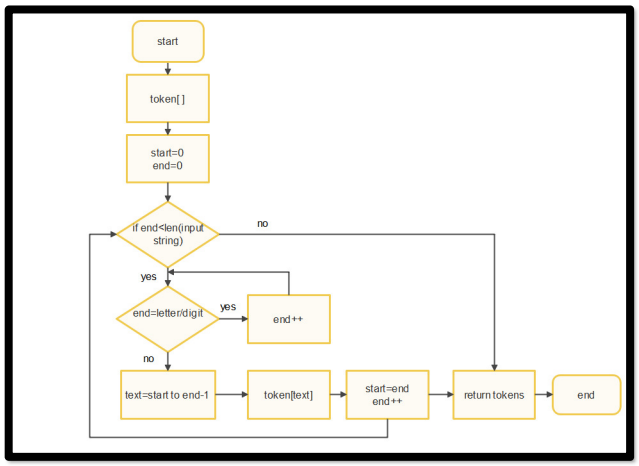

Fig 4: Tokenization

Algorithm:

Step 1: Initialize an empty list to store tokens.
Step 2: Initialize two pointers, `start` and `end`, both set to 0.
Step 3: Loop while `end` is less than the length of the input text:

a. If the character at `end` is a valid token character (e.g., a letter or digit), increment `end`.
b. If the character at `end` is not a valid token character:
   i. Extract the token from the text from `start` to `end - 1`.
   ii. Add the extracted token to the list of tokens.
   iii. Set `start` to `end`.
   iv. Increment `end`.

Step 4: After the loop, check if there is a token remaining from the last iteration (i.e., from `start` to the end of the text).
a. If there is, extract and add this token to the list of tokens.
Step 5: Return the list of tokens as the output.

5.1.2 Web Speech Kit API

The Web Speech API, also known as the Web Kit Speech Recognition API, allows web developers to integrate speech recognition capabilities into their applications. It provides a straightforward interface for accessing speech recognition services directly from the browser, enabling users to interact with websites through voice commands. The API processes spoken language input and converts it into text, which developers can then use to trigger actions or commands within their web applications. By leveraging this API, developers can create more accessible and interactive user experiences, ranging from voice-activated search engines to voice-controlled virtual assistants. The Web Speech API continues to evolve, offering improved accuracy and language support and paving the way for innovative voice-enabled web applications across various platforms and devices.

5.1.3 lemmatization

Lemmatization in natural language processing (NLP) is a technique used to reduce words to their base or canonical form, known as the lemma, which represents the dictionary form of a word. Unlike stemming, which simply chops off suffixes to find the root word, lemmatization considers the morphological analysis of words to produce the lemma. This process involves identifying the part of speech of a word and applying rules specific to each part of speech to obtain the lemma. Lemmatization is particularly useful for tasks like text normalization, where maintaining the semantic meaning of words is crucial. It helps improve the accuracy of downstream NLP tasks such as text classification, information retrieval, and machine translation. Additionally, lemmatization can aid in handling different word forms and variations, contributing to better linguistic analysis and understanding in NLP applications.

5.1.4 Stop word removal

Stop word removal is a crucial preprocessing step in natural language processing (NLP) that involves filtering out common words that carry little semantic meaning, such as "the," "is," and "and." By eliminating these stop words from text data, NLP algorithms can focus on the more significant words, improving the efficiency and accuracy of tasks like text classification, sentiment analysis, and information retrieval. This process helps streamline text processing pipelines and enhances the ability to extract meaningful insights from textual data.

5.2 Sign language to text

5.2.1 OPENCV

OpenCV is a versatile computer vision library that supports multiple programming languages and platforms, including Python, C++, Java, Windows, Linux, OS X, Android, and iOS. OpenCV-Python, specifically personalized for Python developers, seamlessly integrates the robustness of the OpenCV C++ API with Python's simplicity and readability. While Python may be slower than languages like C/C++, its extensibility allows for computationally intensive tasks to be implemented in C/C++ and seamlessly integrated with Python. Using Python-NumPy for optimized numerical operations, OpenCV-Python facilitates efficient array manipulation, enhancing interoperability with other Python libraries such as SciPy and Matplotlib.

5.2.2 CNN

A convolutional neural network (CNN) is a specialized deep learning architecture tailored for analyzing structured network data, notably images. It comprises various layers, namely convolutional layers, pooling layers, and fully connected layers. CNNs employ convolution operations to discern features within input data, followed by pooling operations to condense dimensionality and capture pivotal information. These networks are highly effective in tasks like image classification, object detection, and image recognition due to their ability to automatically learn hierarchical patterns and features directly from the data. Training a CNN involves optimizing the network's parameters (weights and biases) through a process called backpropagation. This process involves passing input data through the network, calculating the loss (or error), and adjusting the parameters to minimize this loss using optimization algorithms.

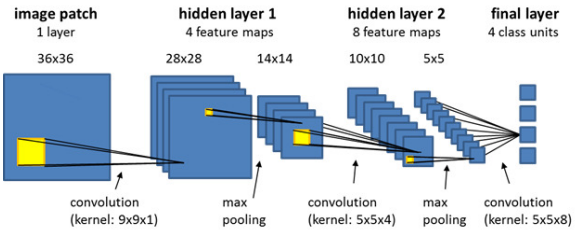

Fig 5: CNN (Convolutional Neural Network)

Input Data: The process begins with input data, which consists of images representing hand gestures in sign language. Each input image is typically represented as a matrix of pixel values.

Convolutional Layers: In C Layers, the input image traverses through one or more layers where each layer applies a series of filters. These filters facilitate the identification of various features, such as edges, textures, shapes, and other characteristics within the image. By leveraging these filters, meaningful patterns are extracted from the input images.
After each convolutional operation, an activation function such as ReLU is applied element-wise. This introduces non-linearity to the network, empowering it to discern complex relationships and patterns within the data.

Pooling Layers: These layers are utilized to decrease the size of the feature maps produced by the convolutional layers. This assists in reducing their spatial dimensions, thereby enhancing manageability. Typical pooling operations, such as max pooling or average pooling, are applied, preserving crucial information while simultaneously decreasing computational load.

Flattening: After passing through a sequence of convolutional and pooling layers, the resulting feature maps are flattened to create a one-dimensional vector. This transformation effectively converts spatial information into a format that can be seamlessly input into fully connected layers.

Connected Layers: The flattened feature vector is then transmitted through one or more fully connected (dense) layers. Each neuron within a dense layer establishes connections with every neuron from the preceding layer, facilitating the network's ability to comprehend intricate relationships among the features extracted from the input images.

Output Layer: The final, fully connected layer generates the network's predictions. The neuron count in this layer aligns with the total classes present in the sign language alphabet (e.g., if recognizing Indian Sign Language, there would typically be 26 output neurons representing each letter of the alphabet). The output layer employs an activation function suitable in the context of multi-class classification. SoftMax is employed to generate a probability distribution across the available classes.

Output: The output layer generates a probability distribution across the classes, illustrating the probability of the input image representing each sign language character. The predicted class corresponds to the one with the highest probability score.

6. RESULT

[1] The output of this system involves generating an equivalent sign language representation for a given English text. It produces a video clip featuring Indian Sign Language (ISL) signs. A predefined database contains video recordings for individual words, and the output video is a compilation of these signs seamlessly merged together. This system takes speech as input through a microphone using a Web Speech Recognition API, which converts speech into text format. The text is then pre-processed using natural language processing (NLP). Text pre-processing consists of many things: tokenization, stemming, lemmatization, and stop word removal. Example: The sentence is "Welcome to Indian Sign Language." It will be displayed on the screen as depicted in Fig. 7. Each word will be broken down and displayed individually, facilitating a clear understanding of the sentence structure. For every word, a corresponding video clip will automatically play. If we wish to pause a video, we can do so by clicking the play/pause button. In Fig. 7, a video clip is shown for the word "welcome," with the word also highlighted for emphasis.

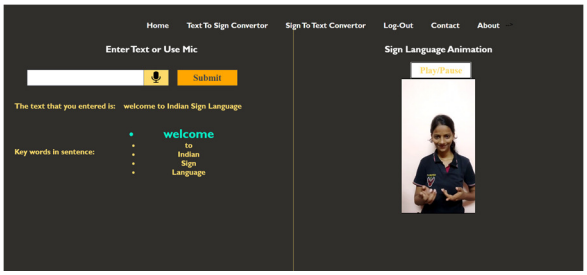

Fig.6: Pre-processing Text and displaying output

Similarly, output for other words is given below:For word “to” there is no corresponding video clip available, the text is further broken down into individual character,

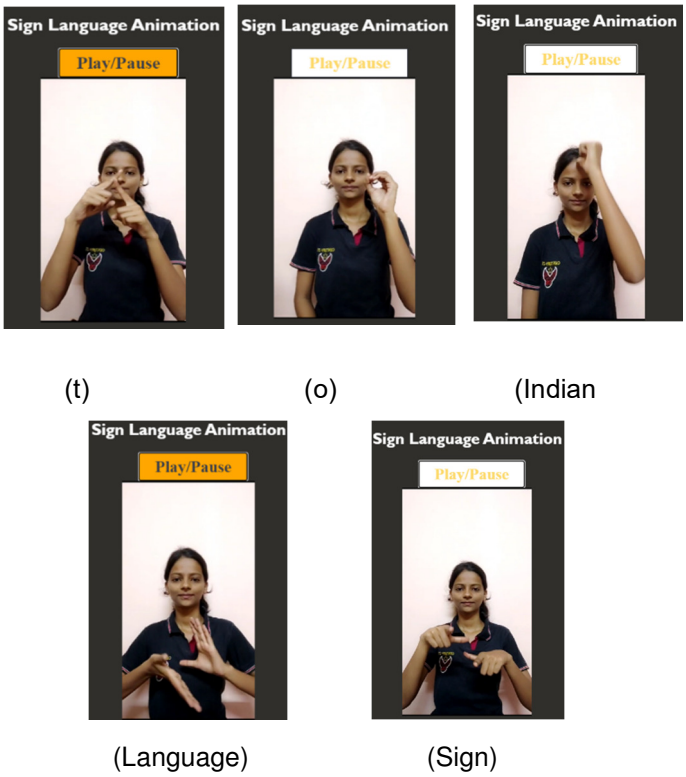

Fig.7: Sample Output

2] Gestures are captured via a camera and serve as input for a Convolutional Neural Network (CNN) algorithm. This algorithm extracts features from the gestures and matches them with corresponding gestures in a dataset. The identified gesture is then outputted by the system.

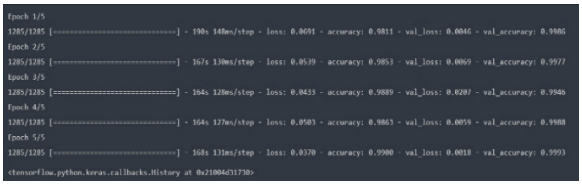

Fig.8: Model summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 128, 128, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 32768)             0
_________________________________________________________________
dense (Dense)                (None, 128)               4194432
_________________________________________________________________
dense_1 (Dense)              (None, 128)               16512
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 96)                12384
_________________________________________________________________
dropout_1 (Dropout)          (None, 96)                0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                6208
_________________________________________________________________
dense_4 (Dense)              (None, 27)                1755
=================================================================
Total params: 4,240,859
Trainable params: 4,240,859
Non-trainable params: 0
```

Fig.9: Training Evaluation

Let's illustrate the functionality of our system using the word "LOW" as an example. Initially, when the individual signs the letter "L," the system analyses the gesture and identifies it as the letter "L." Consequently, the system promptly displays the letter "L" on the interface as shown in fig. 10, providing real-time feedback to the user.

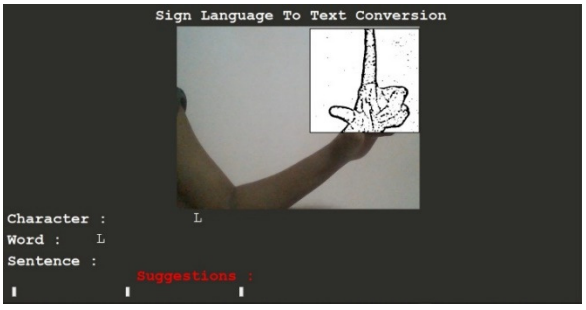

Fig. 10: Letter L

As the interaction progresses, the individual proceeds to sign the letter "O." as shown in fig. 11 In response, the system processes the new gesture, recognizing it as the letter "O" and seamlessly integrates it with the existing displayed content. Through this dynamic process, the system appends the letter "O" to the previously recognized letter "L," forming the sequence "LO" on the interface.

Fig. 11: Letter O

Continuing the sequence, in fig. 12 when the person signs the letter "W," the system engages in the same analysis and recognition process. Upon successfully identifying the gesture as the letter "W," the system intelligently adds it into the existing sequence. By appending the letter "W" to the previously recognized sequence "LO," the system effectively completes the word, the entire word "LOW" is then displayed.

Fig 12: Letter W

7. CONCLUSION

In summary, the proposed state-of-the-art bilateral communication system presents a promising resolution to the obstacles encountered by Deaf and Mute (DnM) individuals when engaging with the broader community. Utilizing state-of-the-art technologies such as machine learning and deep learning, this system effectively bridges the communication gap between DnM individuals and those without hearing or speaking impairments. Through speech-to-sign language and sign language-to-text transcription, facilitated by algorithms and custom datasets, the system enables seamless and inclusive communication.

8. REFERENCES

[1] Ankita Harkude#1, Sarika Namade#2, Shefali Patil#3, Anita Morey #4 1,2,3,4#Department of Information Technology, Usha Mittal Institute of Technology, SNDT Women's University, Juhu-Tara Road, Sir Vitthaldas Vidyavihar, Santacruz(W), Mumbai 400049

[2] Vaidya, O.; Gandhe, S.; Sharma, A.; Bhate, A.; Bhosale, V.; Mahale, R. Design and Development of Hand Gesture based Communication Device for Deaf and Mute People. In Proceedings of the IEEE Bombay Section Signature Conference (IBSSC), Mumbai, India, 4–6 December 2020; pp. 102–106. https://doi.org/10.1109/IBSSC51096.2020

[3] Deb, S.; Suraksha; Bhattacharya, P. Augmented Sign Language Modeling (ASLM) with interaction design on smartphone An assistive learning and communication tool for inclusive classroom. Procedia Comput. Sci. 2018, 125, 492–500. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2017.12.064

[4] M. V.D. Prasad, P. V. V. Kishore, E. Kiran Kumar, and D. Anil Kumar, "Indian sign language recognition system using new fusion based edge operator," J. Theor. Appl. Inf. Technol., vol. 88, no. 3, 2016.

[5] J. Serra, "Image analysis and mathematical morphology," by Academic Press, London, 1982, Cytometry, vol. 4, no. 2, 1983, doi: 10.1002/cyto.990040213.

[6] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction : a survey," Artificial intelligence review pp. 1–54, 2015, doi: 10.1007/s10462-012 9356-9.

[7] Neha Poddar, Shrushti Rao, Shruti Sawant, Vrushali Somavanshi, Prof.Sumita Chandak "Study of Sign Language Translation using Gesture Recognition" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2015.

[8] Natural Language Processing: State of The Art, Current Trends and Challenges Diksha Khurana1, Aditya Koli1, Kiran Khatter1,2 and Sukhdev Singh1,2 1Department of Computer Science and Engineering Manav Rachna International University, Faridabad-121004, India 2Accendere Knowledge Management Services Pvt. Ltd., India

[9] Deaf Mute Communication Interpreter Anbarasi Rajamohan, Hemavathy R., Dhanalakshmi M. (Department of B.M.E., Sri Sivasubramania Nadar College of Engineering, Chennai, Tamil Nadu, India.) International Journal of Scientific Engineering and Technology (ISSN: 2277-1581) Volume 2 Issue 5, pp: 336-341 1 May 2013.