IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Sign Language Recognition

Sign-Speak

Pawar Swaraj, Maheshwari Vardhan, Gosavi Onkar, Sonawane Chaitanya
Mr. P. S. Chavan (Guide)
Department of Computer Technology
K.K Wagh Polytechnic, Nashik, India 422009

Abstract: Sign-Speak is an innovative web application designed to address the communication needs of mute individuals by employing sign language detection technology. In this paper, we present the motivation behind Sign-Speak, outlining the pressing need for effective communication tools for the mute community. The project leverages advanced technologies, including the Django framework for Python-based web applications, and utilizes a model trained with Google Teachable Machine to detect and interpret sign language gestures. It also encompasses the entire development process, from model training with a carefully curated dataset to the implementation of real-time sign language detection in a web environment. Additionally, Sign-Speak incorporates speech synthesis to provide a comprehensive solution for interpreting and vocalizing detected signs. This paper discusses the technical aspects of the project, its implementation details, and the potential impact on enhancing communication accessibility **for mute individuals**

Index Terms **– Machine Learning, Django Framework, Model, Hand Detection, gtts, pyttsx4.**

I. INTRODUCTION

Sign language, as a fundamental means of communication for individuals with hearing and vocal disabilities, plays a crucial role in their daily lives. However, those facing such challenges encounter difficulties in effective communication, necessitating the development of innovative systems to enhance their quality of life. In response to this need, we present a comprehensive system designed to ease communication for individuals with speech and vocal disabilities.

Sign language encompasses a rich array of gestures, involving hand shapes, movements, and facial expressions. Recognizing the complexity of these gestures, our system aims to identify specific elements such as head and hand orientation, movements, facial cues, and body pose. Focusing on American Sign Language, widely utilized by the disabled community, our proposed system targets the recognition of static and dynamic gestures corresponding to the letters a-z and some extra words repeatedly use in day to day lives.

The system's adaptability and performance are achieved in dynamic and minimally cluttered backgrounds through large set of data with different backgrounds and skin tones used during training Incorporating speech recognition, our system utilizes the gtts-pyttsx3, to map spoken alphabets to text with remarkable precision. Subsequently, this text is further mapped to a corresponding image or video, depending on whether the recognized gesture is static or dynamic.

II. LITERATURE REVIEW

[1] The paper presents a novel system to aid in communicating with those having vocal and hearing disabilities. It discusses an improved method for sign language recognition and conversion of speech to signs. The algorithm devised is capable of extracting signs from video sequences under minimally cluttered and dynamic background using skin color segmentation. It distinguishes between static and dynamic gestures and extracts the appropriate feature vector. These are classified using Support Vector Machines. Speech recognition is built upon standard module - Sphinx. Experimental results show satisfactory segmentation of signs under diverse backgrounds and relatively high accuracy in gesture and speech recognition.

[2] The objective of this paper is to create one such system named, Sign Detect which is a Mobile Application which efficiently (1) recognizes the Sign Language; (2) quickly translates it to English and (3) then generates text and audio formats of the same. The system was tested for its accuracy with end users. Also, their User Experience (UX) was remarkable compared to the state-of-art systems used by them. As the system developed is a mobile application, it easily addressed the portability issue. And for wide acceptance, the application was developed in Flutter so that it can be easily installed on Android and iOS phones. Through this work, the authors hope to bridge the vast communication gap that is prevalent in today's world and take a step forward in this direction.

[3] Sign language is used by deaf and hard hearing people to exchange information between their own community and with other people. Computer recognition of sign language deals from sign gesture acquisition and continues till text/speech generation. Sign gestures can be classified as static and dynamic. However static gesture recognition is simpler than dynamic gesture recognition but both recognition systems are important to the human community. The sign language recognition steps are described in this survey. The data acquisition, data preprocessing and transformation, feature extraction, classification and results obtained are examined. Some future directions for research in this area also suggested.

[4] It focuses on a review of the literature on hand gesture techniques and introduces their merits and limitations under different circumstances. The theories of hand segmentation and the hand detection system, which employ the Haar cascade classifier, may be used to construct hand gesture recognition using Python and OpenCV. The use of hand gestures as a natural interface motivates research in gesture taxonomies, representations, and recognition algorithms, as well as software platforms and frameworks, all of which are briefly covered in this paper. We represent a comprehensive review of vision-based sign recognition algorithms published in the previous 16 years, emphasizing the importance of taking these things into consideration in addition to the algorithm's recognition accuracy when predicting its successful in real world applications.

III. SYSTEM ARCHITECTURE

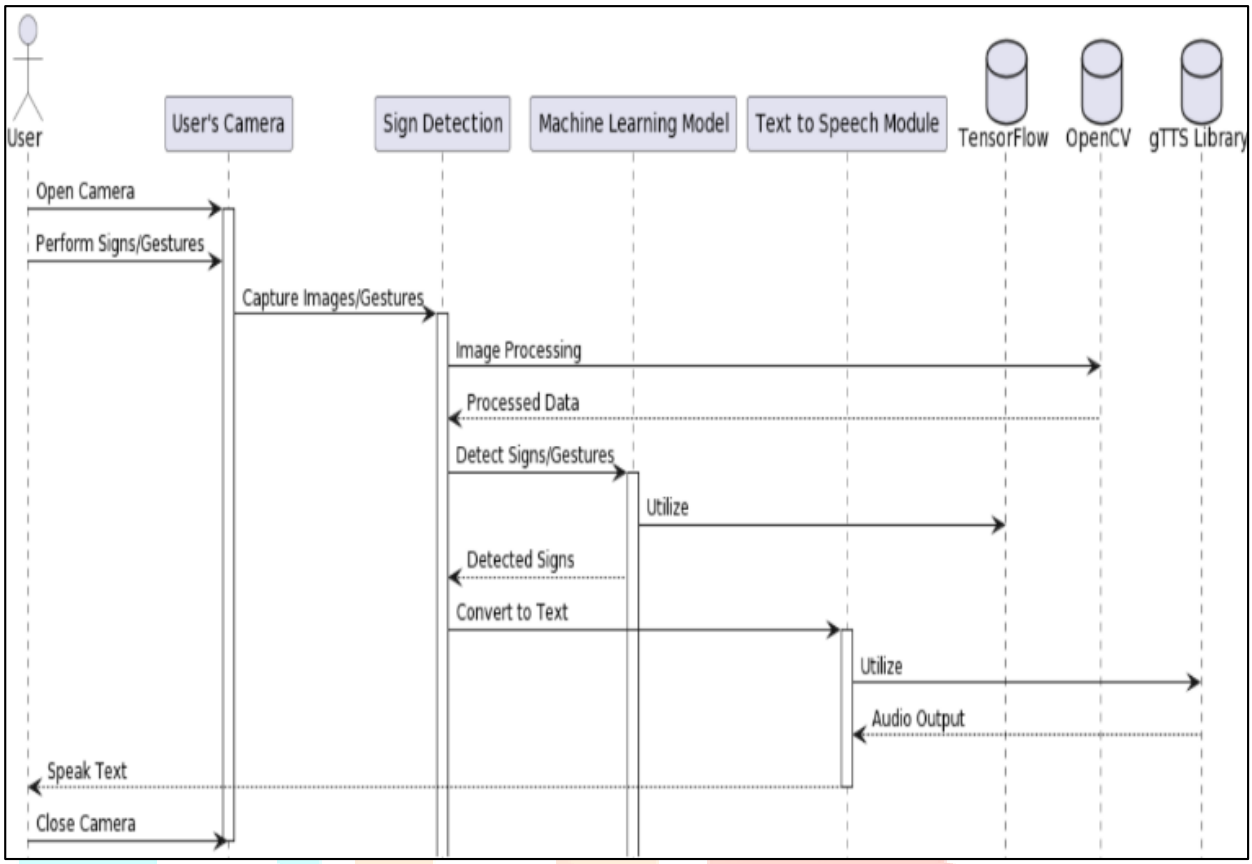

Fig 3.1. system architecture

Actors and Objects:

Actors: In the context of Sign-Speak, actors could be users who interact with the application, including sign language users and those unfamiliar with sign language.

Objects: Objects in the sequence diagram represent components or modules within the Sign-Speak application, such as the user interface, sign recognition module, customizable dictionary module, and feedback system.

Time Flow: The vertical axis represents time. Sign-Speak's sequence diagram captures interactions in real-time as users communicate through sign language and the application responds instantaneously.

Activation Bars: Activation bars represent periods of time when specific components are actively engaged. For example: The user's activation bar would be active when they use sign gestures or interact with the application. The activation bar for the sign recognition module would show when it's processing and recognizing sign gestures. The dictionary module's activation bar would be visible when the user customizes signs.

Messages: Messages in Sign-Speak's context indicate the flow of information between components. For instance. The user initiates a message by inputting sign gestures. The message passes to the sign recognition module, which interprets the signs. The translated message is sent as a response to the user's interface. If the user decides to customize signs, messages are exchanged between the user and the dictionary module.

Synchronous and Asynchronous Communication: Sign-Speak utilizes both synchronous and asynchronous interactions. For instance: Synchronous communication occurs when the user waits for real-time sign recognition results. Asynchronous interactions happen when users send feedback, which doesn't require an immediate response.

In Sign-Speak's system architecture, these elements would be visually depicted to showcase the interactions between users and the application's modules as they communicate through sign language, obtain real-time translations, customize dictionaries, and provide feedback for improvements.

IV.DEVLOPMENT

MODULE-1: GESTURE RECOGNITION

The Gesture Recognition module majorly works towards identifying the hand gestures done by human by image identification method. Here the images or frames in which a human hand is visible is processed and provided as input to the model to classify the label attached with the image and recognize the word. For the model the input image is processed and updated using various libraries and defined functions to make it as corresponding to the required input. At first, the frame with the hand is cropped to the size of human palm visible. This is done with the help of cv2's "HandTrackingModule".

Single Hand Detection (Sign Detection): The Gesture Recognition module focuses on identifying hand gestures through an image identification approach. When a single hand is detected in the frame, the "HandTrackingModule" from the OpenCV library is employed to isolate the human palm. The cropped image is then resized to fit within a 300x300 square while maintaining its aspect ratio. Subsequently, a blank white surface image is generated using the "NumPy" library, onto which the resized hand image is placed. This resulting image serves as the input for the machine learning model, facilitating accurate gesture recognition.

Multiple Hand Detection (Sign Detection with Two Hands): In scenarios where two hands are detected, each hand is processed individually. Utilizing the "HandTrackingModule," two separate images are cropped, each containing one hand. These cropped images are resized to 300x300 and then concatenated into a single image using the "NumPy" library. The concatenated image undergoes another resizing to 300x300, preparing it for further processing. Whether with a single or double hand, the final image is converted into black and white, enhancing visibility and contributing to increased model accuracy. This processed image is then fed into the machine learning model for accurate gesture classification.

The word to be guessed is determined using the "moving average" technique, where a separate prediction buffer accumulates predictions up to a certain limit. Calculations are performed on the buffer to identify the highest occurring or matching gesture, contributing to both smoothed predictions and nearly accurate guesses. This comprehensive process ensures effective gesture recognition, addressing scenarios with both single and multiple hands through appropriate mathematical calculations and the utilization of the "moving average" technique.

Overview of moving average:

As the system captures successive frames, the predictions from the gesture recognition model are accumulated in a buffer, known. The moving average is computed using a convolution operation. This involves sliding a window of a specified size (determined by **window size**) over the buffer of predictions. The convolution operation calculates the average within this window, providing a smoothed output. The length of the smoothed predictions is compared against a predefined buffer size (**buffer size**). To assess the stability of recent predictions, the variance of this subset of smoothed predictions is calculated. Variance measures the extent to which the predictions fluctuate. A threshold, in this case, is used to determine whether the predictions are stable. A lower variance indicates more consistent and less fluctuating predictions. If the variance falls below the threshold, it signifies stable predictions. In the event of stable predictions, the median value of the recent smoothed predictions is extracted. The median is chosen for stability, as it is less sensitive to extreme values compared to the mean. The extracted median value corresponds to the recognized gesture label.

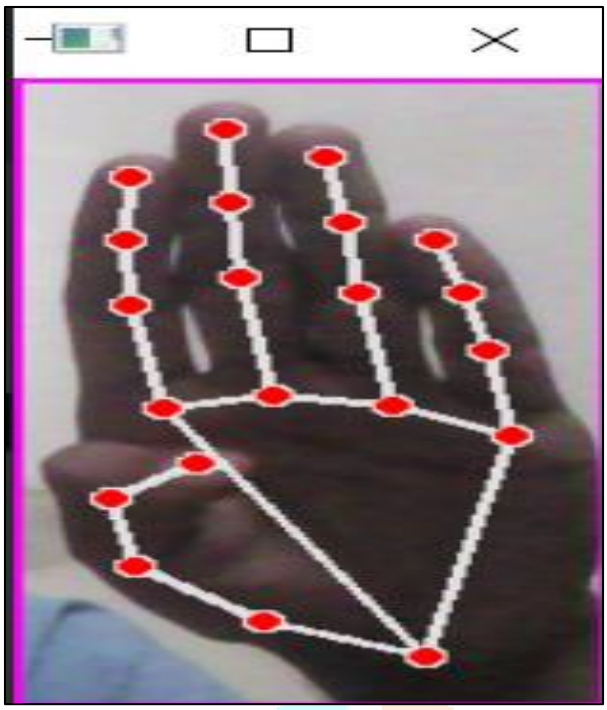

Fig 4.1. actual and image before processing

The above image shows the cropped image of detected hand with the hand points identified within it. The size of image may vary according to posture of hand in above image.

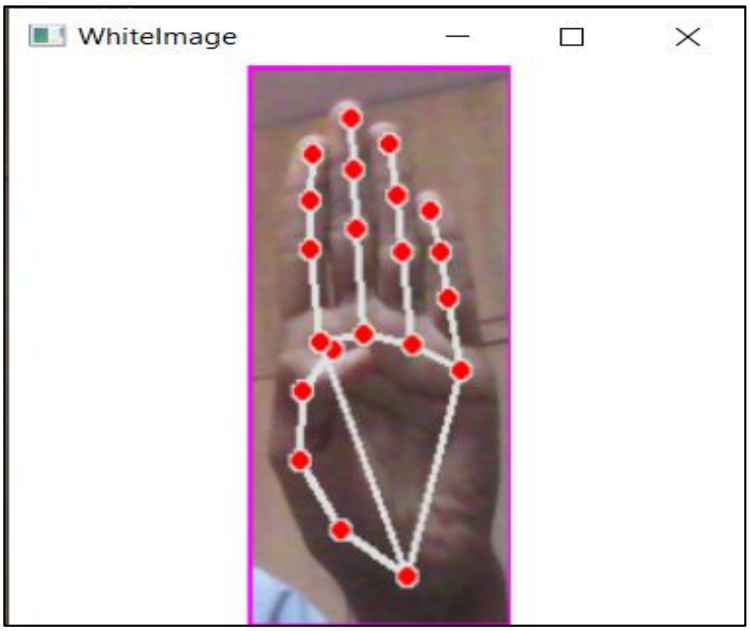

Fig 4.2. input image for ML model using NumPy

The above image shows the hand identified in Fig.2 been resized to the size of 300X300 image. This image is placed on the white surface to make it convenient to pass as an input to the model

Fig 4.3. actual output*:* video feed-single hand

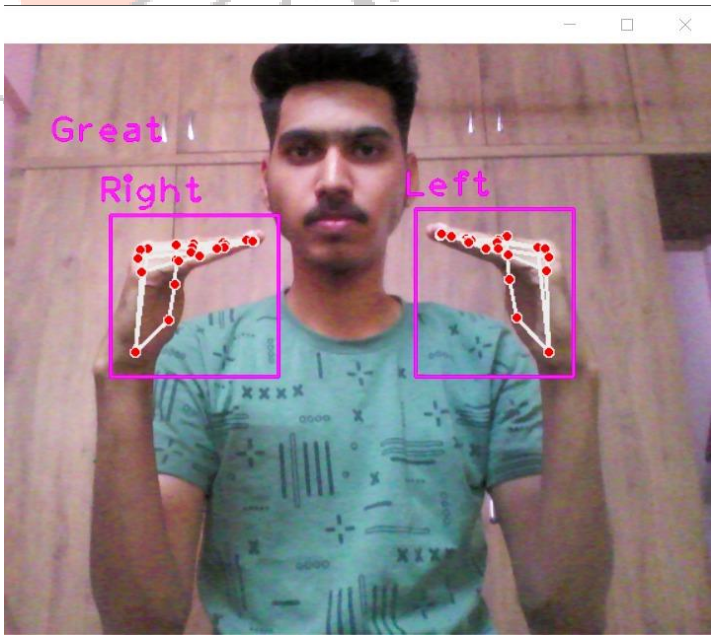

Fig 4.4. actual output*:* video feed-double hand

module-2: Text to speech conversion

The next module consists of generating a feasible output for humans and that is the voice generated output for the recognized hand gestures. This continues to be in loop until the hand gestures are being recognized and video feed is being processed. For the audio conversion we are using “pyttsx3” library of python. This library helps in converting text-to-speech. It is a text-to-speech conversion library in Python and it interfaces with various speech engines. “pyttsx3” uses different backends on different operating systems. On Windows, it uses the SAPI5 speech engine, and on other platforms, it uses espeak. It also allows us to modify some properties of speech output such as speed and voice. In our project we used it to take the string input from the ML Model which is the recognized gesture and convert it to speech output. For example, as shown in above figure that a gesture is recognized to English letter ‘B’ it speaks out the word or character using your machines speaker.

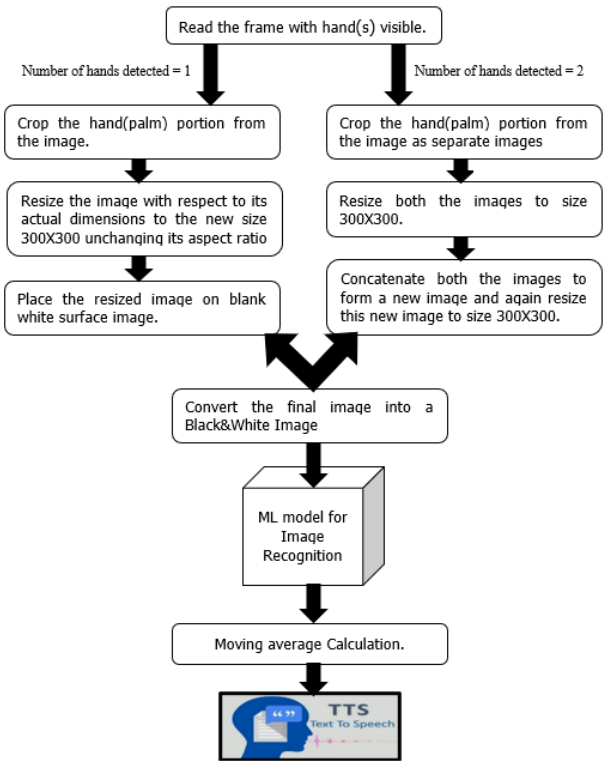

Fig 4.5. Flowchart-Control flow

module-3: Deployment

The deployment phase of our project involves utilizing the Django framework to seamlessly launch our sign detection program. When users initiate the sign detection process by clicking on the "Start Camera/Start Detection" button, the corresponding program runs in the background, initiating the camera feed to be prominently displayed on the user interface. This integration not only ensures a user-friendly experience but also leverages the power of Django to manage and serve our sign detection functionalities effortlessly. The Django framework facilitates the seamless interaction between the user interface and the underlying sign detection logic, making the deployment process efficient and accessible for users with hearing and vocal disabilities.

Additionally, the deployment module leverages Django's security features, ensuring a safe and protected environment for the execution of the sign detection algorithm. Users can confidently engage with the application, knowing that their data and interactions are handled with the utmost security and privacy. In

summary, the deployment module not only focuses on user accessibility but also harnesses the capabilities of the Django framework to create a reliable, secure, and efficient platform for sign language detection also it provides us with accessibility to add on other functionalities related to feedbacks, bot assistants, special learning environments for beginners, and many more to make use of project more intuitive and useful to users.

Fig 4.6. deployment of model on Django server

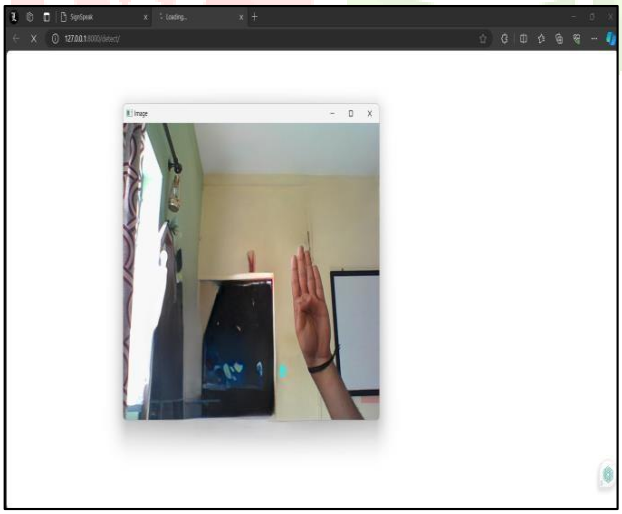

Fig 4.7-A. demonstration (sign B)

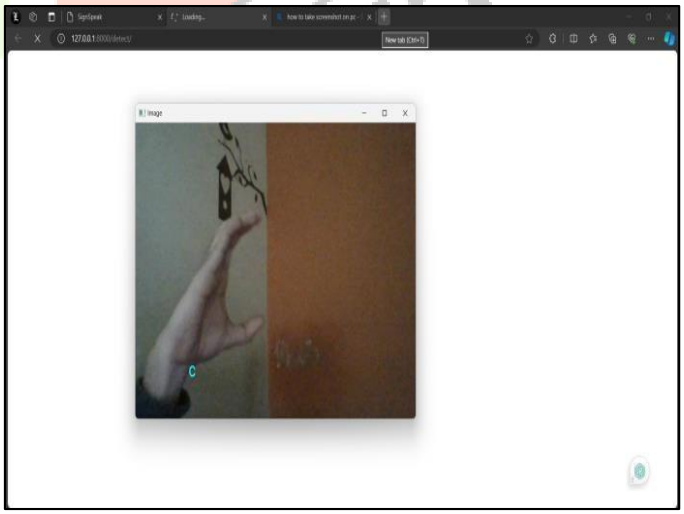

Fig 4.8-B.demonstartion(sign C)

V. RESULTS

Time Interval	Median Value	Accuracy
t1	25	~approx. 20%
t2	20	~approx. 50%
t3	10	~approx. 60%
t4	5	between 70-80%
t5	1	80% and more

Fig 5.1 Accuracy Graph

Table 5.1: Accuracy-Indicator

The accuracy graph of the Sign-Speak application provides a dynamic depiction of its performance over time. The Y-axis, representing the approximate accuracy percentage, unveils how effectively the application interprets sign language gestures. Concurrently, the X-axis, reflecting the median range, serves as a measure of stability, indicating how closely predictions align with each other. The graph unfolds over five distinct time periods (t1, t2, ...) revealing a compelling narrative. As time progresses, the accuracy of sign detection experiences a noticeable surge, coinciding with a reduction in the median range. This inverse correlation suggests that the application becomes increasingly adept at producing accurate predictions while concurrently achieving a heightened level of stability. A pivotal juncture in the graph emerges in the 0-1 range of the median, where the accuracy attains its zenith. This signifies the optimal performance state of the Sign-Speak application, marked by both precision in sign language interpretation and a remarkable level of consistency in predictions.

VI. CONCLUSION

In conclusion, Sign-Speak represents a significant leap forward in enhancing communication and inclusivity for individuals who rely on sign language. The system's innovative features, including real-time sign recognition, customizable sign language dictionaries, interactive learning, and text-to-sign translation, provide users with a powerful and flexible platform to express themselves effectively and naturally. The success of Sign-Speak lies in its ability to bridge the communication gap between sign language users and non-signers, thus breaking down language barriers and fostering inclusivity. By combining advanced computer vision and machine learning technologies with user-friendly interfaces, Sign-Speak offers a seamless and empowering communication experience.

While our sign detection system represents a significant advancement in aiding communication for individuals with hearing and vocal disabilities, there are areas that warrant further attention and improvement. One prominent challenge lies in achieving higher accuracy levels in gesture recognition. Despite the successful implementation of static and dynamic gesture recognition, refining the algorithms to enhance precision and reduce false positives is an ongoing objective also availability of light and camera quality play's important role for accurate sign detection, this dependency of model on external factors needs to be limited as far as possible.

Another area for consideration is the speech generation component. Although our system effectively converts recognized gestures into corresponding alphabets and words, further research and development can be directed towards refining the text-to-speech conversion process.it is observe that conversion of certain words is still unclear and sometimes even mis-pronunciation take place due to this anomaly. Ensuring natural and expressive speech output can significantly enhance the overall user experience and facilitate more effective communication.

Furthermore, the integration of the project with the Django framework, while successful in providing a scalable and secure platform, may benefit from additional optimizations. Future efforts could focus on streamlining the deployment process, enhancing the user interface, and exploring advanced features offered by Django to create a more seamless and user-friendly interaction. In addressing these challenges, our ongoing commitment is to continually refine and optimize our system. By addressing areas of lower accuracy, improving speech generation, and optimizing the integration with the Django framework, we aim to provide an increasingly effective and accessible solution for individuals with speech and hearing disabilities.in further research we are also tried to indulge user-specific vocabulary for each sign's defined by users through their own access to the applicatio

REFERENCES

❖ Literature Review:

[1] Anup Kumar, Karun Thankachan and Mevin M. Dominic, "Sign Language Recognition", Department of Computer Science
Engineering National Institute of Technology, Calicut, India 673601,2016
[2] Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to
Build Intelligent Systems'', Edition-2nd, O'Reilly Media,2019.
[3] 'MediapipeRepositories', https://github.com/google/mediapipe
[4] 'Hand Recognition with OpenCV 'https://github.com/simenandresen/handDetectionCV

❖ Book References:

1.Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent
Systems (2nd Edition).
Author: Aurélien Géron, Publication: O'Reilly Media.
2.Machine Learning in Python: Essential Techniques for Predictive Analysis
Author: Micheal Bowles.

❖ Web Reference:

• https://youtu.be/G4UfUY2NNSQ?si=_PiOQfCvpwujAeSQ
• https://youtu.be/wa2ARoUUdU8
• https://youtu.be/523smwZrlc4?si=Z03duUyU4BycpOzm
• https://ieeexplore.ieee.org/document/8741512
• https://adeshpande3.github.io/