Dr. Babasaheb Ambedkar Technological University
Lonere, Dist. Raigad, Pin 402103, Maharashtra

A
PROJECT REPORT ON

"Indian Sign Language to Text/Speech Translation"

Under the Guidance of

Dr. S. R. Zanwar

Submitted by

Swaraj Solanke (AI4072)
Parth Dhone (AI4073)
Pradyumna Digraskar (AI4075)

For the partial fulfillment for the award of

BACHELOR OF TECHNOLOGY
IN
Artificial Intelligence and Data Science

CSMSS
CHH. SHAHU COLLEGE OF ENGINEERING
Chhatrapati Sambhajinagar – 431011
(2024-25)

CSMSS

Chh. Shahu College of Engineering,

Chhatrapati Sambhajinagar, Maharashtra - 431011

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CERTIFICATE

This is to certify that the Project report entitled

“Indian Sign Language to Text/Speech Translation”

Submitted by
Swaraj Solanke (AI4072)
Parth Dhone (AI4073)
Pradyumna Digraskar (AI4075)

in partial fulfillment for award of the Degree **Bachelor of Technology** in **Artificial Intelligence and Data Science** of **Dr. Babasaheb Ambedkar Technological University**, Lonere, Raigad, during academic year 2024-25 .

Dr. S. R. Zanwar	**Dr. S. R. Zanwar**	**Dr.G.B Dongre**
Guide	**Head of the Department**	**Principal**

CSMSS

Chh. Shahu College of Engineering,

Chhatrapati Sambhajinagar, Maharashtra - 431011

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PROJECT APPROVAL SHEET

Project entitled “**Indian Sign Language to Text/Speech Translation**” submitted by **Swaraj Solanke (AI4072), Parth Dhone(AI4073), Pradyumna Digraskar (AI4075)** is approved for partial fulfillment for the award of **Bachelor of Technology in Artificial Intelligence and Data Science** of **Dr. Babasaheb Ambedkar Technological University**, Lonere, Raigad (M.S.) during academic year 2024-25.

Name and Sign	**Name and Sign**
Internal Examiners	**External Examiners**

Place: Chhatrapati Sambhajinagar

Date:

DECLARATION

We, the students enrolled in the Seventh semester of the B.Tech program in **Artificial Intelligence and Data Science** at CSMSS Chh. Shahu College of Engineering, Chhatrapati Sambhajinagar, hereby assert that our project work titled "**Indian Sign Language to Text/Speech translation,**" submitted to Dr. Babasaheb Ambedkar Technological University, Lonere, Raigad, during the academic year 2024-25, represents original research conducted by us.

This project work is presented as a partial fulfillment of the requirements for the Bachelor of Technology degree in **Artificial Intelligence and Data Science**. The findings presented in this report have not been previously submitted to any other University or Institute for the purpose of obtaining any degree.

Roll No.	**Name of the student**	**PRN No.**	**Signature**
AI4072	Swaraj Solanke	2225331995504	
AI4073	Parth Dhone	2225331995505	
AI4075	Pradyumna Digraskar	2225331995507	

Place: Chhatrapati Sambhajinagar
Date:

ACKNOWLEDGEMENT

We take immense pleasure in presenting this project report, as this page provides us with the opportunity to convey our heartfelt emotions and gratitude.

We extend our sincere thanks to our guide, **Dr. S. R. Zanwar**, whose guidance was invaluable at every step of this project. His motivation and confidence-boosting efforts were instrumental, and we acknowledge that this work would not have been possible without his support and encouragement.

Special appreciation is also extended to the project coordinator, **Dr. S. R. Zanwar** for guiding and motivating us throughout the project. We would like to express our gratitude to the Head of the Department, **Dr. S. R. Zanwar** , and the respected Principal, **Dr.G.B.Dongre** , for providing valuable resources and dedicating their valuable time to review our report.

Finally, we express our thanks to all the staff members of the **Artificial Intelligence and Data Science** and our friends, without whom the completion of this project would not have been possible.

Swaraj Solanke (AI4072)
Parth Dhone (AI4073)
Pradyumna Digraskar(AI4075)
B. Tech. (AI and DS)

CONTENTS

List of Figures

List of Graphs

List of Tables

List of Abbreviations

SN	Symbol	Illustrations
1	ISL	Indian Sign Language.
2	AUC	Area under Curve
3	ROC	Receiver operating Characteristic Curve

ABSTRACT

In the digital era, effective communication between hearing individuals and those with hearing impairments remains a significant challenge. Deaf and mute individuals often experience anxiety when interacting with the general public, leading to social isolation. To mitigate these issues, a mobile application has been developed that recognizes Indian Sign Language (ISL) in real-time and translates it into text and speech. This application employs advanced machine learning architectures.

The application captures sign language gestures using MediaPipe, OpenPose, and OpenCV libraries, which facilitate the accurate detection of sign language movements. Once the signs are recognized, they are translated into various regional languages spoken across India, utilizing models like Indic_model and Transformer for text conversion. This functionality allows users to receive output in their preferred format either as text or speech.

The innovative features of this application aim to bridge the communication gap between hearing individuals and those with hearing impairments. By promoting social integration and reducing feelings of isolation among deaf individuals, it empowers them to engage in educational settings alongside their hearing peers and collaborate effectively in professional environments.

Moreover, this application not only serves as a practical tool for everyday interactions but also has the potential to be expanded into a comprehensive educational platform that can accommodate a broader vocabulary of ISL through both manual and non-manual signs. The integration of computer vision techniques and natural language processing (NLP) further enriches its functionalities, paving the way for more inclusive communication solutions.

Keywords: Indian Sign Language, LSTM, GRU, MediaPipe , OpenCV, Indic_model , Transformer, NLP, computer vision.

Chapter I
INTRODUCTION

1.1 Introduction:

Indian Sign Language is a predominant sign language since the only disability D&M people have been communication related and they cannot use spoken languages hence theonly way for them to communicate is through sign language. Communication is a vital aspect of human interaction, enabling individuals to express their thoughts, emotions, andneeds. However, for individuals with speech or hearing impairments, traditional communication methods can be challenging. Indian Sign Language (ISL) is widely used by the deaf and hard-of-hearing communities in India as a means of communication. Despite its importance, a significant portion of the general population remains unfamiliar with ISL, creating a communication gap between individuals with hearing disabilities andothers. This nonverbal communication of deaf and dumb people called sign language. To bridge this gap, our project focuses on developing a real-time Indian Hand Sign Detectionsystem that converts ISL gestures into text and speech. By leveraging advancements in deep learning and computer vision, this system aims to enhance inclusivity, enabling seamless communication and fostering social integration forindividuals with speech or hearing impairments. The gestures we aim to train are as givenin the image below.

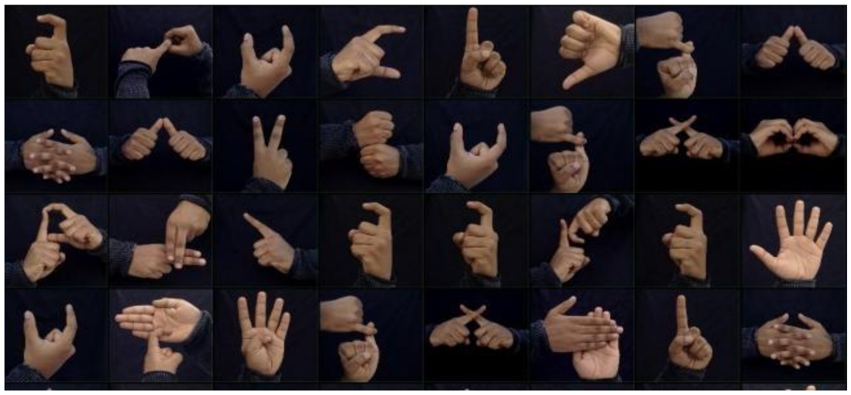


Figure 1.1.1: Sign Gestures Images

1.2 Necessity:

i. Bridging the Communication Gap:

ISL users face significant challenges in communicating with people who do not understand sign language. This project aims to bridge this gap by providing a tool that translates ISL gestures into text and speech, making communication accessibleand inclusive.

ii. Promoting Inclusivity:

An effective hand sign detection system empowers individuals with hearing disabilities to participate more fully in education, employment, and social interactions without the constant need for interpreters.

iii. Enhancing Accessibility:

Public services such as healthcare, banking, and government facilities can become more accessible when ISL-based communication tools are available, ensuring equalopportunities for all citizens.

iv. Leveraging Technological Advancements:

With the growing accessibility of AI and machine learning, it is now feasible to create robust systems that process and interpret ISL gestures in real-time with highaccuracy.

v. Customizing for Indian Context:

Most existing hand sign detection systems cater to American Sign Language (ASL),which differs significantly from ISL. A system tailored to Indian cultural and linguistic nuances is essential for effectively addressing the needs of ISL users.

vi. Encouraging Learning and Adoption:

This project can also serve as an educational tool for individuals interested inlearning ISL, increasing awareness and adoption of sign language in society

.

1.3 Challenges:

i. **Continuous Detection Challenges**:
Preventing the model from repeatedly detecting the same gesture or recognizingunintended gestures when no hand signs are present is a key challenge for real- time applications.

ii. **Speech Output Quality:**
Generating natural and contextually appropriate speech output requires integratingtext-to-speech systems that support multiple languages, accents, and tones to match the user's preferences and needs.

iii. **Integration with User-Friendly GUI:**
Designing an intuitive and responsive graphical user interface that can display thecamera feed, detected text, speech output, and word suggestions effectively is critical to ensure ease of use for all users.

iv. **Hardware and Cost Constraints:**
Ensuring the system runs efficiently on affordable and widely available hardware,such as smartphones or low-cost computers, is necessary for accessibility but presents optimization challenges.

v. **Sign Ambiguity and Context Awareness:**
Certain ISL signs can have multiple meanings depending on the context. Incorporating context-awareness into the system to interpret signs accurately ischallenging.

1.4 Applications:

i. **Assistive Communication for Individuals with Hearing Impairments:**
Provides a reliable tool for individuals with hearing or speech impairments to communicate seamlessly with non-signing individuals by translating Indian SignLanguage (ISL) into text and speech.

ii. **Educational Tools:**
Acts as an aid for teaching ISL to students, educators, and the general population,fostering greater awareness and adoption of sign language.

iii. **Customer Service and Public Assistance:**
Improves accessibility in customer-facing services such as banks, hospitals, retailoutlets, and government offices, enabling ISL users to communicate their needs effectively.

iv. **Healthcare Accessibility:**
Facilitates communication between healthcare providers and patients with hearing impairments, ensuring accurate diagnosis and treatment.

v. **Employment and Workplace Integration:**
Assists in creating inclusive workplaces by enabling communication betweenemployees with hearing impairments and their colleagues or clients.

vi. **Educational Institutions:**
Assists teachers and students in inclusive classrooms, allowing better interactionbetween hearing-impaired students and their peers or instructors.

vii. **Public Announcements and Accessibility:**
Can be integrated into public address systems to translate spoken announcementsinto sign language for individuals with hearing impairments, ensuring inclusivityin events, airports, or railway stations.

Chapter II

LITERATURE SURVEY

2.1 Introduction

Indian Sign Language is a predominant sign language since the only disability D&M people have been communication related and they cannot use spoken languages hence theonly way for them to communicate is through sign language.

Indian Sign Language is a predominant sign language since the only disability D&M people have been communication related and they cannot use spoken languages hence theonly way for them to communicate is through sign language.

Communication is the process of exchange of thoughts and messages in various ways suchas speech, signals, behavior and visuals. Deaf and dumb (D&M) people make use of theirhands to express different gestures to express their ideas with other people.

Gestures are the nonverbally exchanged messages and these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language

Table 2.2: Literature Survey Table

Title of paper	Date of publication	Author	Summary	Reference
Real-Time Sign Language Detection Using TensorFlow, OpenCV, and Python	2024	Prashant Verma	Discusses a system for real-time detection of sign language using TensorFlow and OpenCV. Focuses on Indian Sign Language (ISL) and leverages machine learning for gesture-to-text conversion. Uses computer vision techniques to avoid wearables and emphasizes GPU-accelerated training for efficient detection.	https://doi.org/10.22214/ijraset.2022.43439
Sign Language Recognition Using Deep Learning	2023	Harini Priya	Explores using deep learning models like CNNs and LSTMs for ISL recognition. Focuses on real-time applications and highlights the challenges in interpreting sequential gestures. The model integrates temporal features for better gesture-to-text translation accuracy.	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10533962&isnumber=10533882
Conversion of Sign Language Into Text	2023	Shalini Gupta, Mehul Mehta	Proposes a vision-based sign language recognition system using gesture recognition. The paper emphasizes low-cost solutions for ISL translation into text and explores gesture classification using machine learning models and image processing techniques	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10452617&isnumber=10452427
Sign Language Recognition Based on Computer Vision	2023	Divya Sharma, Arjun Mehta	Investigates the use of computer vision and artificial neural networks for ISL recognition. Incorporates preprocessing techniques to enhance gesture detection and highlights the use of non-manual features like facial expressions in recognition tasks.	https://doi.org/10.1007/s11042-023-15583-8

Sign Language Conversion to Text and Speech	2024	Sneha Rao, Manish Tiwari	Develops a bilingual system for ISL, converting gestures to text and speech in regional languages. Uses sequence-to-sequence deep learning models with attention mechanisms, showcasing high accuracy in aligning gestures with text and speech outputs.	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10674922&isnumber=10674743
A Novel Approach for Detecting Real-Time Indian Sign Language Using Deep Learning	March 2024	Sarvesh Moraskar, Anindita A. Khade	This study presents a real-time ISL recognition system using YOLOv8 for object detection, processing entire images in one pass to predict bounding boxes and class probabilities for gestures. It emphasizes the system's potential for applications in gesture-controlled robotics and human-computer interaction.	https://doi.org/10.1063/5.0200585
Real-time Sign Language Translation Using Computer Vision and Machine Learning	April 2024	Anil Gupta, Meera Joshi	The paper discusses a system leveraging CNNs and MediaPipe for real-time gesture detection, transforming recognized gestures into text and speech using natural language processing and text-to-speech APIs. It incorporates TensorFlow for training models and OpenCV for real-time image processing.	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10533962&isnumber=10533882
Development of ISL Recognition System Using Artificial Neural Networks	December 2023	Divya Sharma, Arjun Mehta	This research focuses on using a neural networks (ANNs) for ISL recognition, highlighting improv in detecting gestures with non-m features such as facial expressio system integrates preprocessing techniques and achieves signific accuracy improvements for mul gesture detection.	https://doi.org/10.1007/s10209-024-01162-7
Indian Sign Language Translation System for	October 2023	Sneha Rao, Manish Tiwari	The paper introduces a bilingual ISL system converting gestures to text and speech in regional languages. It	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber

Regional Languages Using Deep Learning			employs sequence-to-sequence models with attention mechanisms to align sign gestures with contextual text translations, demonstrating effective handling of complex regional vocabulary.	=10481626&isnumber=10481530
Real-Time Sign Language Translator for Deaf and Mute	2024	Kevin Thomas, Xin Sui	Describes advancements in real-time machine translation for deaf and mute users. The study integrates wearable-free computer vision techniques to enhance accessibility through ISL translation into text and speech in regional languages..	https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10136074&isnumber=10136027

2.3 Objective

i. **Accurately Detects ISL Gestures:**

Create a robust machine-learning model capable of recognizing and interpretingIndian Sign Language (ISL) hand gestures, including both static and dynamic signs, with high accuracy and efficiency.

ii. **Converts Gestures to Text:**

Translate the recognized ISL gestures into meaningful and grammatically correcttext, making the system accessible to individuals unfamiliar with sign language.

iii. **Generates Speech Output:**

Enable text-to-speech functionality to convert the detected ISL gestures into clearand natural speech in multiple languages, ensuring inclusivity and ease of communication.

iv. **Supports Two-Handed Signs:**

Extend detection capabilities to include two-handed gestures and ensure seamlessrecognition of complex signs.

v. **Provides Contextual Suggestions:**

Implement intelligent word suggestions and Autocompletion to assist users in forming complete sentences efficiently, reducing repetitive input and improvingcommunication speed.

vi. **Operates in Real-Time:**

Design the system to process gestures and generate outputs in real-time withminimal latency, ensuring smooth and uninterrupted interaction.

Chapter III

SYSTEM MODELLING

3.1 Introduction

System modelling is a critical phase in the development of our Indian Hand Sign to Text and Speech system. It provides a structured framework to conceptualize, design, and implement the system effectively. By breaking down the complex processes involved in recognizing Indian Sign Language (ISL) gestures and converting them into text and speech, system modelling helps ensure clarity, scalability, and accuracy.

For this project, system modelling involves defining the key components, interactions, and workflows required for real-time hand gesture recognition, text conversion, and speech generation. It establishes the technical roadmap for integrating computer vision, machine learning, natural language processing, and text-to-speech technologies into a cohesive system.

By developing a detailed system model, the project can move forward with a clear understanding of the interactions and dependencies between various components, paving the way for a robust, reliable, and efficient solution.

3.2 Model Development:

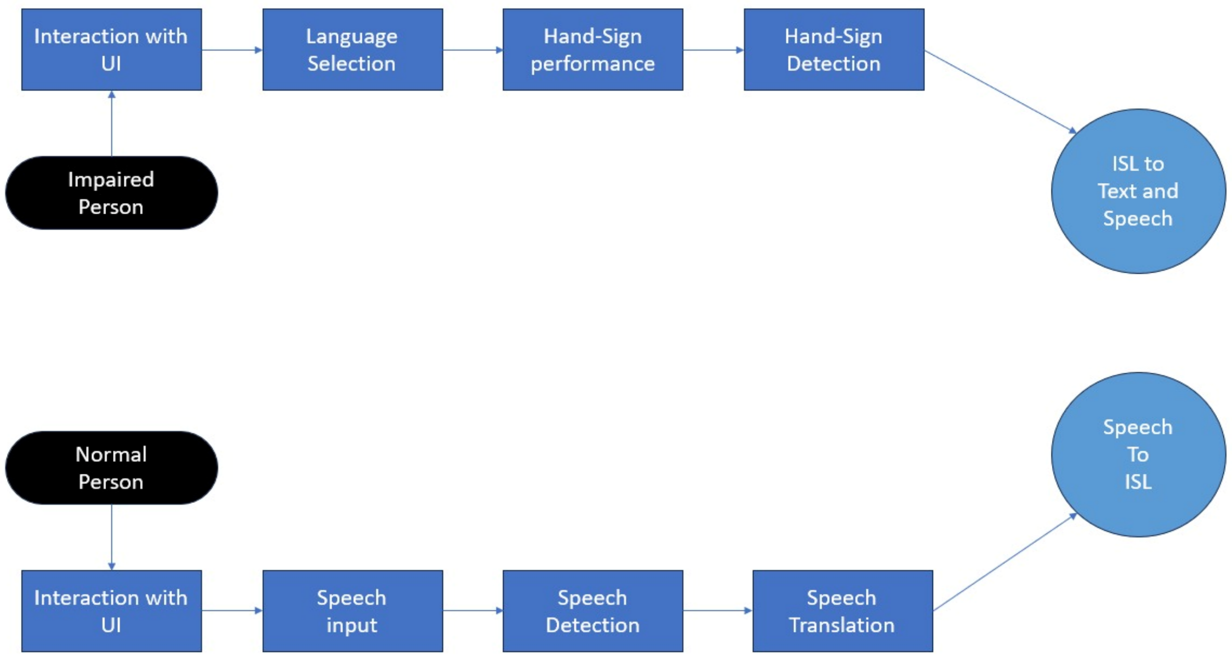


Fig 3.2.1 Basic Workflow

This flow will indicate the basic flow and interaction of our model where we are targeting two type of users one will be the mute or deaf person who knows what sign languages are and other one is for the normal person will communicate with the deaf or mute person.

Initially if the impaired person will interact with the UI, he will select the native language of the region and then person will start signing ISL and the first model will detect the signs In addition, it will convert it to text as well as speech into the users selected language in Real-Time.

In addition, if the normal person interact with the website then the speech taken from that person will automatically detect the user's native language and will be converting it to the Indian Hand Signs for the impaired person.

1. **Indian Sign Language to Text /Speech:**

   When user click on this module, he will direct to new window where he will show the sign image in real time that sign image is detected in real time and converted into textand speech.

2. **Speech/Text to Indian sign language:**

   When user click on this module he will direct to next page where he will get the mic option when user click on the mic he will speak in real time the model preprocessed itand provide the output of sign image output window.

3. **Indian Sign To Text :**

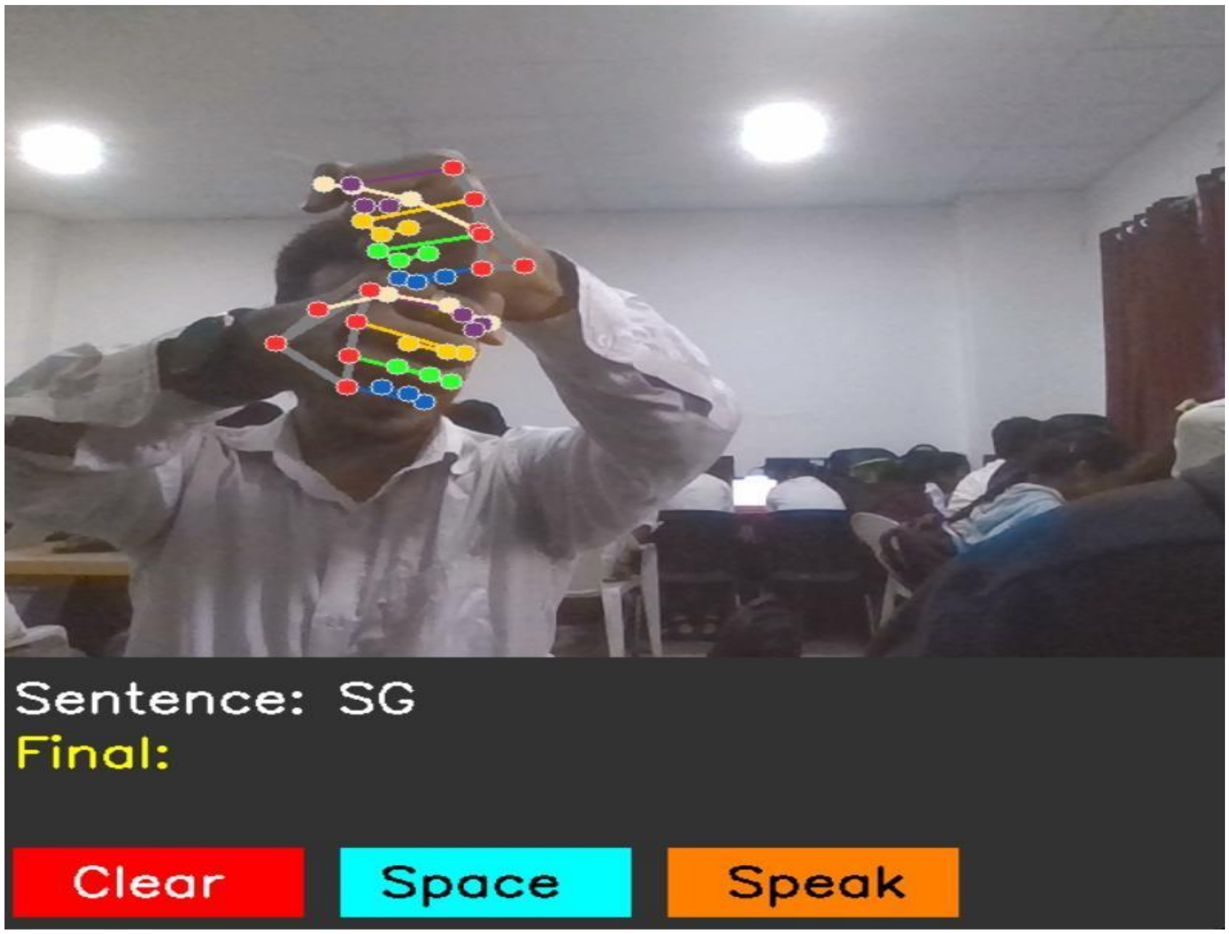


Figure 3.2.2: Indian Sign To Text Page

This is window of sign to text module where user is showing the sign image in real time that images is detected in real time in window .he will get the suggestion of word , providethree buttons for clear the text and providing the space in between the words . Once the sentence is created that can be hear by normal people.

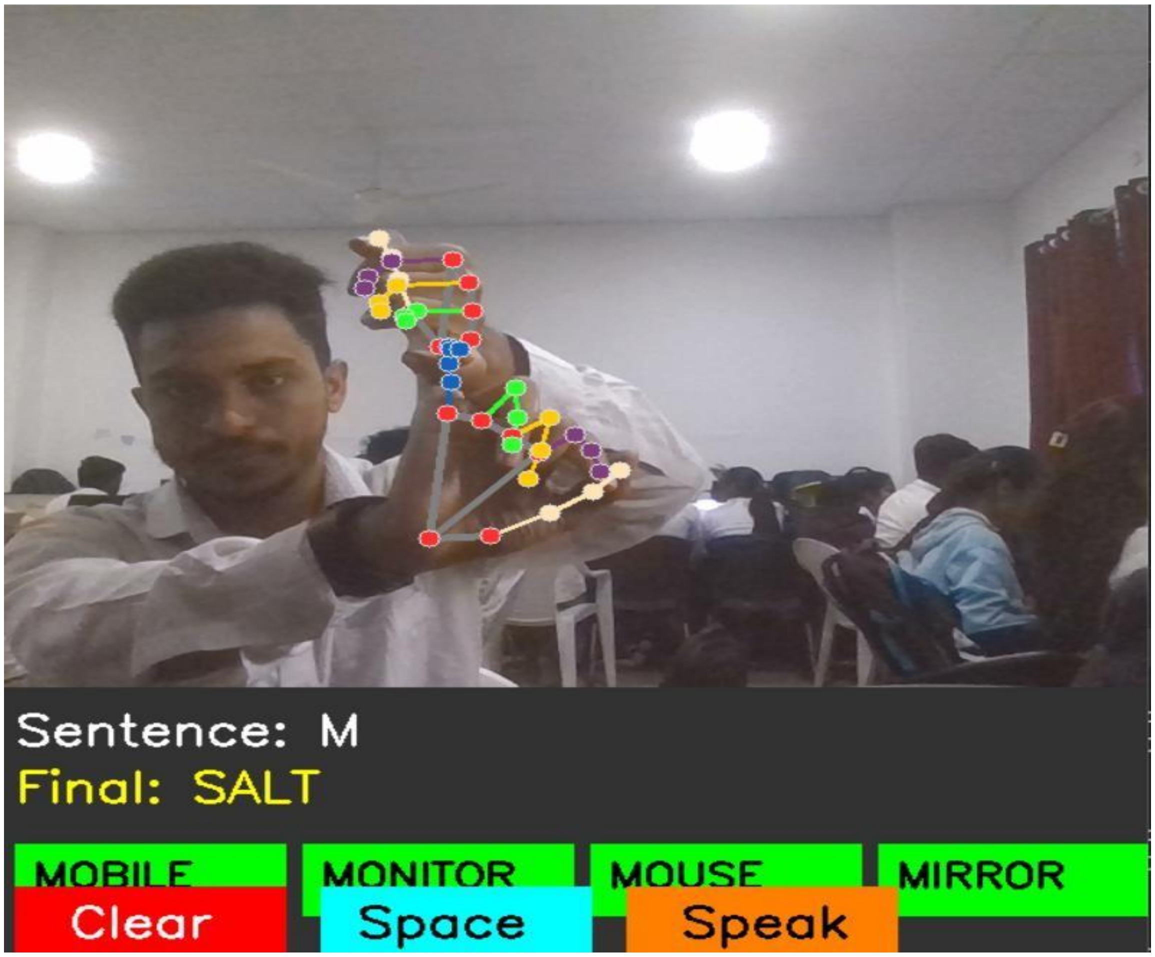


Figure 3.2.3: Module Indian Sign To Text Page

Here we can see the suggestion provide by model to deaf person so the deaf person canwe communication more quickly.

3.3 Architecture Diagram

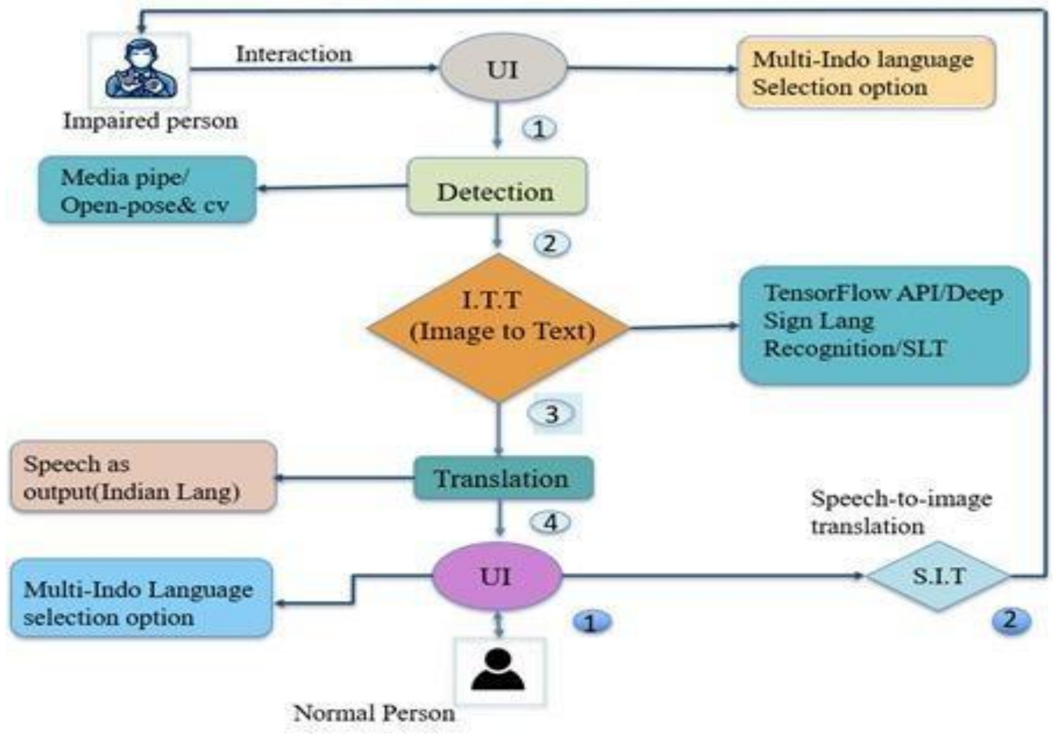

Interaction
UI
Multi-Indo language Selection option
Impaired person
1
Media pipe/ Open-pose& cv
Detection
2
I.T.T (Image to Text)
TensorFlow API/Deep Sign Lang Recognition/SLT
3
Speech as output(Indian Lang)
Translation
4
Speech-to-image translation
Multi-Indo Language selection option
UI
S.I.T
1
2
Normal Person

Figure 3.3.1: Architecture Diagram

Initially when Deaf and Mute user visits our application, he will get an option for selecting multiple Indian regional-native languages. Once he clicks one of the options then the camera of that particular device starts and it will perform detection of the Indian Hand Gestures. The detected gestures will be converted to Letters using the Media-pipes Hand Connection System .Now this detected letters will form a particular sentences this sentences will be now converted to the user’s native language.

Now the Normal user will be selecting the second option (Speech to Image) and he will give the speech as input in the native language. After this the detected text will be convertedto image which are already presented in dataset.

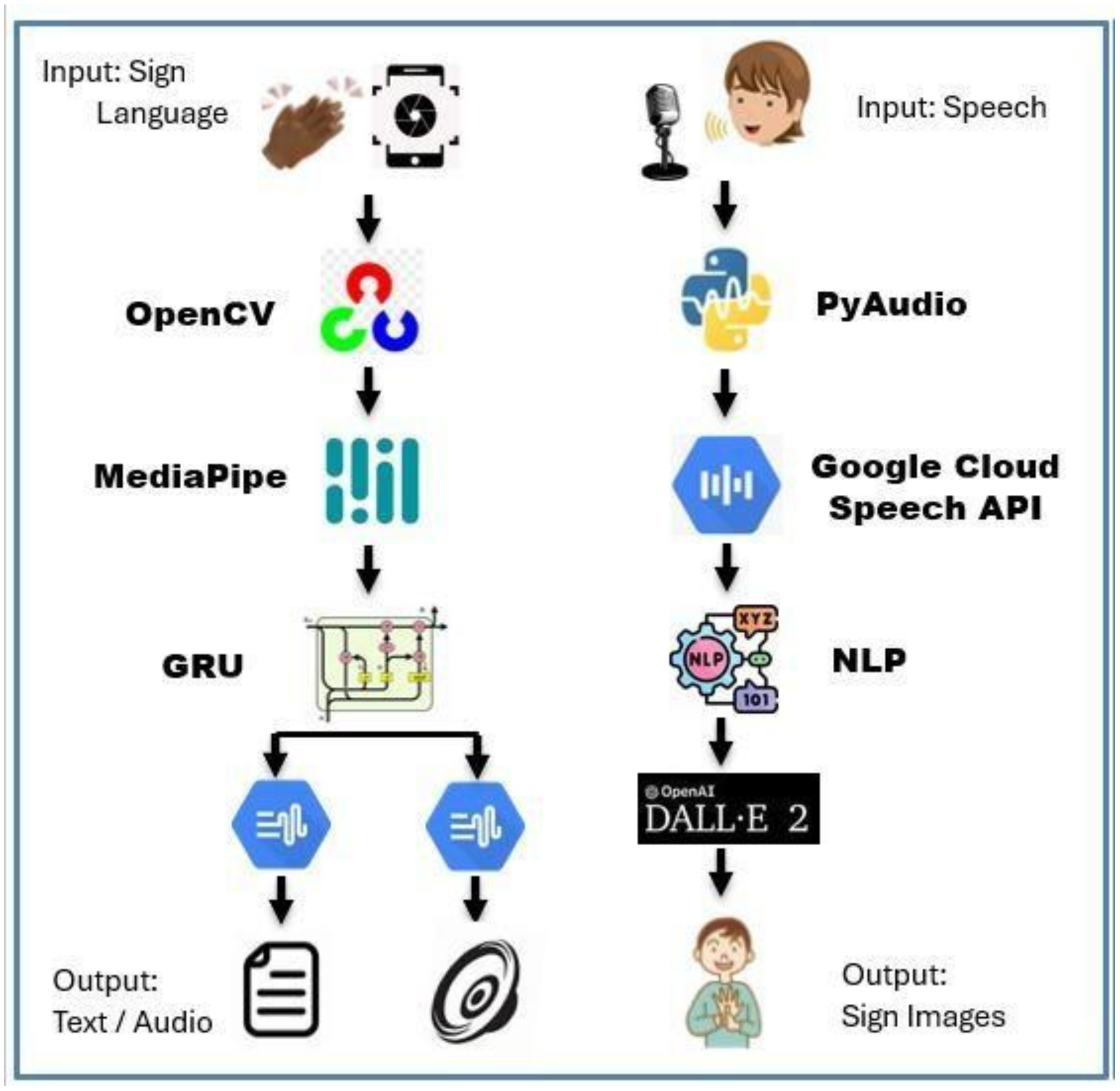


Figure 3.3.2: Working Diagram

This is the proper working diagram of the project this will explain proper flow along with the working of our project.

1. **Sign to Text/speech:**

When a user visits the website, they can select a preferred language. Based on this choice,the system uses **OpenCV** and **MediaPipe** to detect signs in real-time. A GRU model to form sentences, leveraging GRU's ability to manage long- term dependencies, processes detected signs.

Finally, the generated text is converted into audio using **Google Text-to-Speech API**, enabling seamless communication through both text and voice.

2. **Speech To sign:**

In the second module, users can select the "Speech to Sign" option for seamless communication. Using the **PyAudio** module, the system captures the user's real-time voice input in their native language. This audio is then converted into English text usingthe

3. Google Speech-to-Text API:

The translated text is processed to display the corresponding signs, facilitating effective communication between users of different languages. This module ensures inclusivity and ease of interaction by bridging language gaps through speech recognition and sign visualization.

3.4 Algorithms

1. **Random Forest:**

   Random Forest helps in the **Indian Hand Sign to Text and Speech** project by effectively classifying hand gestures detected through the video feed into corresponding text representations.

2. **Gesture Classification:**

   The main goal of the project is to identify Indian Sign Language (ISL) signs and convert them into text and speech. Random Forest is used as the machine-learning model to classifythe extracted features from hand landmarks into distinct sign categories (e.g., letters A-Z, numbers 1-9).

3. **Handling Complex Variability:**

   Random Forest is robust and can handle complex, non-linear relationships in the data. This adaptability is important for recognizing signs that vary due to different hand orientations, lighting conditions, or varying user hand sizes.

4. **Feature Importance:**

   Random Forest provides feature importance scores, helping to identify which landmarks and calculated distances/angles contribute most to accurate gesture classification. This insight can guide further model refinement and feature engineering.

5. **Scalability:**

   Random Forest can scale well with large datasets, which is beneficial for projects that require a diverse range of training data to account for different sign variations and users.

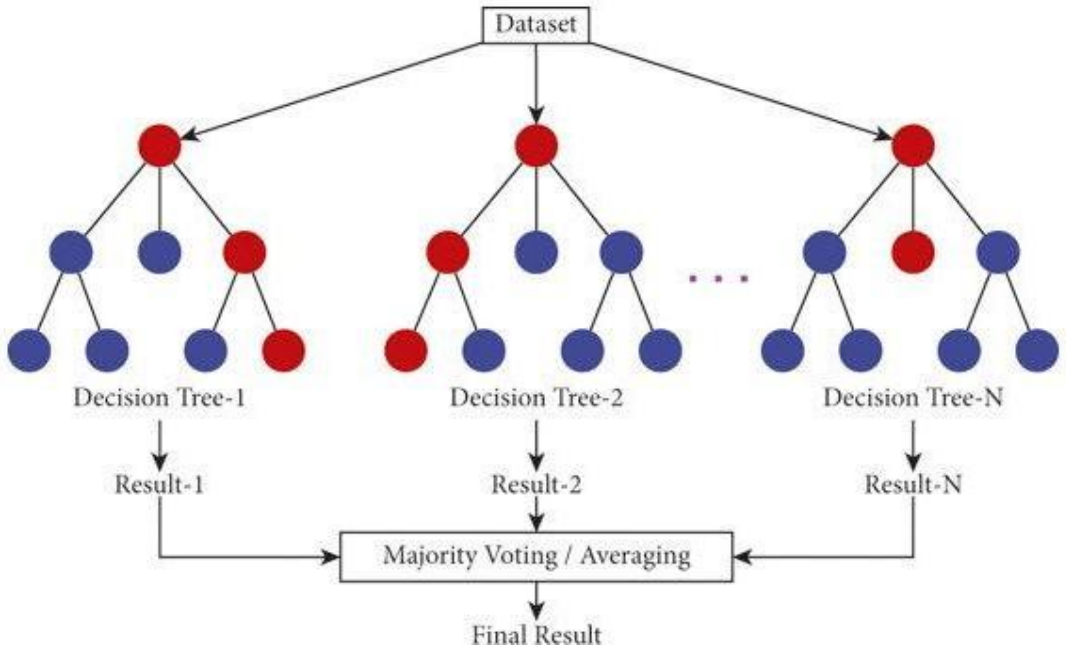


Figure 3.4.1: Random Forest Algorithm

Random Forest adds robust and scalable machine learning capabilities to the backend ofthe "**Indian Hand Sign to Text and Speech"** system, enabling it to accurately and efficiently classify hand gestures in real-time, supporting seamless interaction and communication. A Random Forest model is initialized using the RandomForestClassifier class from **scikit-learn**. The model creates multiple decision trees (e.g., 126 or more, as defined by estimators). Each tree is built using a random subset of the training data, with bootstrapping applied (random sampling with replacement).Each tree splits data based on a randomly selected subset of features, reducing the correlation between trees and improving overall model robustness. Each decision tree learns to classify the input features by splitting the data based on the feature that results in the best separation of classes at each node. The final prediction for each tree is made based on the majority vote of its leaves (the output of the last decision-making node).

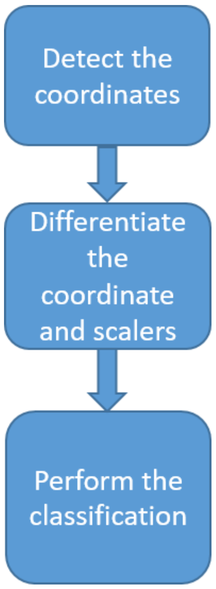

Detect the coordinates
Differentiate the coordinate and scalers
Perform the classification

Figure 3.4.2: Working of Algorithm

Here the algorithm will detect the coordinates set by the MediaPipe framework on each image of each class in the dataset after detection of the coordinates on the dataset the algorithm will differentiation of the coordinates along with the scalers.

After this, the differentiated scalars and the coordinates saved in the dictionary file, which will contain both in the key and value pair and now will be making the classification easier.

Now as the dictionary is ready then the same MediaPipe framework will also be detecting the signs performed by the impaired person and will be showing the result in Real-time.

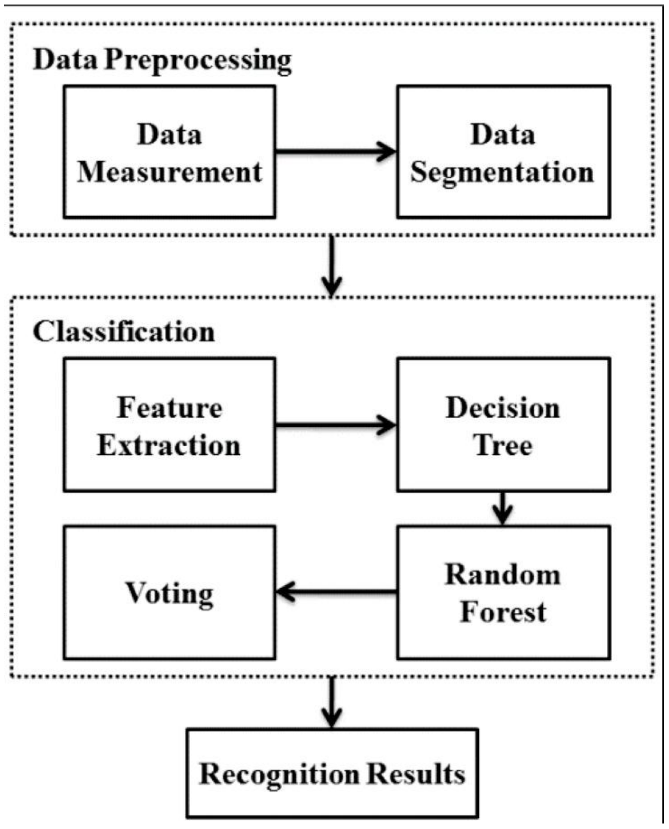


Figure 3.4.3:Algorithm Flow Diagram.

The diagram illustrates Data Preprocessing and Classification. In the Data Preprocessing stage, the process begins with data measurement, where raw data is collected and recorded, such as signals, readings, or images, depending on the context. This raw data is then segmented into meaningful parts or units through data segmentation, preparing it for further analysis.

In the Classification stage, the preprocessed data undergoes feature extraction, where significant and relevant attributes or characteristics are derived from the segmented data. These features are then fed into a decision tree, a basic classification method that categorizes the data based on its features. To enhance accuracy and robustness, the results from multiple decision trees are aggregated using a Random Forest, which is an ensemble learning technique. The outputs of the individual trees are combined through a voting mechanism to make the final classification decision.

Finally, the process concludes with the Recognition Results, where the classified outputs are presented, completing the overall recognition task.

3.5 Technology Stack

1) Frontend Technologies:

- HTML (Hypertext Mark-up Language):

HTML is the standard mark-up language used to create the structure and content ofweb pages. It defines the layout and organization of elements such as text, images, buttons, forms, and videos on the webpage.

- CSS (Cascading Style Sheets):

CSS is used to control the visual presentation and styling of HTML elements, including layout, colours, fonts, and animations..

- JavaScript:

JavaScript is a programming language used to add interactivity and dynamic functionality to web pages

2) Backend Technologies:

- Python:

The primary programming language for implementing machine-learning models and handling the backend logic.

- Flask:

Backend framework for handling API requests and connecting the frontend with the machine learning models.

3) Machine Learning Frameworks:

- Media Pipe:

Media Pipe is a powerful framework for real-time perception tasks, developed by Google, which offers tools for detecting and tracking body, hand, and facial landmarks.

- Random Forest:

Random Forest helps in the "**Indian Hand Sign to Text and Speech"** project by effectively classifying hand gestures detected through the video feed into corresponding text representations

Chapter IV

RESULT AND DISCUSSION

4.1 Performance Metric

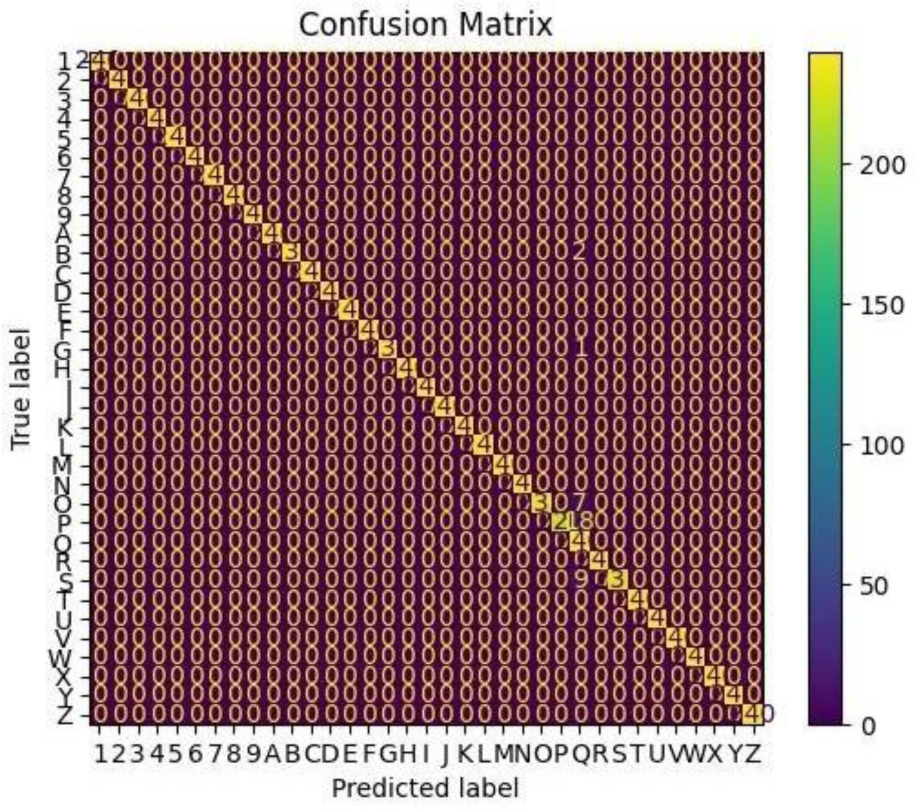


Figure 4.1.1: Confusion Metrics

The above metrics (i.e. Figure 4.1.1) will show the confusion metrics of the model, which is using the MediaPipe framework. This metrics showing the detection confidence level of each class of the Indian Hand Signs.

Both axis that is the x axis will elaborate us about the predicted label and the y axis will also elaborate us about the True labels and the highlighted area will tell us about the confidence levels of the model about each signs.

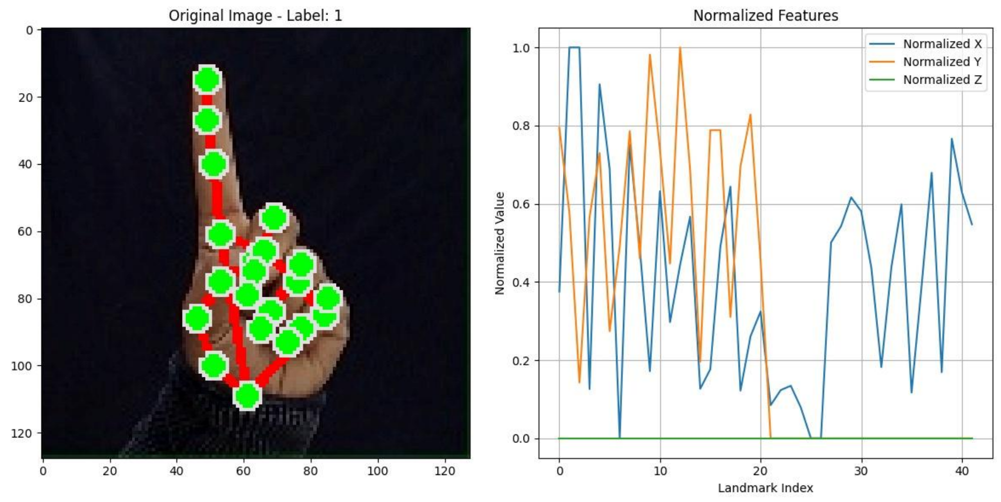


Figure 4.1.2: Normalized Graph of 1

The tracking points on the sign image are displayed in the above figure. The 48-point tracking system was used to track the hand gesture using Media Point to identify the data points.

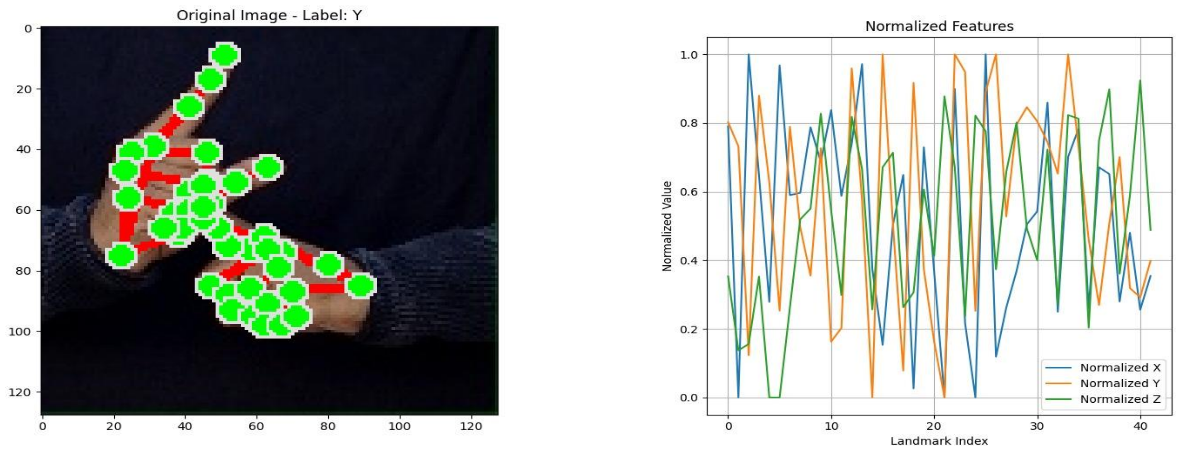


Figure 4.1.3: Normalized Graph of letter Y

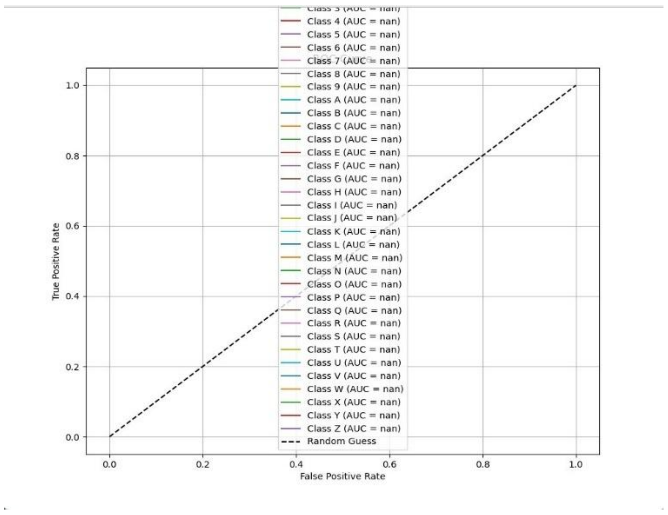


Figure 4.1.4: AUC curve

The AUC (area under the curve) above displays the label with the corresponding letter. Here, I've checked the relationship with each label from the plot above to make sure we understand the relationships with each sign and its true positive rate.

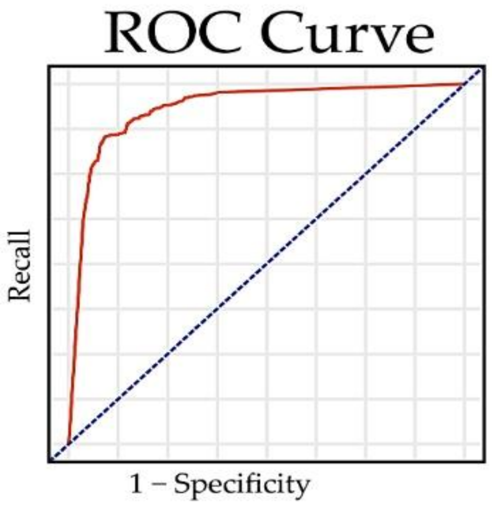
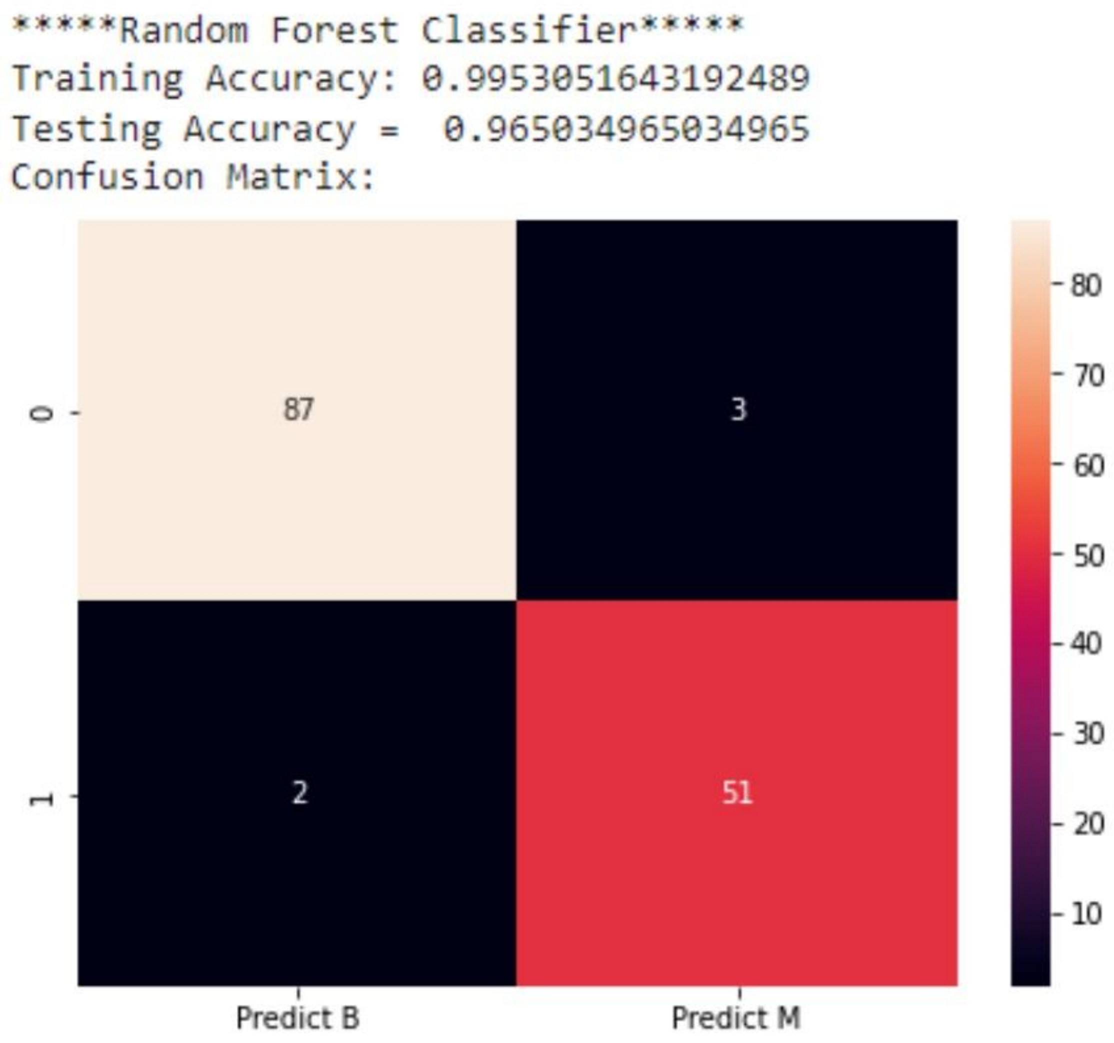


Figure 4.1.5: ROC Curve

The above ROC curve shows the relationship between the Recall vs Specificity . this shows that how many classes are actually predicted correctly by the model among all classes and how much classes are negatively or positively classified.



```
*****Random Forest Classifier*****
Training Accuracy: 0.9953051643192489
Testing Accuracy =  0.965034965034965
Confusion Matrix:
```

Figure 4.1.6: Confusion Matrix.

The above figure illustrate prediction on test image letters ,we have tested the model for two letter one for B and other for M and getting good accuracy after training on multiple images. As shown in the figure the True positive and False Positive rate of the letter the model is predicted the exactly same letter as training on it.

4.2 Dataset Descriptions:

The dataset consists of 35 distinct classes, encompassing 9 numerical digits (1 through 9) and 26 alphabetic characters (A through Z). In total, the dataset contains 42,000 images, with an equal distribution of 1,200 samples per class. Each image is formatted with dimensions of 128×128 pixels, ensuring consistency in size across the dataset. The imagesare in RGB color format, comprising three channels to capture a full spectrum of colors.

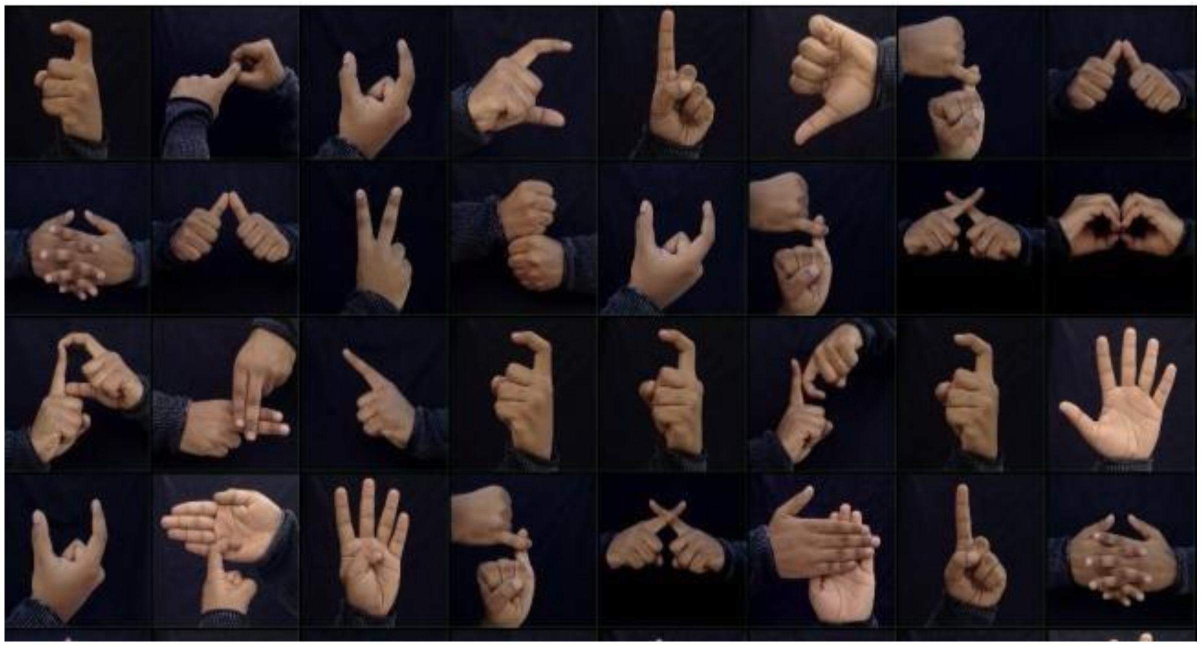


Figure 4.2.1:Different Sign images.

4.3 Results:

1.Homepage:

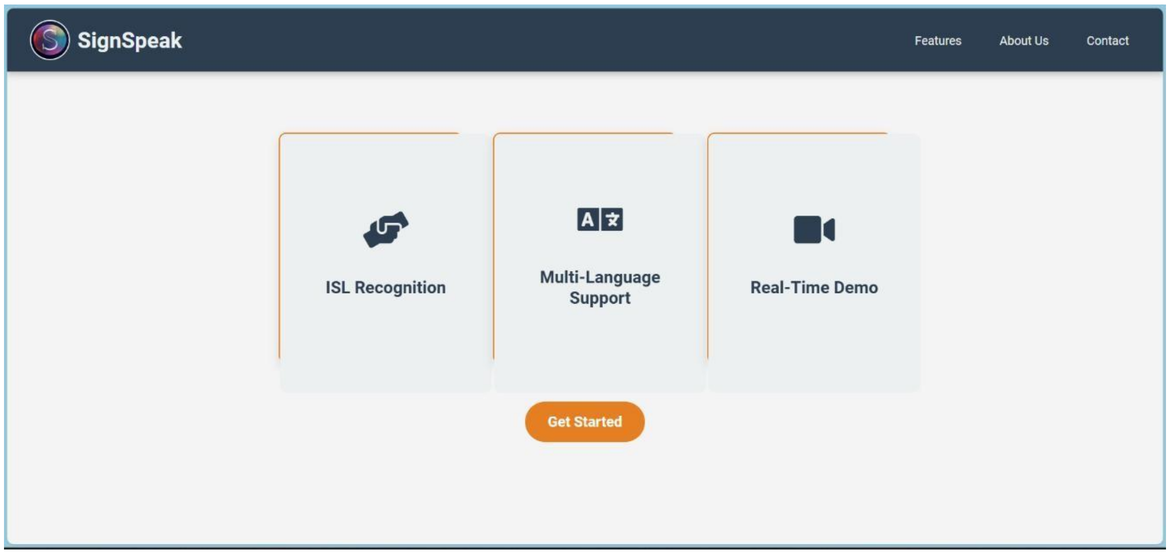


Figure 4.3.1: Homepage

This is the home page of the website. Once the user visit the websites he will find the get started button once he click on this direct to next page of the website. Where he will get the two option Indian sign to text and speech to sign.

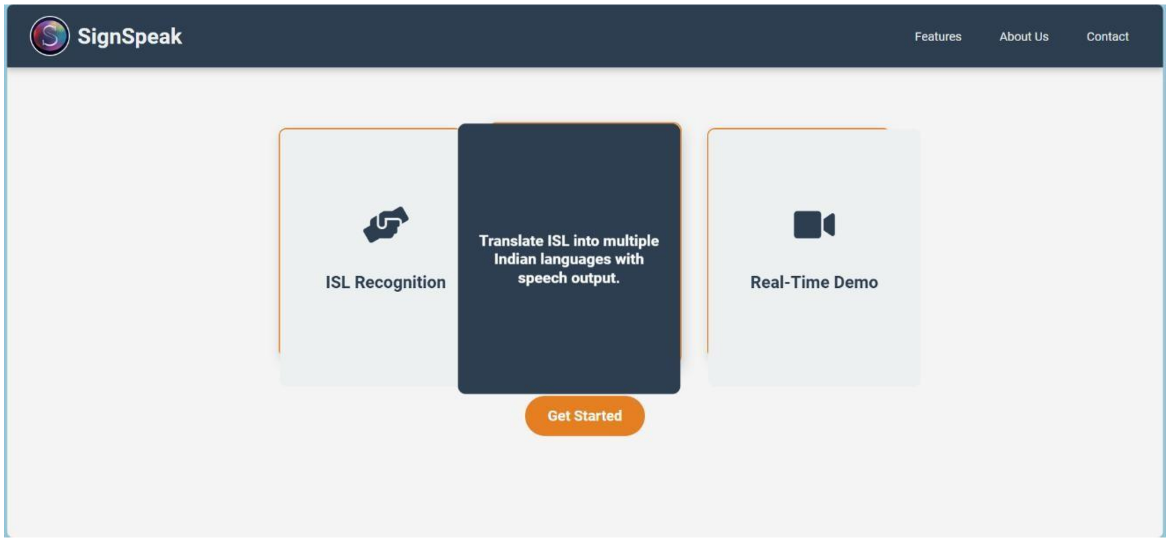


Figure 4.3.2: Homepage of Web-App

2.modules:

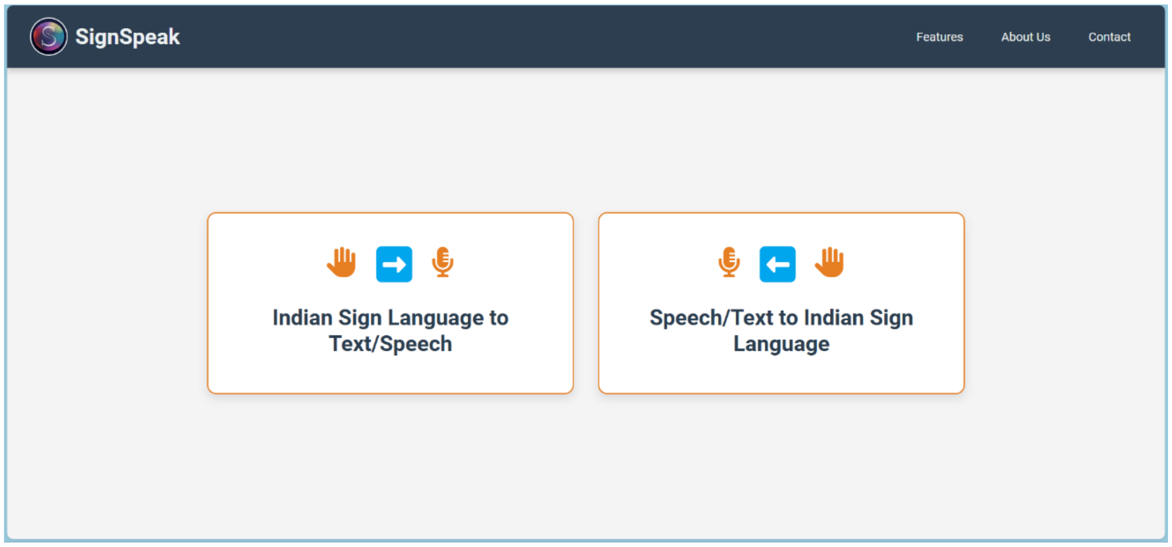
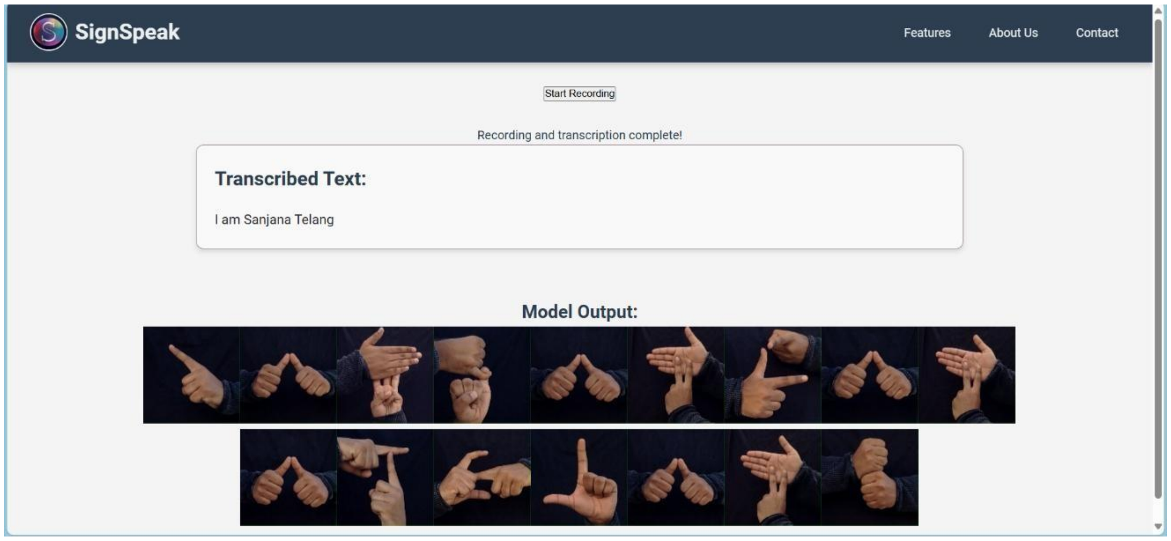

SignSpeak
Features
About Us
Contact
Indian Sign Language to Text/Speech
Speech/Text to Indian Sign Language

Figure 4.3.3: Module Page

This is second page of the website where he will get the two option Indian sign language to text and vice versa option. Once he will click on one of the button will direct to next page of the modules



Recording and transcription complete!

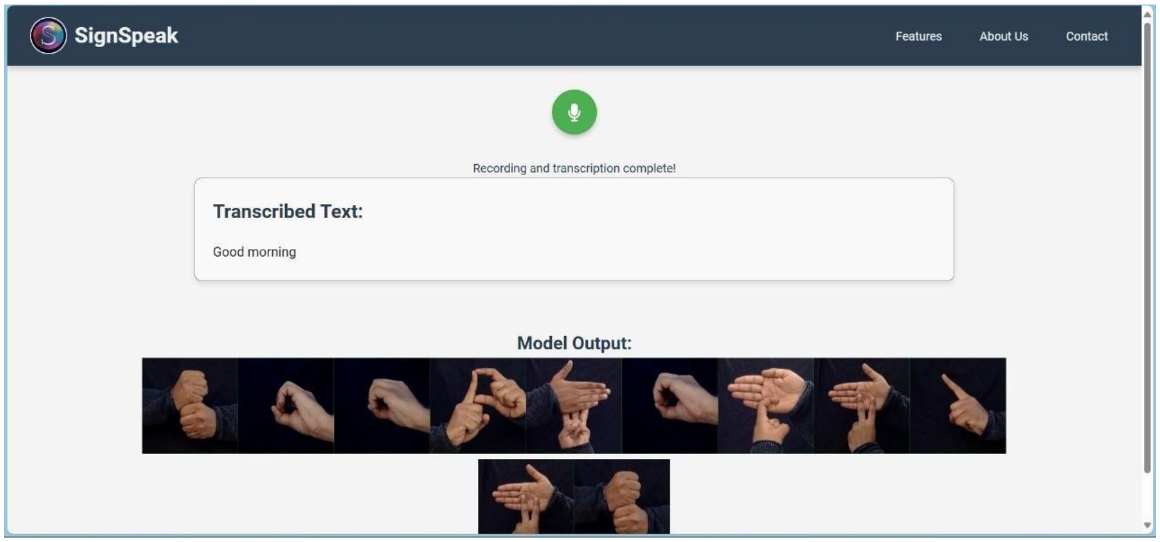
Transcribed Text:

I am Sanjana Telang

Model Output:

Figure 4.3.4:Speech To Indian Sign Language

The diagram above illustrates how the model processes spoken words in real time in their native tongue, converts them into English for context, and outputs corresponding sign images.



Recording and transcription complete!

Transcribed Text:

Good morning

Figure 4.3.5: Module Speech To Indian Sign Language Page

In this module user will provide the speech in real time using mic button .user can provide speech into different regional-native languages in real time that can be provide to model it will provide corresponding sign image in output window.

Table 4.4: Model Accuracy and Loss

Model	**Accuracy**	**Loss**
CNN	78.8%	0.269
Ada Boost	80%	0.22
Random Forest	88%	0.15
VG classifier	84%	0.19

This table lists every model used for the project along with the accuracy and loss function that go with it. We tried a number of models; at first, we trained the CNN architecture model, which produced very poor accuracy; then, we tried other training methods to improve accuracy.

We will retain the final model as random forest, which we are using in real time to predict the words in the sequence, after iterating through several models and discovering that the random forest algorithm produces more accuracy than any other model.

Chapter V
CONCLUSION

5.1 Conclusion

In conclusion, the Indian Sign Language translation application is a transformative tool that addresses the pressing need for effective communication between hearing and deaf individuals. By enabling real-time recognition of ISL gestures and translating them into text or speech, the application fosters greater social inclusion and reduces the isolation experienced by many in the deaf community. Its user-centric design not only enhances accessibility but also encourages active participation in educational and professional environments.

5.2 Future Scope:

The future scope of this project is expansive. Enhancements can be made to broaden the vocabulary of recognized signs, incorporating more complex phrases and expressions that reflect everyday communication needs. Future iterations could also include:

Integration with Augmented Reality (AR), users couldvisualize signs in their environment, making learning and communication more intuitive.

Facial Expression Recognition: Incorporating facial cues into the gesture recognition process would allow for a more nuanced understanding of sign language, as facialexpressions are integral to conveying meaning in ISL.

Multi-Language Support: Expanding the application to support multiple sign languages across different regions would promote inclusivity on a larger scale.

Community Engagement: Developing platforms for users to contribute to the dataset by sharing new signs or regional variations would enhance the application's adaptability and relevance.

References:

[1]Ming Jin Cheok, Zaid Omar, and Mohamed Hisham Jaward. A review of hand gestureand sign language recognition techniques. International Journal of Ma- chine Learning and Cybernetics, 10(1):131–153, 2019.

[2]Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. Sign language recognition using convolutional neural networks. In European conference on computer vision, pages 572–578. Springer, 2014.

[3]Ankita Wadhawan and Parteek Kumar. Deep learning-based sign language recognitionsystem for static signs. Neural computing and applications, 32(12):7957–7968, 2020.

[4]By great learning team ,"Real-Time Object Detection Using TensorFlow", december 25 ,2021 https://www.mygreatlearning.com/blog/object-detection-usingtensorflow/

[5]Jeffrey Dean, minute 0:47 / 2:17 from YouTube clip "TensorFlow: Open source machine learning". Google. 2015. Archived from the original on November 11, 2021."It is machine learning software being used for various kinds of perceptual and language understanding tasks.

[6]Li, Y.; Chen, X.; Zhang, X.; Wang, K.; Wang, Z.J. A sign-component-based framework for Chinese sign language recognition using accelerometer and sEMG data. IEEE Trans. Biomed. Eng. 2012, 59, 2695–2704. [Google Scholar] [PubMed]

[7]Ding, L.; Martinez, A.M. Modelling and recognition of the linguistic components in American sign language. Image Vis. Comput. 2009, 27, 1826–1844. [Google Scholar] [CrossRef] [PubMed]

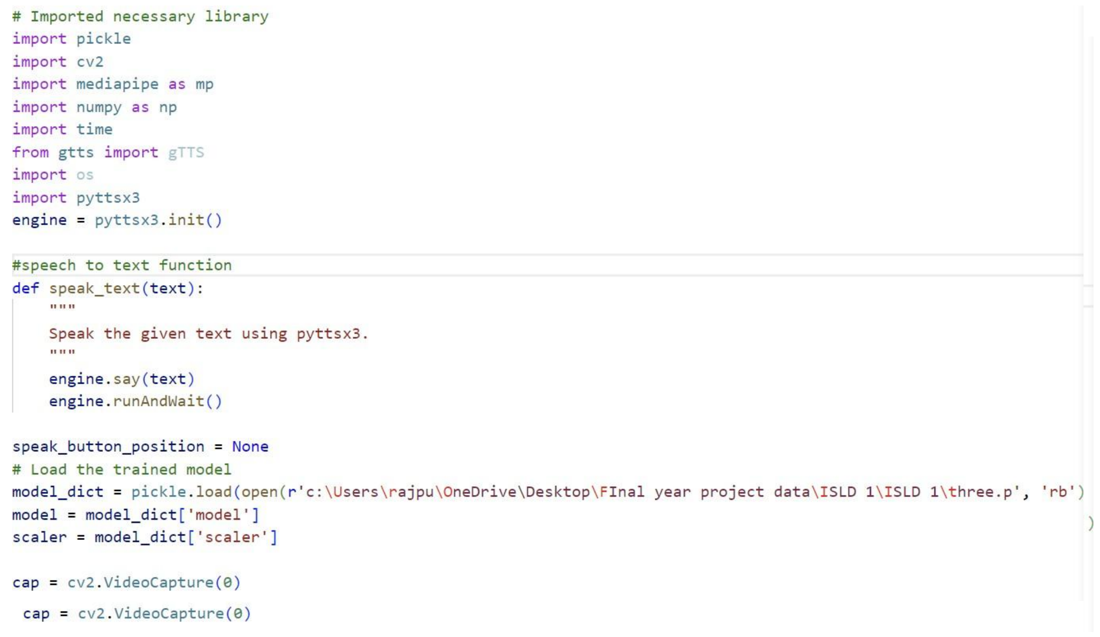
[8]Trigueiros, P.; Ribeiro, F.; Reis, L.P. Vision-based Portuguese sign language recognition system. Adv. Intell. Syst. 2014, 275, 605–617. [Google Scholar]

[9]Fang, G.; Gao, W.; Zhao, D. Large vocabulary sign language recognition based on fuzzy decision trees. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 2004, 34, 305–314. [Google Scholar] [CrossRef]

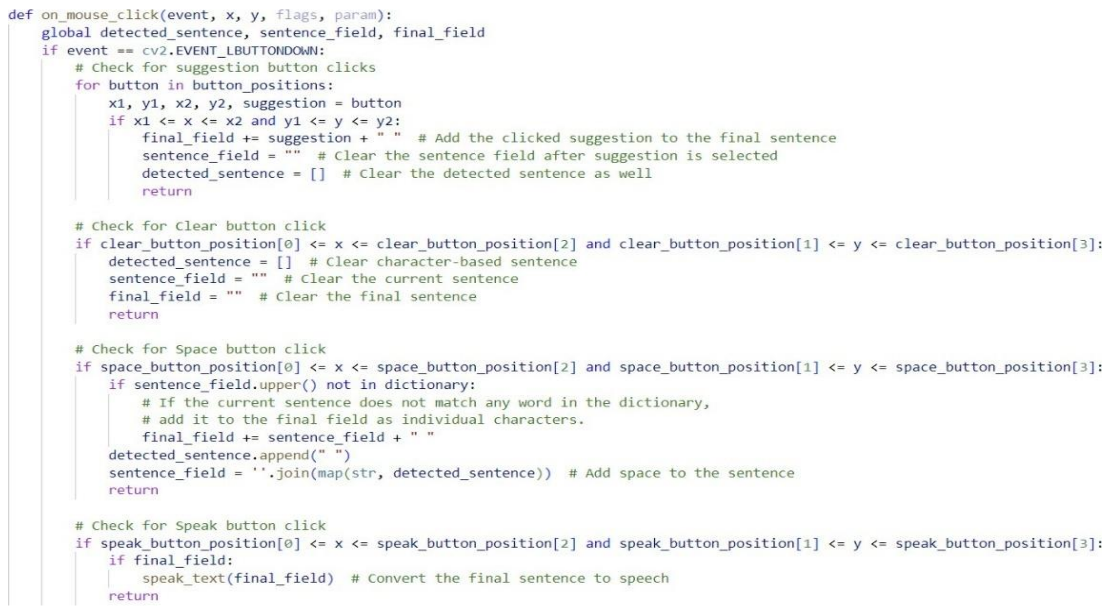
[10] Doliotis, P.; Athitsos, V.; Kosmopoulos, D.; Perantonis, S. Hand Shape and 3D Pose Estimation Using Depth Data From a Single Cluttered Frame. In Advances in Visual Computing—8th International Symposium, ISVC 2012.

Annexures

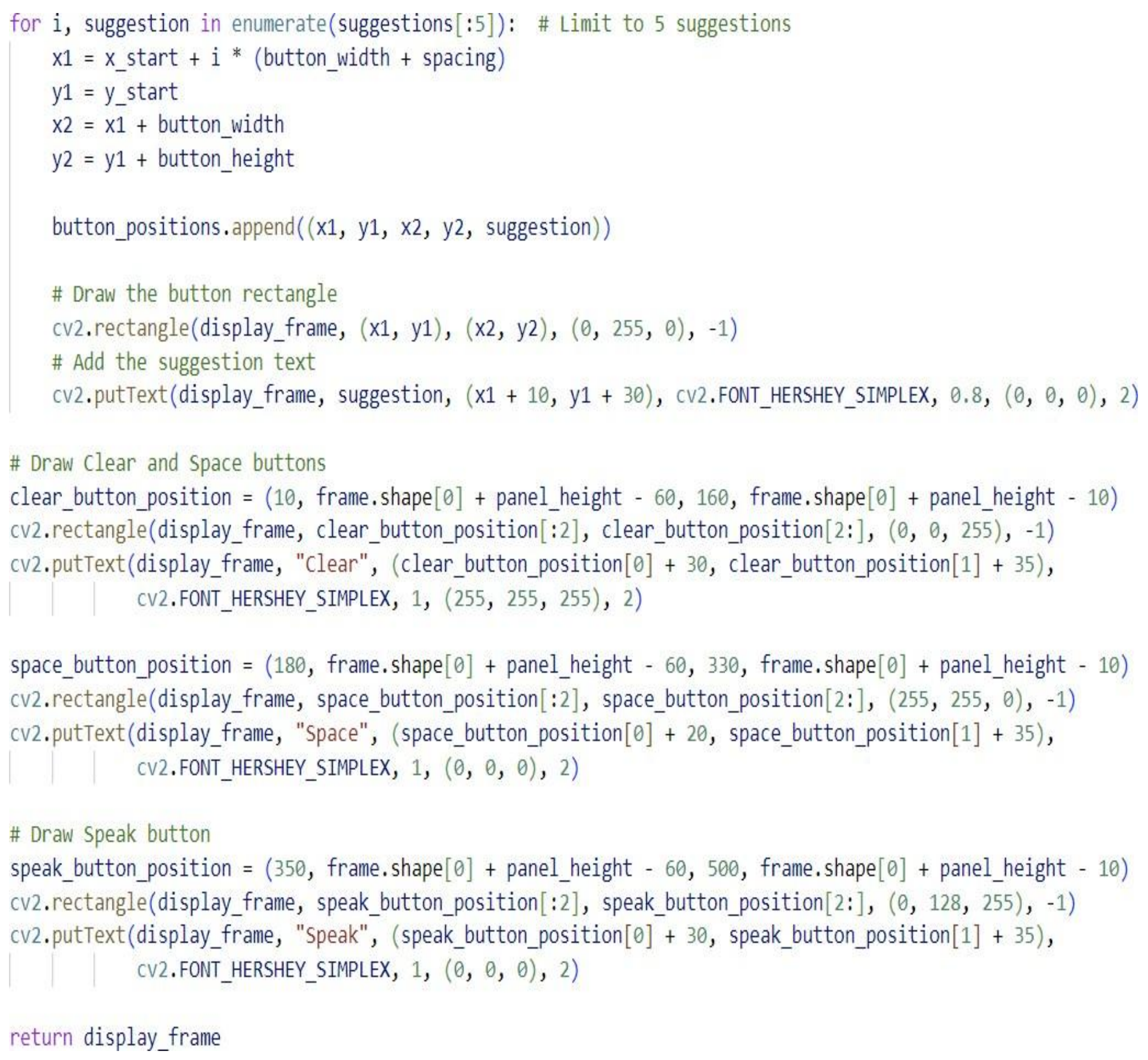
Annexure A:



```python
# Imported necessary library
import pickle
import cv2
import mediapipe as mp
import numpy as np
import time
from gtts import gTTS
import os
import pyttsx3
engine = pyttsx3.init()

#speech to text function
def speak_text(text):
    """
    Speak the given text using pyttsx3.
    """
    engine.say(text)
    engine.runAndWait()

speak_button_position = None
# Load the trained model
model_dict = pickle.load(open(r'c:\Users\rajpu\OneDrive\Desktop\FInal year project data\ISLD 1\ISLD 1\three.p', 'rb'))
model = model_dict['model']
scaler = model_dict['scaler']

cap = cv2.VideoCapture(0)

 cap = cv2.VideoCapture(0)
```

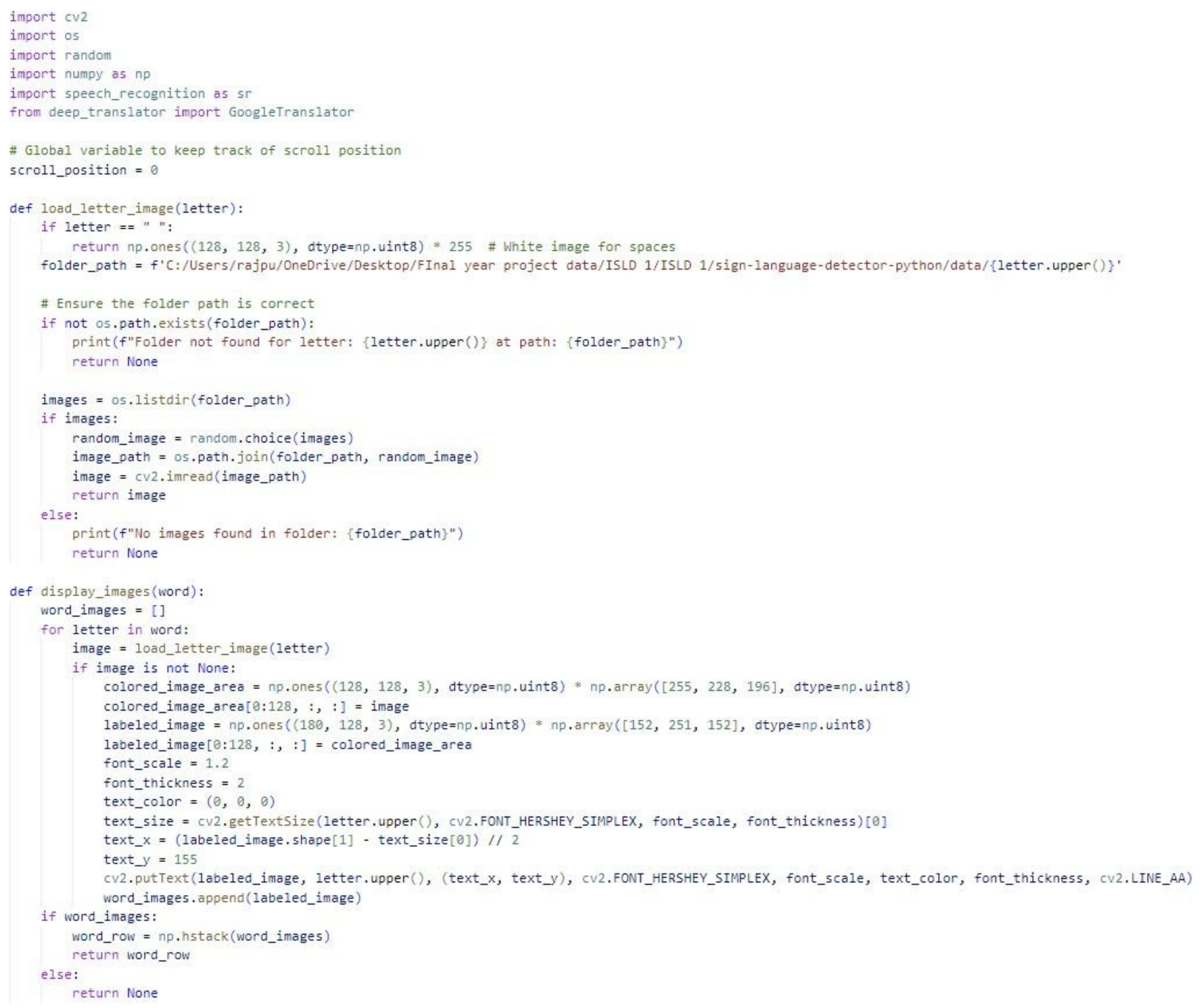


```python
def on_mouse_click(event, x, y, flags, param):
    global detected_sentence, sentence_field, final_field
    if event == cv2.EVENT_LBUTTONDOWN:
        # Check for suggestion button clicks
        for button in button_positions:
            x1, y1, x2, y2, suggestion = button
            if x1 <= x <= x2 and y1 <= y <= y2:
                final_field += suggestion + " "  # Add the clicked suggestion to the final sentence
                sentence_field = ""  # Clear the sentence field after suggestion is selected
                detected_sentence = []  # Clear the detected sentence as well
                return

        # Check for Clear button click
        if clear_button_position[0] <= x <= clear_button_position[2] and clear_button_position[1] <= y <= clear_button_position[3]:
            detected_sentence = []  # Clear character-based sentence
            sentence_field = ""  # Clear the current sentence
            final_field = ""  # Clear the final sentence
            return

        # Check for Space button click
        if space_button_position[0] <= x <= space_button_position[2] and space_button_position[1] <= y <= space_button_position[3]:
            if sentence_field.upper() not in dictionary:
                # If the current sentence does not match any word in the dictionary,
                # add it to the final field as individual characters.
                final_field += sentence_field + " "
            detected_sentence.append(" ")
            sentence_field = ''.join(map(str, detected_sentence))  # Add space to the sentence
            return

        # Check for Speak button click
        if speak_button_position[0] <= x <= speak_button_position[2] and speak_button_position[1] <= y <= speak_button_position[3]:
            if final_field:
                speak_text(final_field)  # Convert the final sentence to speech
            return
```

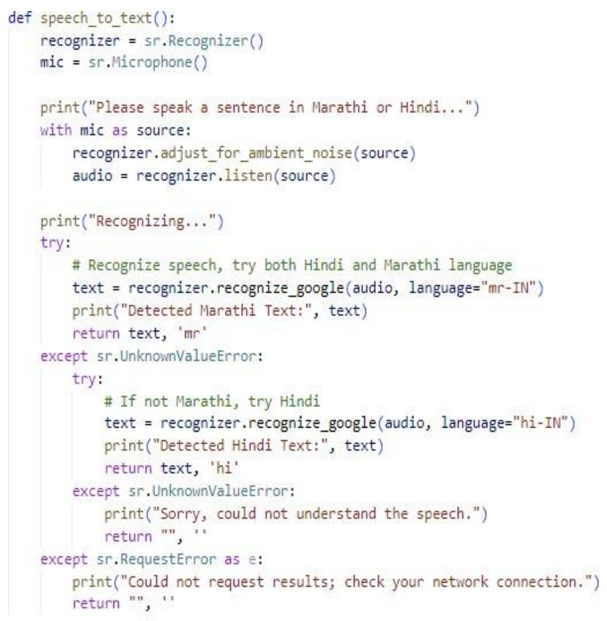


```
for i, suggestion in enumerate(suggestions[:5]):  # Limit to 5 suggestions
    x1 = x_start + i * (button_width + spacing)
    y1 = y_start
    x2 = x1 + button_width
    y2 = y1 + button_height

    button_positions.append((x1, y1, x2, y2, suggestion))

    # Draw the button rectangle
    cv2.rectangle(display_frame, (x1, y1), (x2, y2), (0, 255, 0), -1)
    # Add the suggestion text
    cv2.putText(display_frame, suggestion, (x1 + 10, y1 + 30), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 0), 2)

# Draw Clear and Space buttons
clear_button_position = (10, frame.shape[0] + panel_height - 60, 160, frame.shape[0] + panel_height - 10)
cv2.rectangle(display_frame, clear_button_position[:2], clear_button_position[2:], (0, 0, 255), -1)
cv2.putText(display_frame, "Clear", (clear_button_position[0] + 30, clear_button_position[1] + 35),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)

space_button_position = (180, frame.shape[0] + panel_height - 60, 330, frame.shape[0] + panel_height - 10)
cv2.rectangle(display_frame, space_button_position[:2], space_button_position[2:], (255, 255, 0), -1)
cv2.putText(display_frame, "Space", (space_button_position[0] + 20, space_button_position[1] + 35),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)

# Draw Speak button
speak_button_position = (350, frame.shape[0] + panel_height - 60, 500, frame.shape[0] + panel_height - 10)
cv2.rectangle(display_frame, speak_button_position[:2], speak_button_position[2:], (0, 128, 255), -1)
cv2.putText(display_frame, "Speak", (speak_button_position[0] + 30, speak_button_position[1] + 35),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)

return display_frame
```

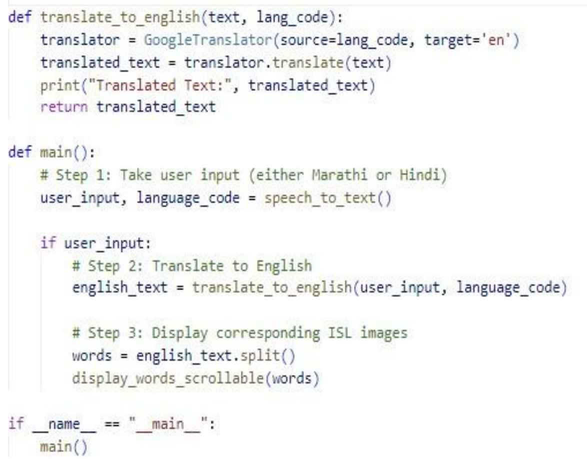
Annexure B:



```python
import cv2
import os
import random
import numpy as np
import speech_recognition as sr
from deep_translator import GoogleTranslator

# Global variable to keep track of scroll position
scroll_position = 0

def load_letter_image(letter):
    if letter == " ":
        return np.ones((128, 128, 3), dtype=np.uint8) * 255  # White image for spaces
    folder_path = f'C:/Users/rajpu/OneDrive/Desktop/FInal year project data/ISLD 1/ISLD 1/sign-language-detector-python/data/{letter.upper()}'

    # Ensure the folder path is correct
    if not os.path.exists(folder_path):
        print(f"Folder not found for letter: {letter.upper()} at path: {folder_path}")
        return None

    images = os.listdir(folder_path)
    if images:
        random_image = random.choice(images)
        image_path = os.path.join(folder_path, random_image)
        image = cv2.imread(image_path)
        return image
    else:
        print(f"No images found in folder: {folder_path}")
        return None

def display_images(word):
    word_images = []
    for letter in word:
        image = load_letter_image(letter)
        if image is not None:
            colored_image_area = np.ones((128, 128, 3), dtype=np.uint8) * np.array([255, 228, 196], dtype=np.uint8)
            colored_image_area[0:128, :, :] = image
            labeled_image = np.ones((180, 128, 3), dtype=np.uint8) * np.array([152, 251, 152], dtype=np.uint8)
            labeled_image[0:128, :, :] = colored_image_area
            font_scale = 1.2
            font_thickness = 2
            text_color = (0, 0, 0)
            text_size = cv2.getTextSize(letter.upper(), cv2.FONT_HERSHEY_SIMPLEX, font_scale, font_thickness)[0]
            text_x = (labeled_image.shape[1] - text_size[0]) // 2
            text_y = 155
            cv2.putText(labeled_image, letter.upper(), (text_x, text_y), cv2.FONT_HERSHEY_SIMPLEX, font_scale, text_color, font_thickness, cv2.LINE_AA)
            word_images.append(labeled_image)
    if word_images:
        word_row = np.hstack(word_images)
        return word_row
    else:
        return None
```



```
def speech_to_text():
    recognizer = sr.Recognizer()
    mic = sr.Microphone()

    print("Please speak a sentence in Marathi or Hindi...")
    with mic as source:
        recognizer.adjust_for_ambient_noise(source)
        audio = recognizer.listen(source)

    print("Recognizing...")
    try:
        # Recognize speech, try both Hindi and Marathi language
        text = recognizer.recognize_google(audio, language="mr-IN")
        print("Detected Marathi Text:", text)
        return text, 'mr'
    except sr.UnknownValueError:
        try:
            # If not Marathi, try Hindi
            text = recognizer.recognize_google(audio, language="hi-IN")
            print("Detected Hindi Text:", text)
            return text, 'hi'
        except sr.UnknownValueError:
            print("Sorry, could not understand the speech.")
            return "", ''
    except sr.RequestError as e:
        print("Could not request results; check your network connection.")
        return "", ''
```



```
def translate_to_english(text, lang_code):
    translator = GoogleTranslator(source=lang_code, target='en')
    translated_text = translator.translate(text)
    print("Translated Text:", translated_text)
    return translated_text

def main():
    # Step 1: Take user input (either Marathi or Hindi)
    user_input, language_code = speech_to_text()

    if user_input:
        # Step 2: Translate to English
        english_text = translate_to_english(user_input, language_code)

        # Step 3: Display corresponding ISL images
        words = english_text.split()
        display_words_scrollable(words)

if __name__ == "__main__":
    main()
```