Sign language to text conversion

Introduction:

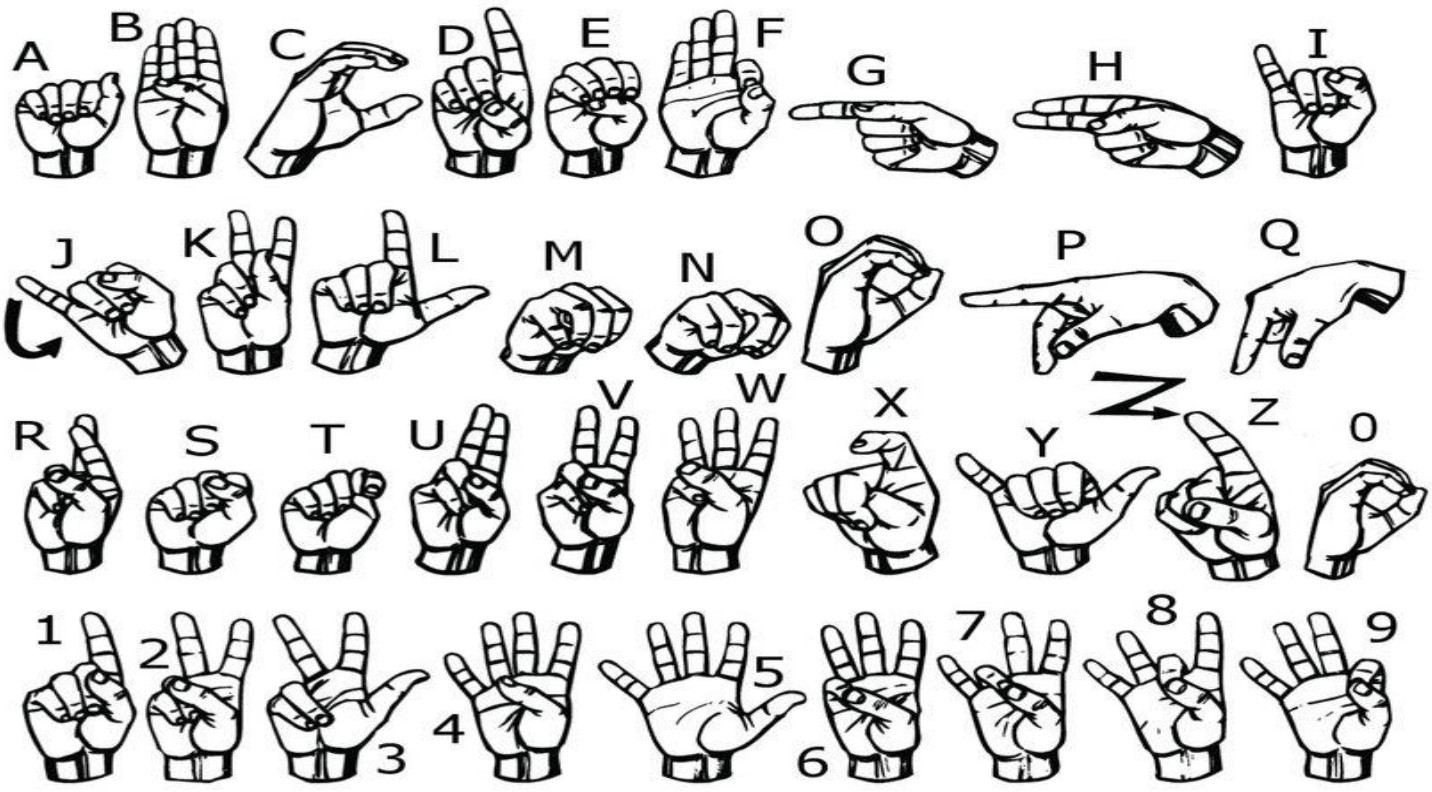
American sign language is a predominant sign language Since the only disability Deaf and Dumb (hereby referred to as D&M) people have is communication related and since they cannot use spoken languages, the only way for them to communicate is through sign language. Communication is the process of exchange of thoughts and messages in various ways such as speech, signals, behavior and visuals. D&M people make use of their hands to express different gestures to express their ideas with other people. Gestures are the non-verbally exchanged messages and these gestures are understood with vision. This nonverbal communication of deaf and dumb people is called sign language. A sign language is a language which uses gestures instead of sound to convey meaning combining hand-shapes, orientation and movement of the hands, arms or body, facial expressions and lip-patterns. Contrary to popular belief, sign language is not international. These vary from region to region.

Sign language is a visual language and consists of 3 major components

Fingerspelling	Word level sign vocabulary	Non-manual features
Used to spell words letter by letter .	Used for the majority of communication.	Facial expressions and tongue, mouth and body position.

Minimizing the verbal exchange gap among D&M and non-D&M people turns into a want to make certain effective conversation among all. Sign language translation is among one of the most growing lines of research and it enables the maximum natural manner of communication for those with hearing impairments. A hand gesture recognition system offers an opportunity for deaf people to talk with vocal humans without the need of an interpreter. The system is built for the automated conversion of ASL into textual content and speech.

In our project we primarily focus on producing a model which can recognize Fingerspelling based hand gestures in order to form a complete word by combining each gesture. The gestures we aim to train are as given in the image below.



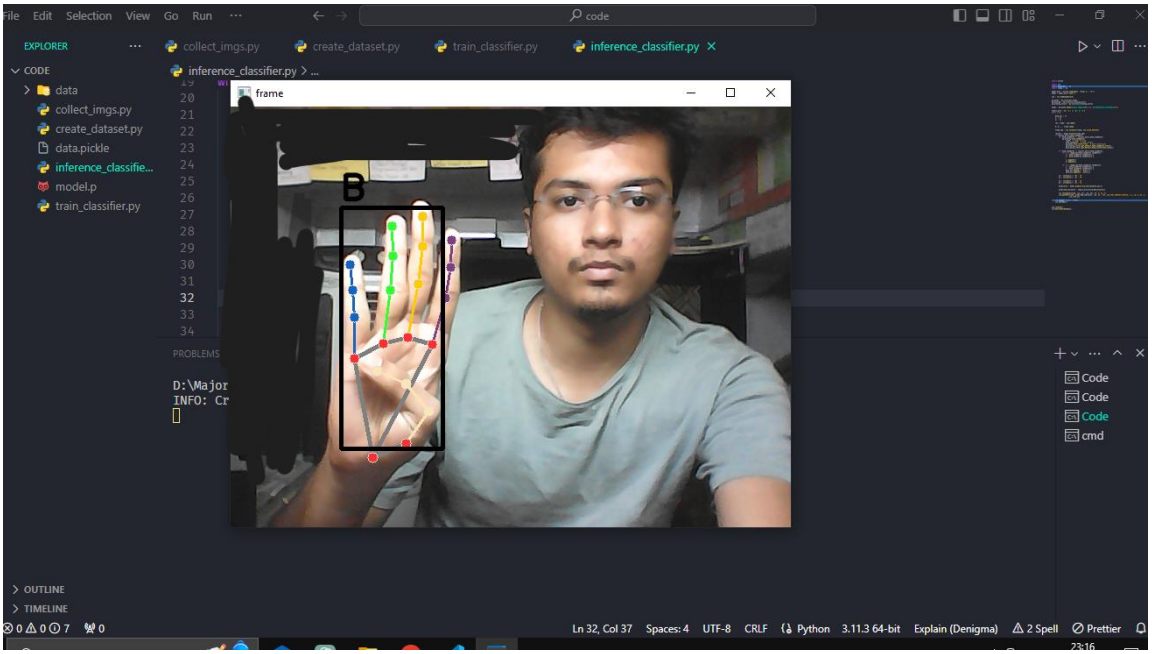
Methodology:

1. Data collection
   For the project we tried to find already made datasets but we couldn't find dataset in the form of raw images that matched our requirements. All we could find were the datasets in the form of RGB values. Hence, we decided to create our own data set. Steps we followed to create our data set are as follows.
   We used Open computer vision (OpenCV) library in order to produce our dataset. Firstly, we captured around 100 images for A, B and C symbol in ASL (American Sign Language)

2. Data processing
   Video to Image Conversion: For video-based data, extract frames (images) from the videos. This is often necessary because many sign language recognition models work with still images.
   Image Enhancement: Enhance the image quality if necessary, which may include denoising, contrast adjustment, and image resizing.
   Hand Detection: Use computer vision techniques to detect and isolate the signer's hand within the images. This step is essential for isolating the hand from the background.
   Hand Landmark Detection: Employ methods like MediaPipe to detect and track hand landmarks or keypoints within the signer's hand. These landmarks represent specific points on the hand, such as fingertips and knuckles, that are used for sign recognition.
   Data Augmentation: Create variations of your dataset by applying transformations like rotation, translation, and flipping to increase dataset diversity and robustness.
   Data labeling: Label the extracted features with their corresponding sign language glosses or spoken language translations. These labels serve as the target output for training and testing.

3. Train model
   Then we build a random forest classifier for a hand gesture recognition task. It loads previously extracted hand landmark data, splits it into training and testing sets, trains a random forest classifier, evaluates its accuracy and saves the trained model to a file. Pickle module is used to load and save python objects as serialized binary data. Scikit-learn is used for building and evaluating the random forest classifier, as well as for splitting the dataset. Numpy is used for numerical operations. Split the data into training and testing sets using scikit-learns train_test_split function. Here 80% of the data is used for training (x_train, y_train) and 20% is used for testing (x_test, y_test) the stratify parameter ensures that the class distribution is preserved in the split. Serialize and save the trained random forest model to a file called 'model.p' using pickle. This allows to use the trained model for inference without retraining it.

4. Testing
   Then we use a pre trained model to classify hand gesture captured from a webcam in real time using the MediaPipe library. Our model captures frames from the webcame, detects hand landmarks, predicts gestures, and displays the results in real time.



Conclusion:

As we have created a demo of our model we are exploring different approaches and trying it to make our model stand out with accurate results and we are expanding our dataset as well once all the work is done we will be creating a block diagram of our UI and the expand the dataset till Z and we will explore more sign languages and used more advance approaches to finish our work.