A Software Group Project Report

On

AI-Based Real-Time Sign Language to Voice Translator

Submitted by

Karan Sinh R Chauhan (IU2341052319)
Yomin Navadiya (IU2341052291)
Nisarg Modi (IU2241050142)
Daivik Sutaria (IU2241050253)

Guided by

Professor Pooja Sisodiya Mam

In partial fulfillment for the award of the degree Of

BACHELOR OF TECHNOLOGY

In

Computer Engineering

INSTITUTE OF TECHNOLOGY AND ENGINEERING INDUS UNIVERSITY

CAMPUS, RANCHARDA, VIA-THALTEJ

AHMEDABAD-382115, GUJARAT, INDIA,

WEB: www.indusuni.ac.in

Nov-Dec 2025

Table Of Content

LIST OF FIGURES

ACKNOWLEDGEMENT

We would like to express our profound gratitude to everyone who contributed to the successful completion of this project. Their encouragement, cooperation, and invaluable guidance have been instrumental in shaping our work.

First and foremost, we wish to convey our sincere appreciation to our project guide, Professor Pooja Sisodiya, for her constant support, motivation, and insightful feedback throughout the course of this work. Her expertise and mentorship have been an inspiration and have guided us through every challenge we encountered.

We also extend our thanks to the Department of Computer Engineering, Indus University, for providing us with a conducive environment, essential resources, and encouragement to carry out this project successfully.

We are deeply grateful to our family and friends for their patience, moral support, and continuous encouragement, which motivated us to push through difficult stages of the research and implementation process.

Finally, we acknowledge all the authors, developers, and researchers whose work and open-source libraries in the fields of Artificial Intelligence, Computer Vision, and Natural Language Processing served as invaluable references and tools during our project development.

This project has not only enhanced our technical knowledge but also taught us the value of teamwork, perseverance, and innovation. We sincerely thank everyone who made this project a meaningful and enriching experience.

ABSTRACT

The AI-Based Real-Time Sign Language to Voice Translator is designed to bridge the communication gap between the hearing-impaired community and non-signers by converting sign gestures into audible speech and text in real time. The system leverages advancements in computer vision, machine learning, and text-to-speech synthesis to interpret hand gestures captured through a webcam and translate them into corresponding spoken language.

Using the MediaPipe framework for hand landmark detection and a Convolutional Neural Network (CNN) for gesture classification, the proposed model achieves robust recognition of static sign language alphabets even under varying lighting and background conditions. Post recognition, the text output is converted into speech using the pyttsx3 library, providing an intuitive and interactive user experience.

The application is lightweight, scalable, and designed to support multiple sign languages and regional speech outputs in future versions. This research demonstrates that combining AI-driven vision systems with natural language generation can create a powerful assistive communication tool for the deaf and hard-of-hearing communities. The system aims to promote inclusivity, accessibility, and social integration by enabling seamless two-way interaction between signers and non-signers.

Chapter 1:
Introduction

American sign language is a predominant sign language Since the only disability D&M people have is communication related and they cannot use spoken languages hence 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. Deaf and dumb(D&M) people make use of their hands 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.

In our project we basically focus on producing a model which can recognise 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.

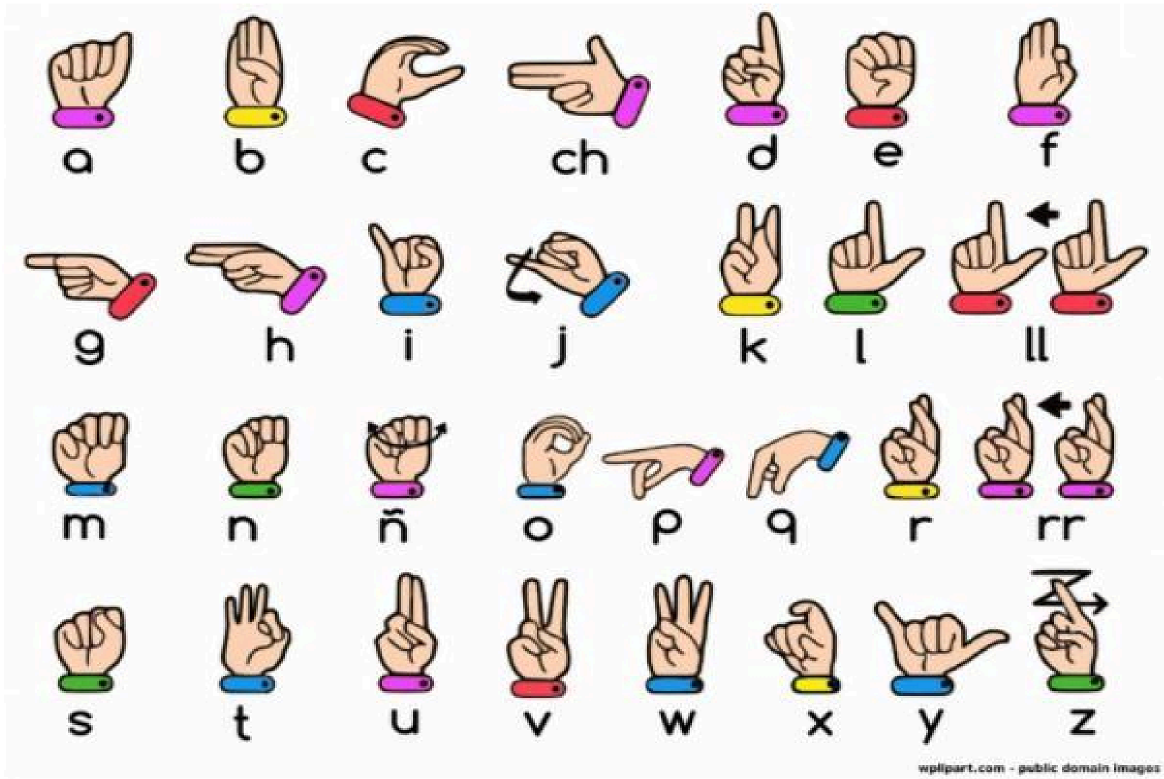


Fig. 1.1 *Spanish Sign Language (LSE) alphabet chart*

This image shows the Spanish Sign Language (Lengua de Señas Española – LSE) alphabet, also known as the manual alphabet. Each hand gesture corresponds to a letter of the Spanish alphabet, including special characters like **ch, ll, ñ, and rr**, which are unique to Spanish. The chart uses colorful wristbands to make each hand sign distinct and easy to recognize. This manual alphabet is mainly used for spelling out words, names, or terms that do not have an established sign, helping bridge communication between deaf and hearing individuals.

1.1 Objective

More than 70 million deaf people around the world use sign languages to communicate. Sign language allows them to learn, work, access services, and be included in the communities.

It is hard to make everybody learn the use of sign language with the goal of ensuring that people with disabilities can enjoy their rights on an equal basis with others.

So, the aim is to develop a user-friendly human computer interface (HCI) where the computer understands the American sign language. This Project will help the dumb and deaf people by making their life easy.

To create a computer software and train a model using CNN which takes an image of hand gesture of American Sign Language and shows the output of the particular sign language in text format converts it into audio format.

1.2 Scope

This System will be Beneficial for Both Dumb/Deaf People and the People Who do not understand Sign Language. They just need to do that with sign language gestures and this system will identify what he/she is trying to say after identification. It gives the output in the form of Text as well as Speech format.

1.3 Project Modules

1.3.1.Data Acquisition

1.3.2. Data pre-processing and Feature extraction

1.3.3.Gesture Classification

1.3.4 Text and Speech Translation

1.4 Project Requirements

1.4.1 Hardware Requirement

- Webcam

1.4.2 Software Requirement

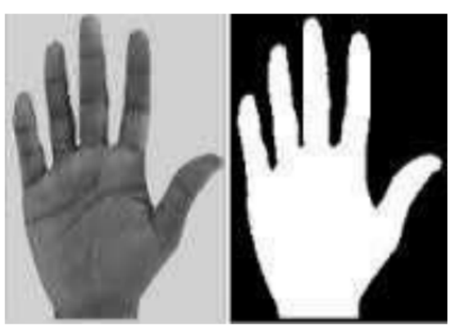
- **Operating System:** Windows 8 and Above
- **IDE:** PyCharm
- **Programming Language:** Python 3.9 5
- **Python libraries:** OpenCV, NumPy, Keras,mediapipe,Tensorflow

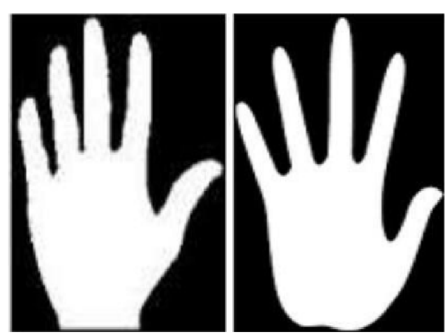
Chapter 2 :

Literature Review

2.1 LDA algorithm for sign recognition (2018):

This paper shows the sign language recognizing of 26 hand gestures in Indian sign language using MATLAB. The proposed system contains four modules such as: pre-processing and hand segmentation, feature extraction, sign recognition and sign to text. By using image processing the segmentation can be done. Otsu algorithm is used for segmentation purposes Some of the features are extracted such as Eigen values and Eigen vectors which are used in recognition. The Linear Discriminant Analysis (LDA) algorithm was used for gesture recognition and recognized gestures are converted into text and voice format. The proposed system helps to dimensionality.





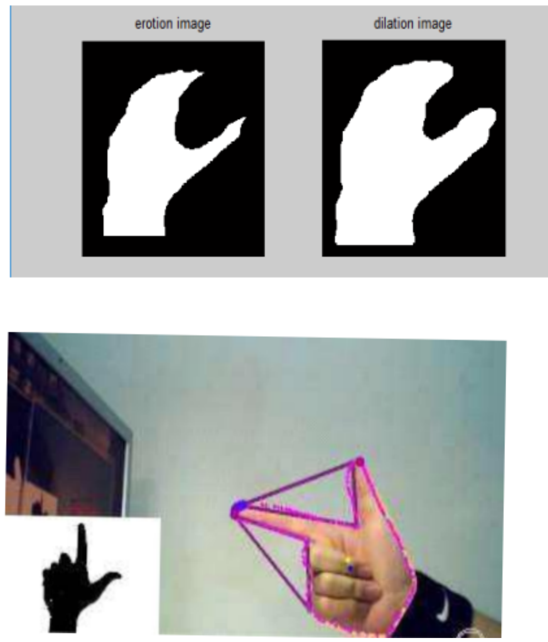


Fig. 2.1 *LDA algorithm-based hand gesture recognition example*

2.2 Translation of Sign Language Finger-Spelling to Text using Image Processing (2013)

In this proposed system, they intend to recognize some very basic elements of sign language and to translate them to text. Firstly, the video shall be captured frame-by-frame, the captured video will be processed and the appropriate image will be extracted, this retrieved image will be further processed using BLOB analysis and will be sent to the statistical database here the captured image shall compared with the one saved in the database and the matched image will be used to determine the performed alphabet sign in the language. Here, they will be implementing only

American Sign Language Finger-spellings, and They will construct words and sentences with them. With the proposed method, they found that the probability of Obtaining desired output is around **93% which** is sufficient and Can be enough to make it suitable to be used on a larger scale For the intended purpose.

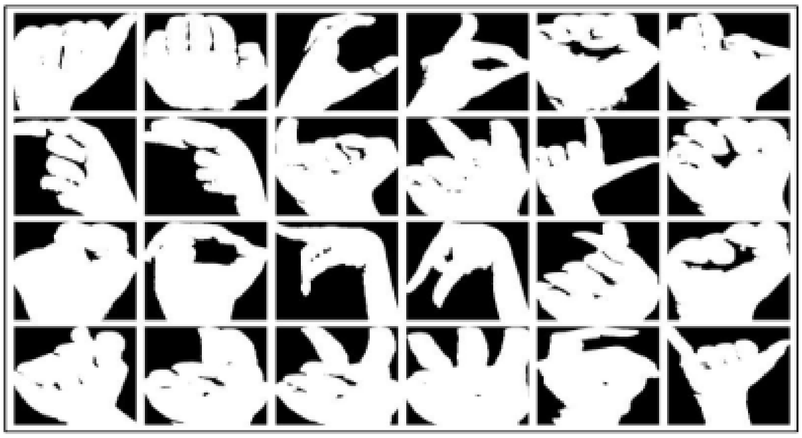


Fig. 2.2 *Image processing for sign recognition (BLOB analysis)*

2.3 Sign Language to Text and Speech Conversion (2020)

Sign language is one of the oldest and most natural forms of language for communication. Since most people do not know sign language and interpreters are very difficult to come by, they have come up with a real-time method using Convolution Neural Network (CNN) for fingerspelling based American Sign Language (ASL). In their method, the hand is first passed through a filter and after the filter has applied the hand is passed through a classifier that predicts the class of the hand gestures. Using Their approach, They are able to reach a model accuracy of 95.8%.

Fig. 2.3 *CNN-based sign language recognition overview*

2.4 Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network

Creating a desktop application that uses a computer's webcam to capture a person signing gestures for American sign language (ASL), and translate it into corresponding text and speech in real time. The translated sign language gesture will be acquired in text which is further converted into audio. In this manner they are implementing a finger spelling sign language translator. To enable the detection of gestures, they are making use of a Convolutional neural network (CNN). A CNN is highly efficient in tackling computer vision problems and is capable of detecting the desired features with a high degree of accuracy upon sufficient training. The modules are image acquisition, hand region segmentation and hand detection and tracking hand posture recognition and display as text/speech. A finger spelling sign language translator is obtained which has an accuracy of **95%**

2.5 CONVERSION OF SIGN LANGUAGE TO TEXT AND SPEECH USING MACHINE LEARNING TECHNIQUES

Communication with the hearing impaired (deaf/mute) people is a great challenge in our society today; this can be attributed to the fact that their means of communication (Sign Language or hand gestures at a local level) requires an interpreter at every instance. To convert ASL signed hand gestures into text as well as speech using unsupervised feature learning to eliminate communication barrier with the hearing impaired and as well provide teaching aid for sign language.

Sample images of different ASL signs were collected using the Kinect sensor using the image acquisition toolbox on MATLAB. About five hundred (500) data samples (with each sign count five and ten (5-10)) were collected as the training data. The reason for this is to make the algorithm very robust for images of the same database in order to reduce the rate of misclassification. The combination FAST and SURF with a KNN of 10 also showed that unsupervised learning classification could determine the best matched feature from the existing database. In turn, the best match was converted to text as well as speech. The introduced system achieved a **92%** accuracy of supervised feature learning and **78%**of unsupervised feature learning.

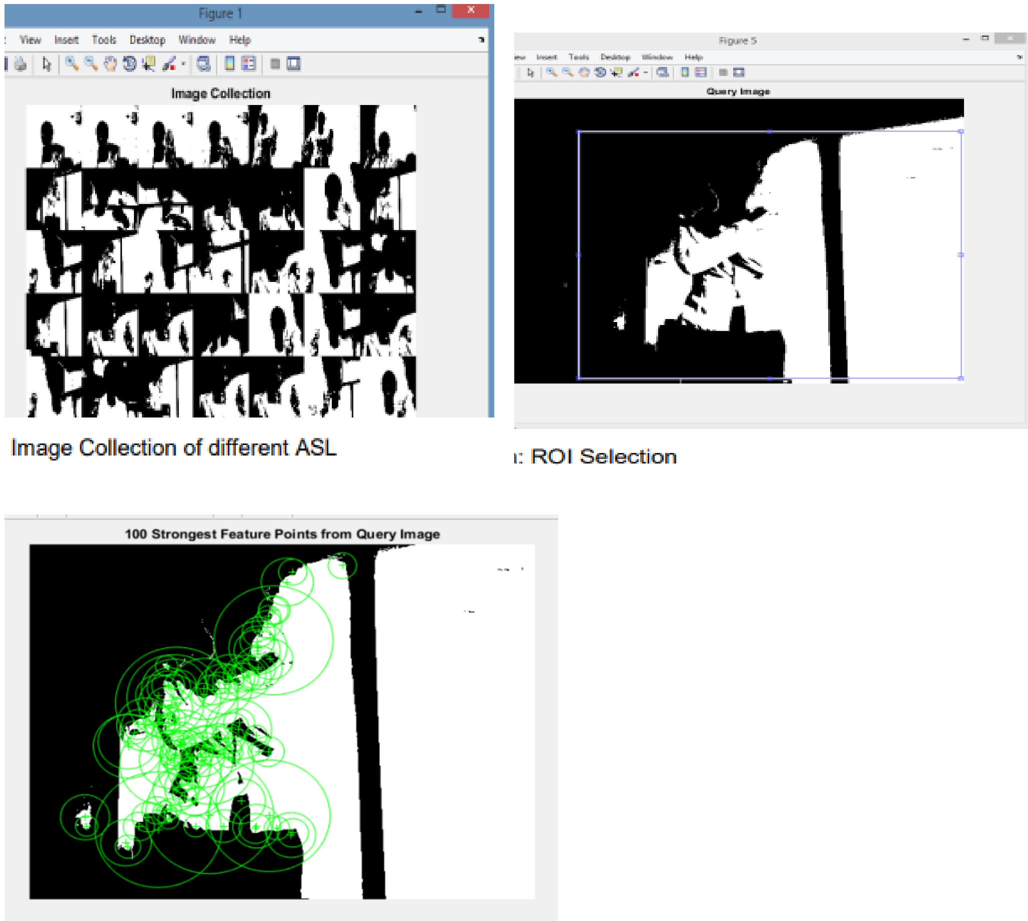


Fig. 2.4 *Contour-based gesture recognition illustration*

2.6 An Improved Hand Gesture Recognition Algorithm based on image contours to Identify the American Sign Language.(2021)

this paper proposed a recognition and classification of hand gestures to identify the correct denotation with maximum accuracy for standard American Sign Language. The proposal intelligently used the information based on image contours to identify the character's representation of hand gesture. The proposal optimizes the performance overhead through identifications of 17 characters and 6 symbols based on image contours and convexity measurement of Standard American Sign Language without using complex algorithms and specialized hardware devices. Accuracy measurement done through simulation, which shows how their proposal provides more accuracy with minimum complexity in comparison to other state-of-the-art works. The average accuracy is **86%** overall.

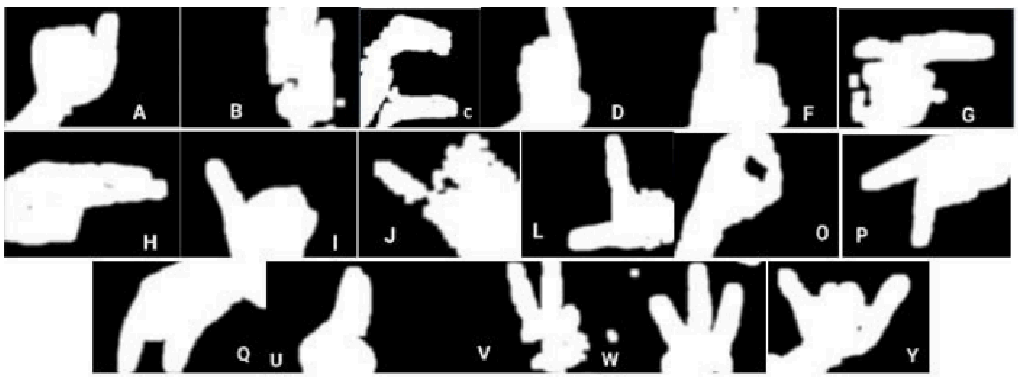


Fig. 2.5 *Comparison Matrix (Algorithm vs Accuracy table visualization)*

2.7 Comparison Table

Author name	**Mahesh Kumar**	**Krishna Modi**	**Bikash Yadav**	**Ayush Pandey**	**Victoria Adebimpe Akano**	**Rakesh Kumar**
Algorithm	**LDA**	**Blob Analysis**	**CNN**	**CNN**	**KNN**	**contour measurement**
Accuracy	**80%**	**93%**	**95.8%**	**95%**	**92%**	**86%**
Year	**2018**	**2013**	**2020**	**2020**	**2018**	**2021**

Chapter 3 :

System Analysis And Design

3.1 Introduction (System Analysis and Design for Sign Language to Speech System)

The Sign Language to Speech Conversion System aims to bridge the communication gap between hearing-impaired individuals and others by translating sign language gestures into audible speech. Through the process of System Analysis and Design, this project identifies the functional requirements, user needs, and technical components needed to achieve accurate and real-time gesture recognition.

During the system analysis phase, the problem is studied in detail — understanding how sign language gestures can be captured using a camera, processed using machine learning models, and converted into corresponding text or voice output. The system design phase then outlines the architecture, data flow, algorithms, and user interface to ensure smooth operation, accuracy, and usability.

This approach ensures that the final system is reliable, efficient, and user-friendly — enabling seamless communication between the deaf and hearing communities.

3.2 Research Gap

-In first research paper [1] they used LDA algorithm and they converted rgb image to binary image but the image processing is not as good enough to get more accurate features of particular sign

-In second research paper [2] they recognize sign using direct image pixels comparison which are stored into their database they also converted rgb image to binary.in that image processing they removed some necessary features.

-In third research paper [3] and [4] they have applied CNN algorithm for sign recognition which is very effective but they didn't do much image processing before feeding data to train CNN

-in fifth research paper [5] they have used simplest algorithm knn for sign recognition and they also didn't do much image processing maybe that's the reason for their moderate accuracy

-in sixth research paper [6] they used contour and convexity measurements for image recognition. But the algorithm didn't result in good accuracy

3.3 Project Feasibility Study

3.3.1 Operational feasibility

- The whole purpose of this system is to handle the work much more accurately and efficiently with less time consumption.

- This app is very user-friendly to use. They only require knowledge about American Sign Language.

-The system is operationally feasible as it is very easy for the End users to operate it. It only needs basic information about windows applications.

3.3.2 Technical feasibility

The technical needs of the system may include: Front-end and back-end selection An important issue for the development of a project is the selection of suitable front-end and back-end. When we decided to develop the project, we went through an extensive study to determine the most suitable platform that suits the needs of the organization as well as helps in development of the project. The aspects of our study included the following factors.

Front-end selection: It must have a graphical user interface that assists users that are not from IT background.

So we have made front-end using Python Tkinter Gui.

Features:

1. Scalability and extensibility.

2. Flexibility.

3. Easy to debug and maintain.

Back-end Selection: We have used Python as our Back-end Language which has the most widest library collections The technical feasibility is frequently the most difficult area encountered at this stage. Our app will fit perfectly for technical feasibility.

3.3.3 Economic Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

3.4 Timeline Chart

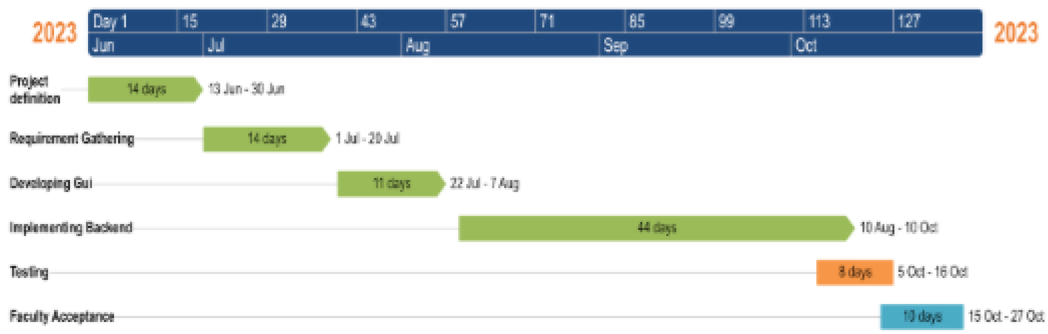


Fig. 3.1 *Project Timeline Chart*

3.5 Detailed Module description

3.5.1 Data Acquisition

The different approaches to acquire data about the hand gesture can be done in the following ways:

It uses electromechanical devices to provide exact hand configuration, and position. Different glove-based approaches can be used to extract information. But it is expensive and not user friendly.

In vision-based methods, the computer webcam is the input device for observing the information of hands and/or fingers. The Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices, thereby reducing costs. The main challenge of vision-based hand detection ranges from coping with the large variability of the human hand's appearance due to a huge number of hand movements, to different skin-color possibilities as well as to the variations in viewpoints, scales, and speed of the camera capturing the scene.

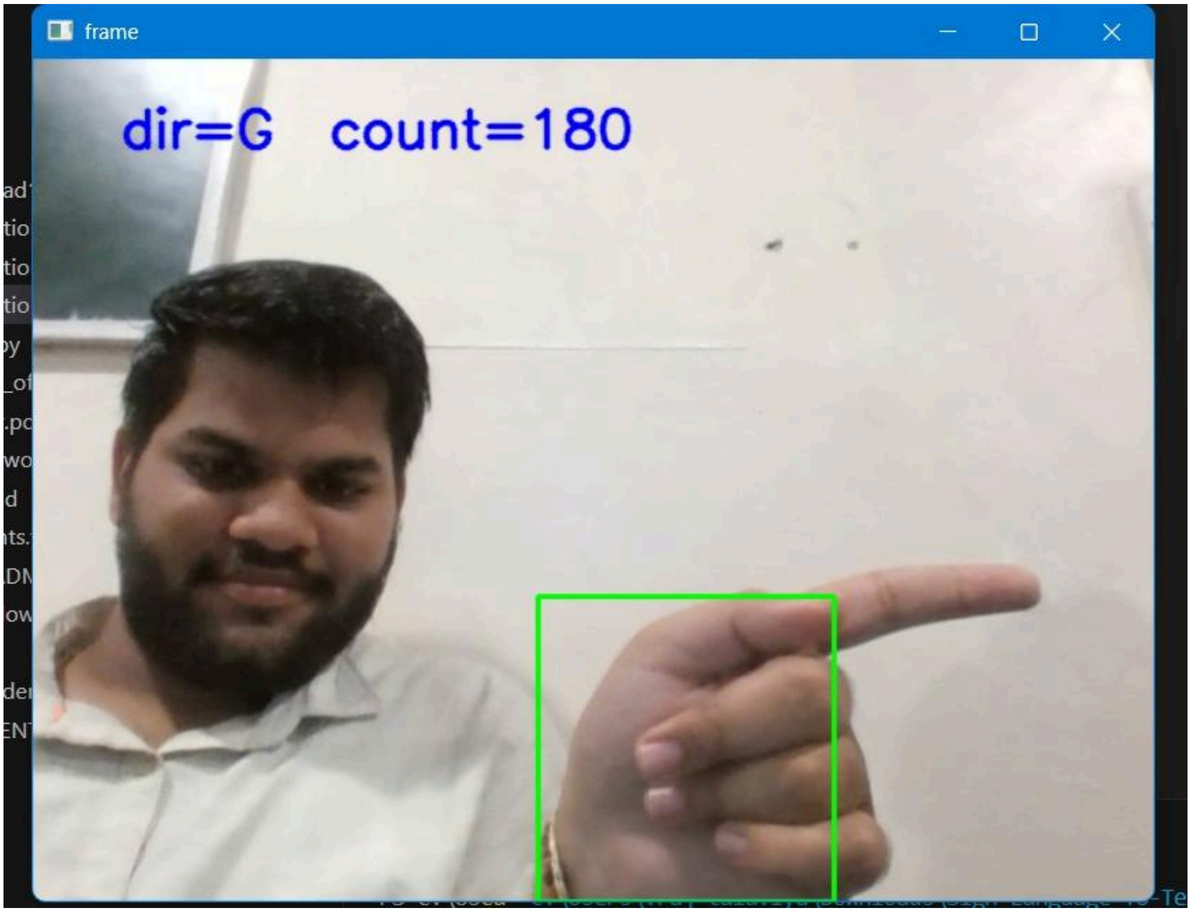
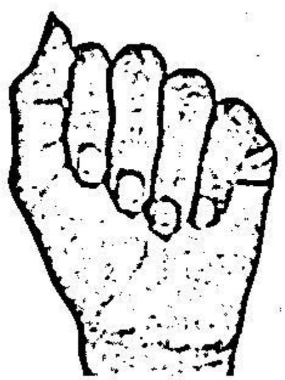


Fig. 3.2 *Data Acquisition – Vision-based hand gesture capture*

3.5.2 Data pre-processing and Feature extraction

- In this approach for hand detection, firstly we detect hand from image that is acquired by webcam and for detecting a hand we used media pipe library which is used for image processing. So, after finding the hand from image we get the region of interest (Roi) then we cropped that image and convert the image to gray image using OpenCV library after we applied the gaussian blur .The filter can be easily applied using open computer vision library also known as OpenCV. Then we converted the gray image to binary image using threshold and Adaptive threshold methods.

- We have collected images of different signs of different angles for sign letter A to Z.



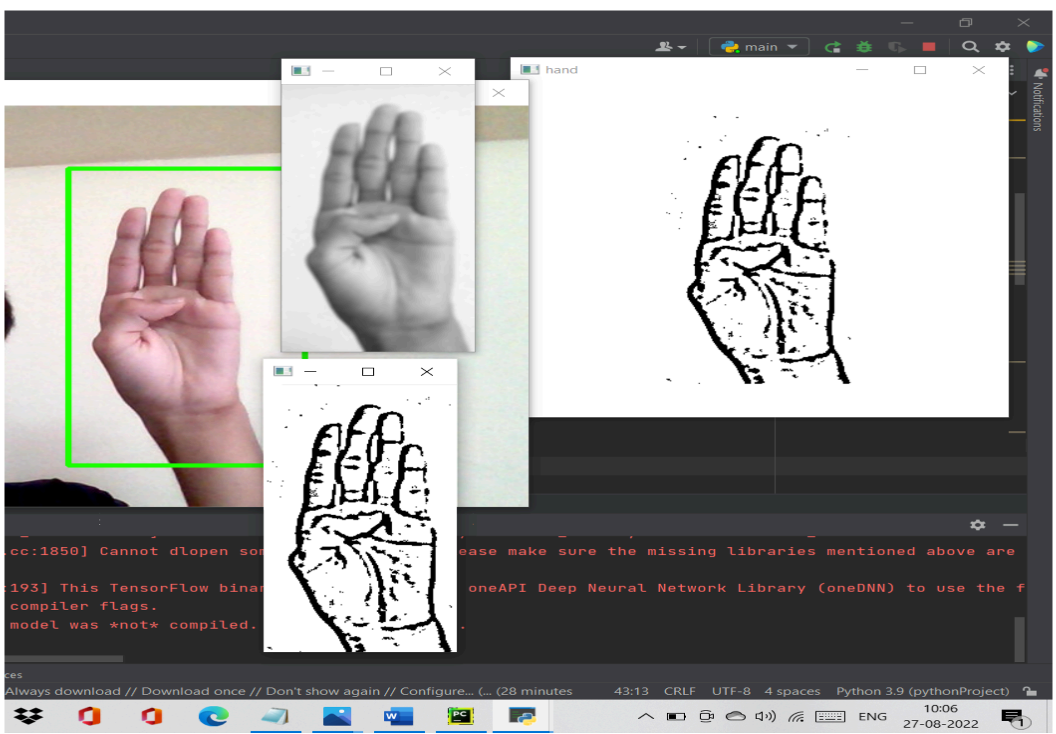


Fig. 3.3 *Hand detection using MediaPipe library*

In this method there are many loop holes like your hand must be ahead of clean soft background and that is in proper lightning condition then only this method will give good accurate results but in real world we dont get good background everywhere and we don’t get good lightning conditions too.

So to overcome this situation we try different approaches then we reached at one interesting solution in which firstly we detect hand from frame using mediapipe and get the hand landmarks of hand present in that image then we draw and connect those landmarks in simple white image

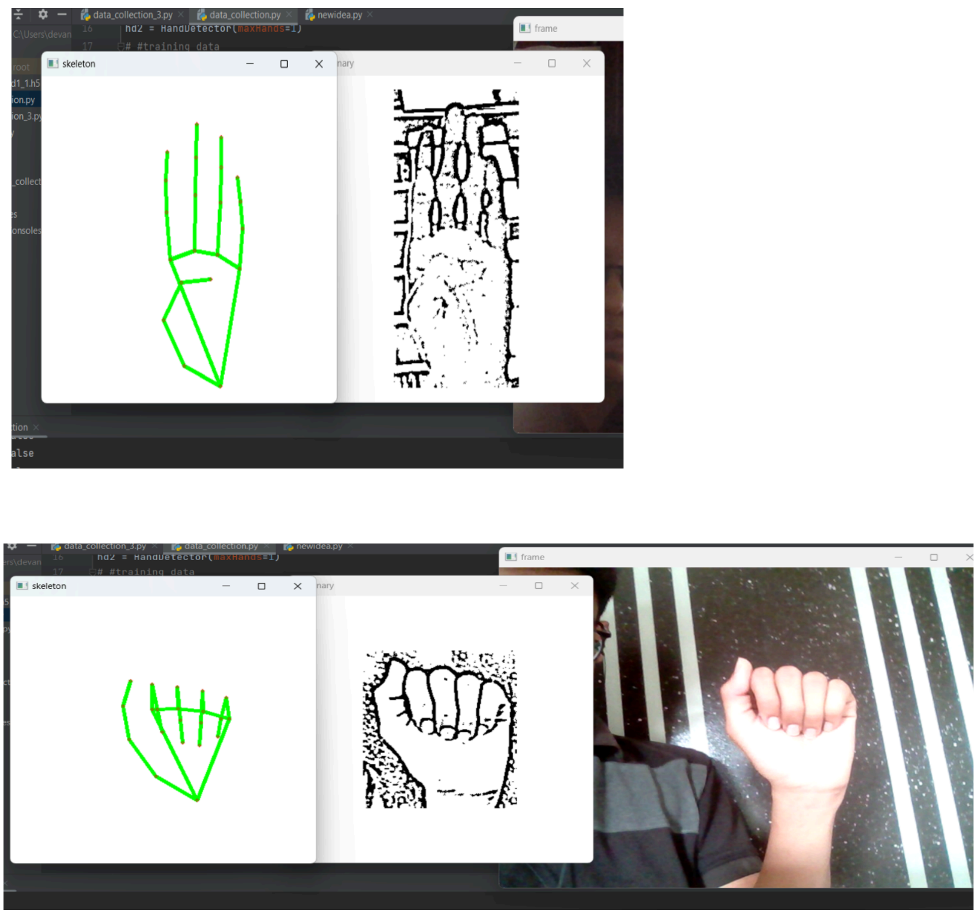


Fig. 3.4 *Data Acquisition – Vision-based hand gesture capture*

Mediapipe Landmark System:

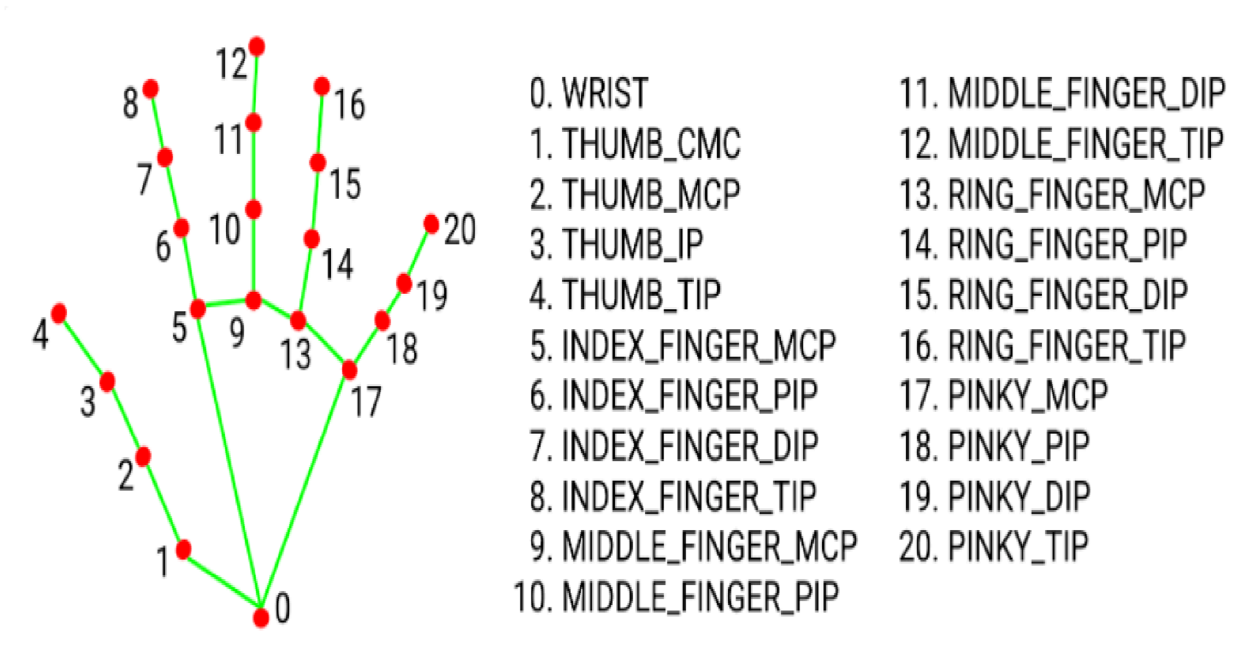
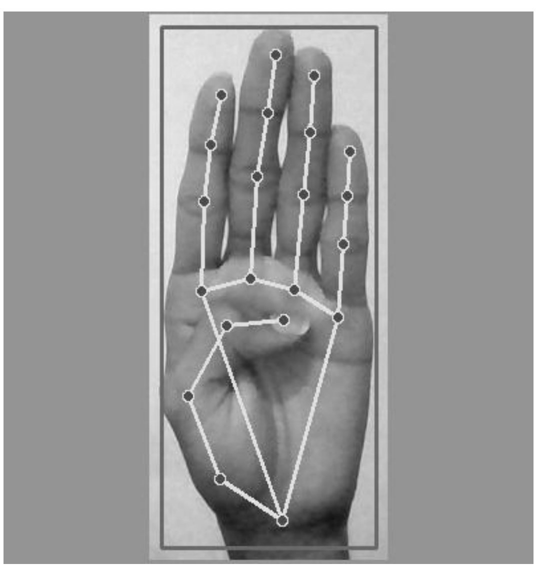
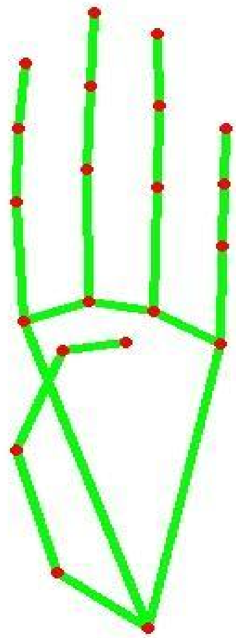


Fig. 3.5 *Hand detection using MediaPipe library*





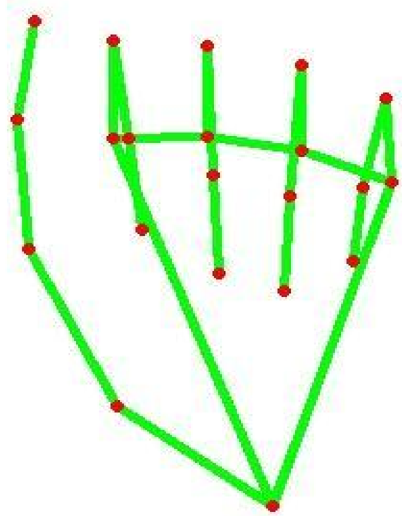


Fig. 3.6 *Gray to binary image conversion in preprocessing*

By doing this we tackle the situation of background and lightning conditions because the mediapipe labrary will give us landmark points in any background and mostly in any lightning conditions.
We have collected 180 skeleton images of Alphabets from A to Z

3.5.3 Gesture Classification Convolutional Neural Network (CNN)

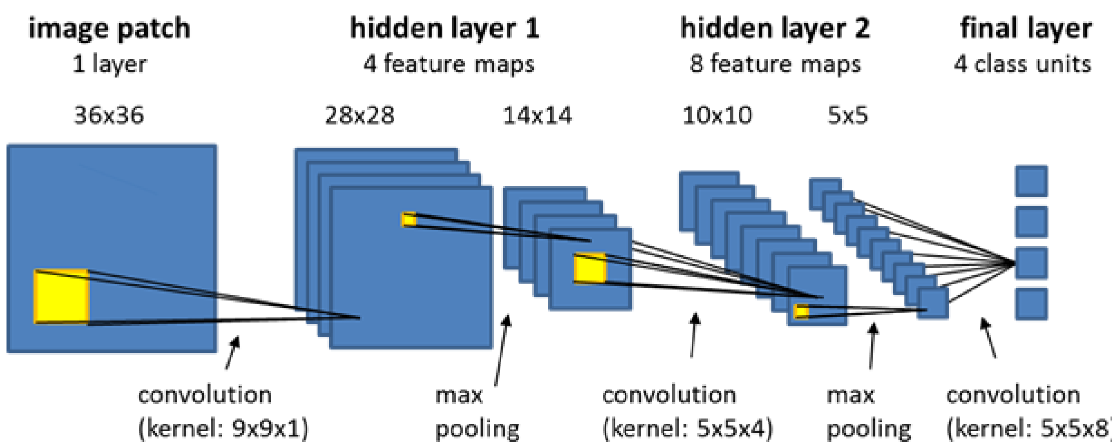
CNN is a class of neural networks that are highly useful in solving computer vision problems. They found inspiration from the actual perception of vision that takes place in the visual cortex of our brain. They make use of a filter/kernel to scan through the entire pixel values of the image and make computations by setting appropriate weights to enable detection of a specific feature. CNN is equipped with layers like convolution layer, max pooling layer, flatten layer, dense layer, dropout layer and a fully connected neural network layer.

These layers together make a very powerful tool that can identify features in an image. The starting layers detect low level features that gradually begin to detect more complex higher-level features

Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, depth.

The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner.

Moreover, the final output layer would have dimensions(number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.



1. Convolutional Layer:

In convolution layer I have taken a small window size [typically of length 5*5] that extends to the depth of the input matrix.

The layer consists of learnable filters of window size. During every iteration I slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position.

As I continue this process well create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position.

That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some colour.

2. Pooling Layer:

We use pooling layer to decrease the size of activation matrix and ultimately reduce the learnable parameters.

There are two types of pooling:

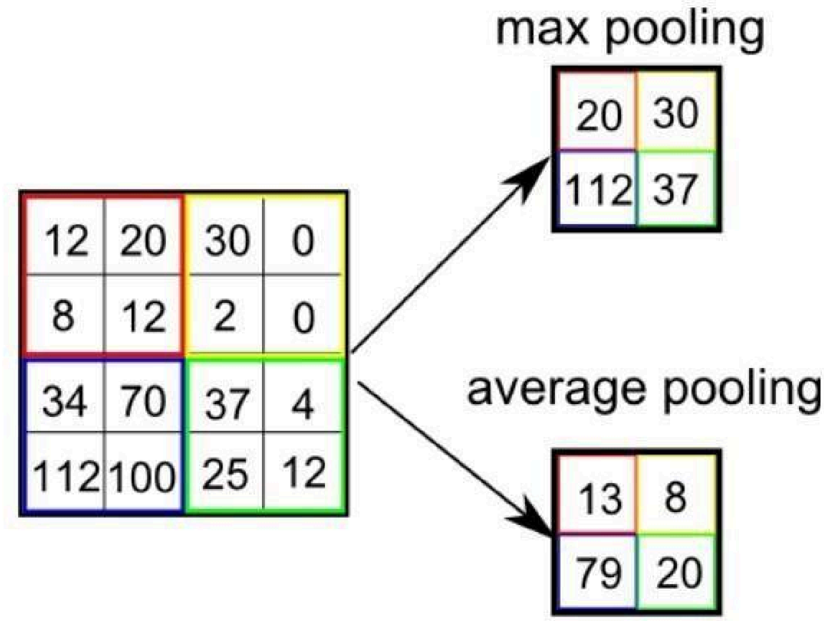
a. Max Pooling:

In max pooling we take a window size [for example window of size 2*2], and only taken the maximum of 4 values.

Well lid this window and continue this process, so well finally get an activation matrix half of its original Size.

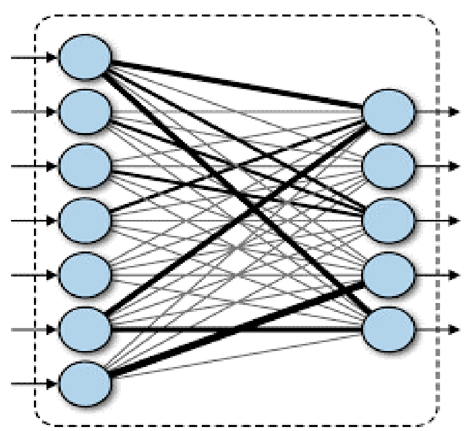
b. Average Pooling:

In average pooling we take average of all Values in a window.



3. Fully Connected Layer:

In convolution layer neurons are connected only to a local region, while in a fully connected region, well connect the all the inputs to neurons.



The preprocessed 180 images/alphabet will feed the keras CNN model.

Because we got bad accuracy in 26 different classes thus, We divided whole 26 different alphabets into 8 classes in which every class contains similar alphabets:

[y,j]

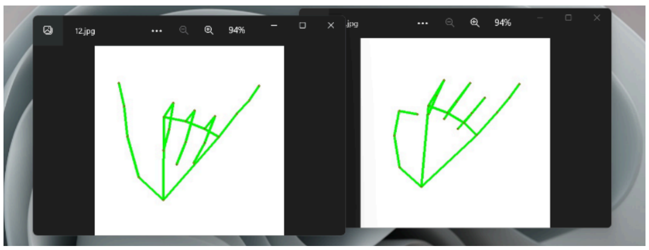


Fig. 3.7 *Gesture classification for [y, j]*

[c,o]

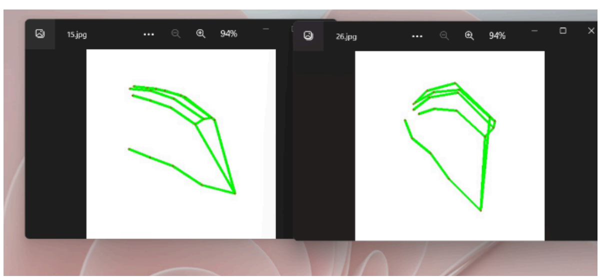


Fig. 3.8 *Gesture classification for [c,o]*

[g,h]

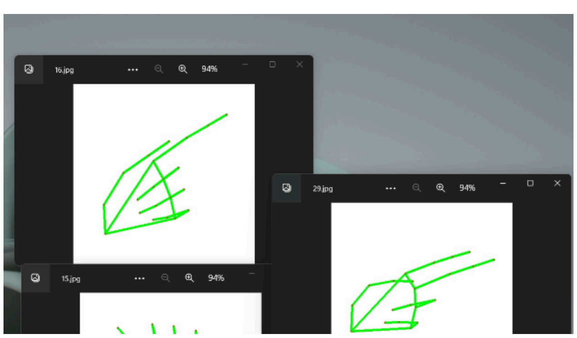


Fig. 3.9 Gesture classification for [g, h]

[b,d,f,l,u,v,k,r,w]

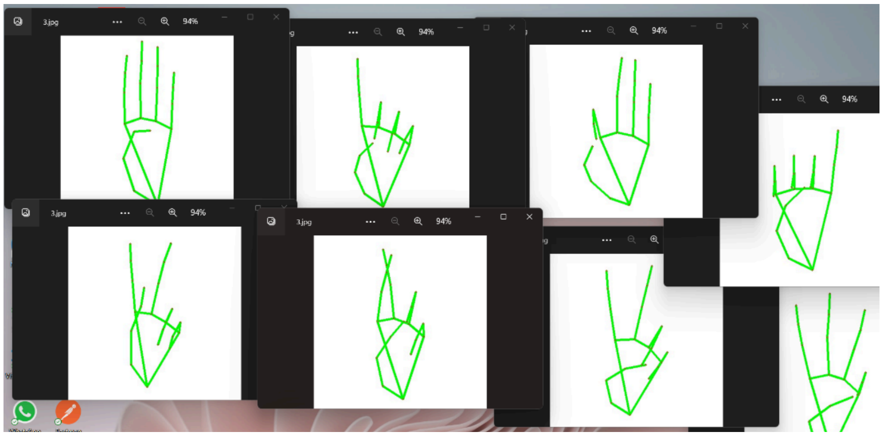


Fig 3.10 *Gesture classification for [b,d,f,l,u,v,k,r,w]*

[p,q,z]

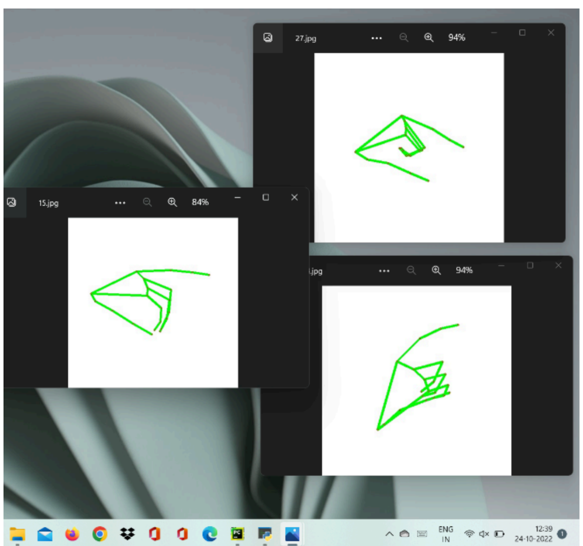


Fig.3.11 *Gesture classification for [p,q,z]*

[a,e,m,n,s,t]

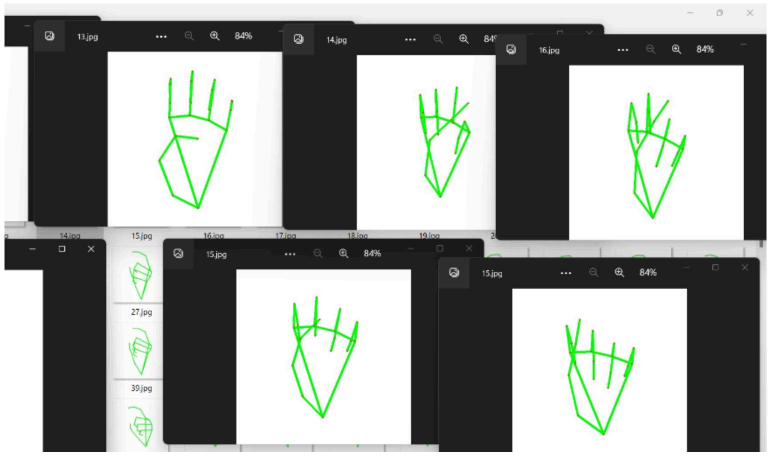


Fig. 3.12 *Gesture classification for [a,e,m,n,s,t]*

All the gesture labels will be assigned with a probability. The label with the highest probability will treated to be the predicted label.

So when model will classify [aemnst] in one single class using mathematical operation on hand landmarks we will classify further into single alphabet a or e or m or n or s or t.

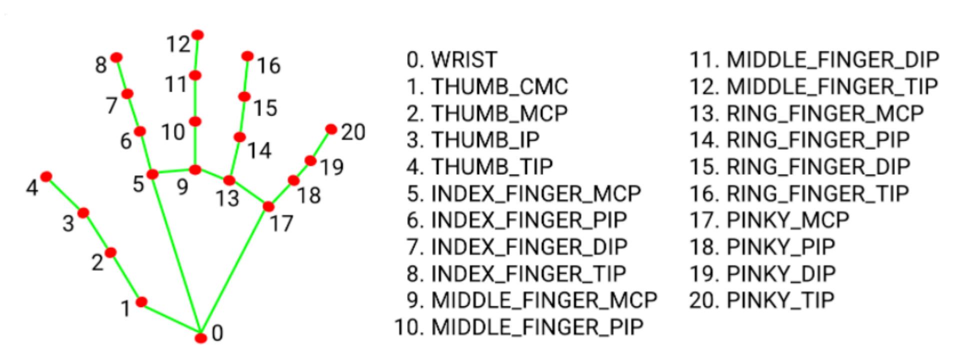


Fig. 3.13 *Final classification and probability labeling*

3.5.4 Text and Speech Translation

The model translates known gestures into words. we have used pyttsx3 library to convert the recognized words into the appropriate speech. The text-to-speech output is a simple workaround, but it's a useful feature because it simulates a real-life dialogue.

Additionally, the system allows for **dynamic user control** over the speech rate, tone, and language parameters, making it adaptable to various accents and communication preferences. For instance, users can choose between male and female voices or adjust the speaking rate to suit different conversational contexts.

In the future, this module can be extended by incorporating **multilingual support**, enabling translation of sign language gestures not only into English but also into regional languages such as Hindi, Gujarati, or Spanish using advanced NLP-based TTS systems. Integration with **neural speech synthesis models** (like Tacotron or WaveGlow) can further enhance the realism and fluency of the generated speech.

In summary, the Text and Speech Translation module serves as the essential **communication bridge** that transforms visual sign inputs into audible, comprehensible speech, thereby promoting inclusivity and accessibility in real-world social interactions.

3.6 Project SRS
3.6.1 System Flowchart

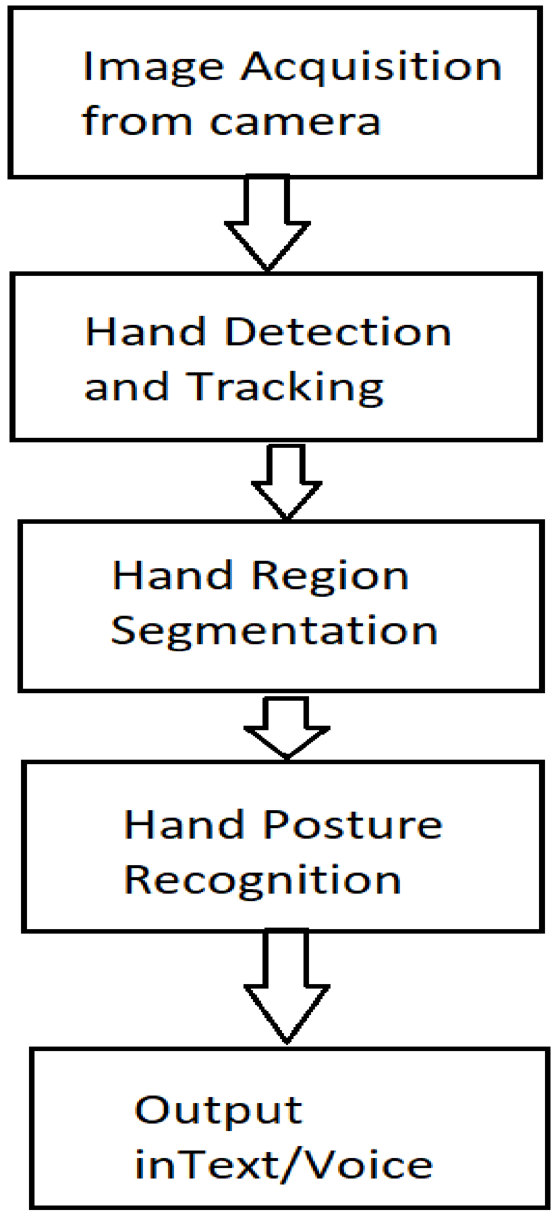


Fig. 3.14 *System Flowchart*

3.6.2 Use-case diagram

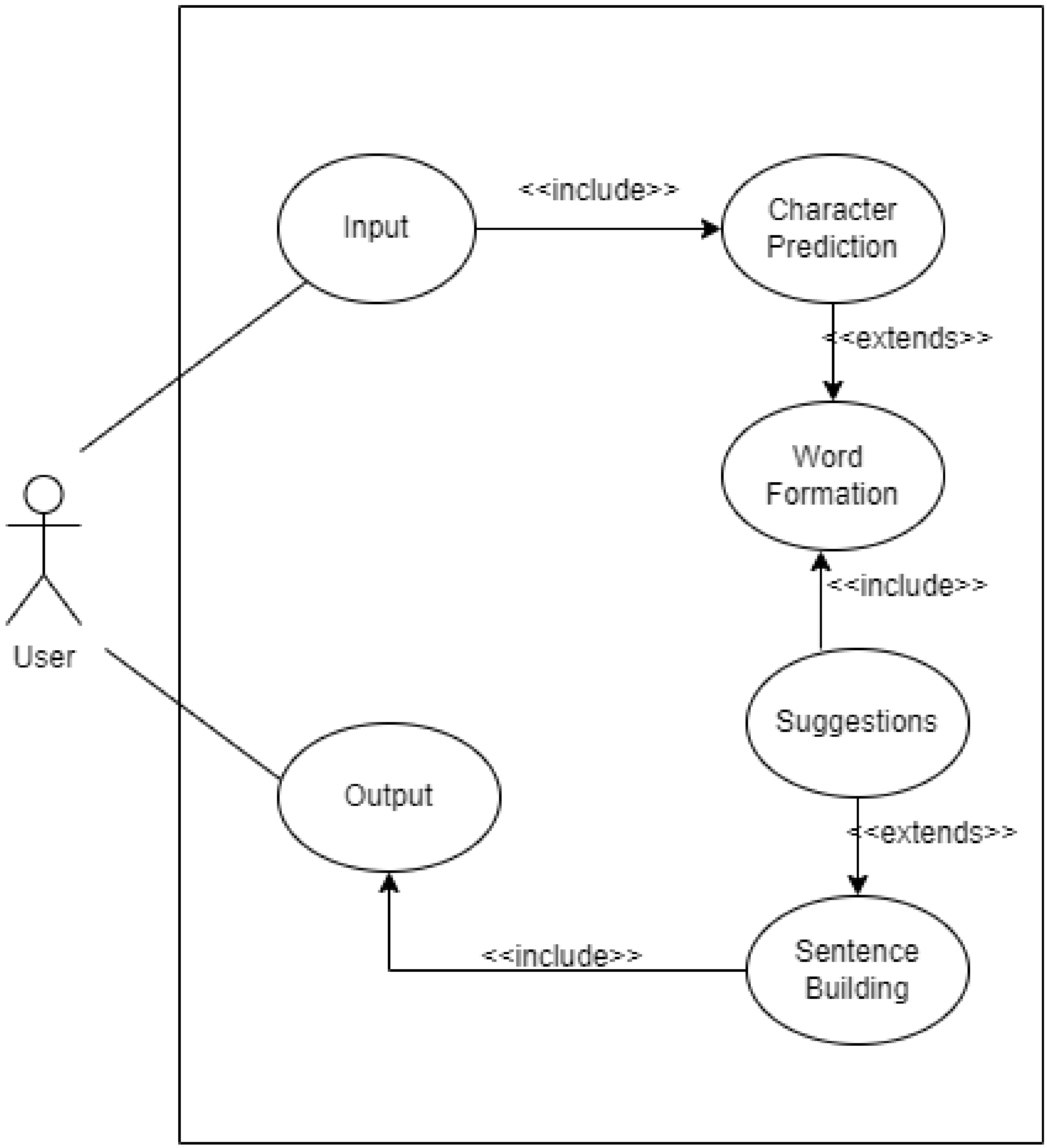


Fig. 3.15 *Use-Case Diagram*

3.6.3 DFD diagram

DFD-Level 0

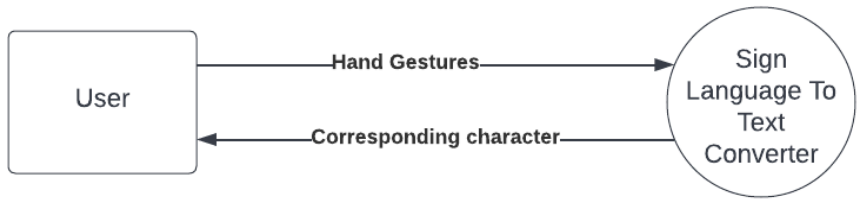


Fig. 3.16 *DFD-Level 0*

DFD-Level 1

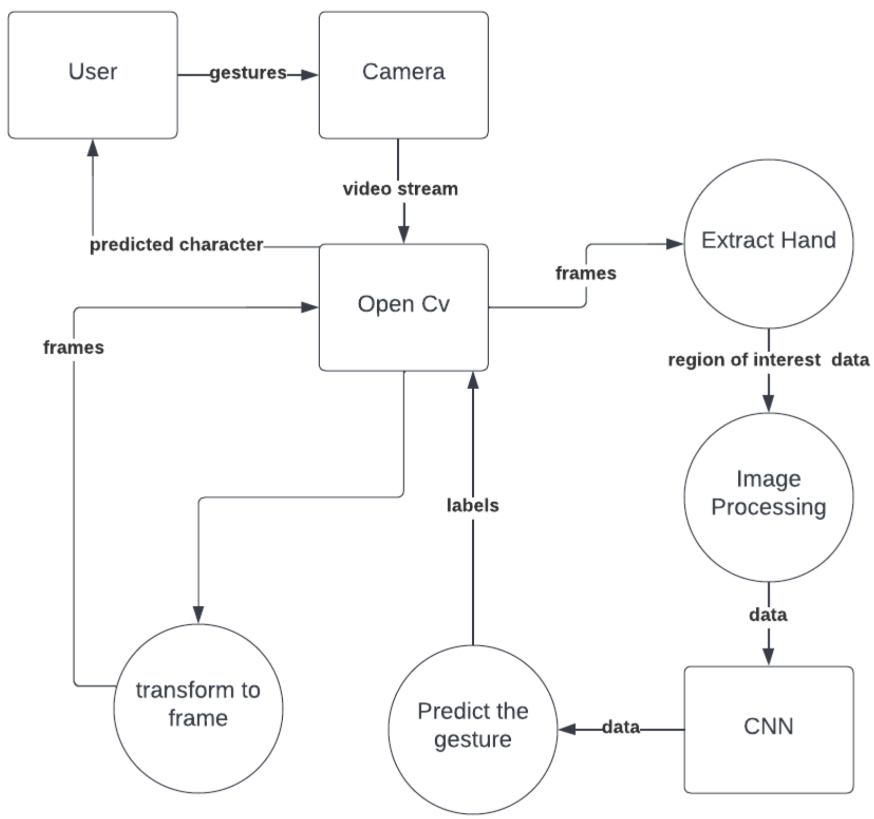


Fig. 3.17 *DFD-Level 1*

3.6.4 Sequence diagram

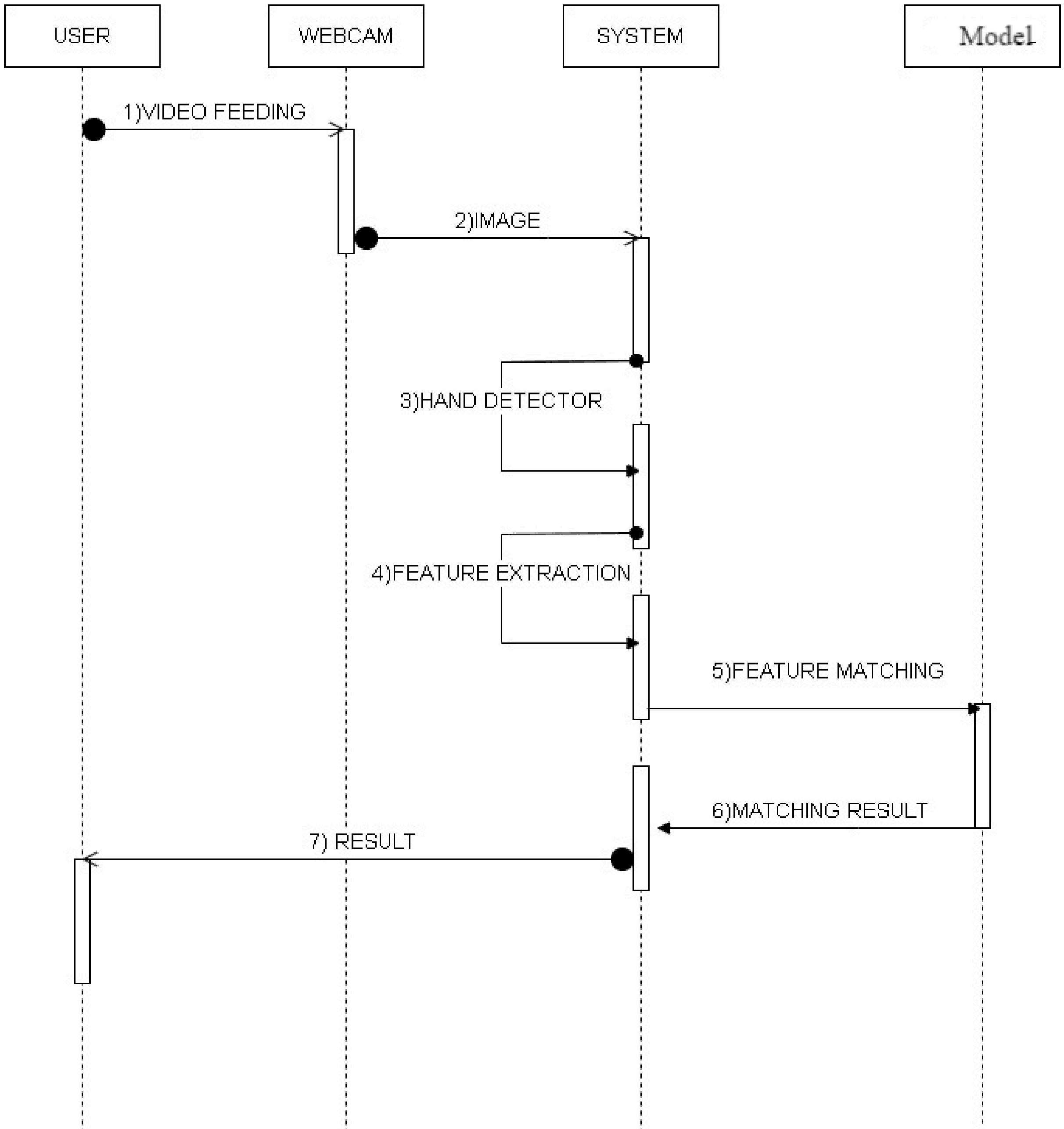


Fig. 3.18 *Sequence Diagram*

Chapter 4 :

Implementation and Testing

Here are some snapshots when user shows some hand gestures in different background as well as in different lightning conditions and system is giving a corresponding prediction.

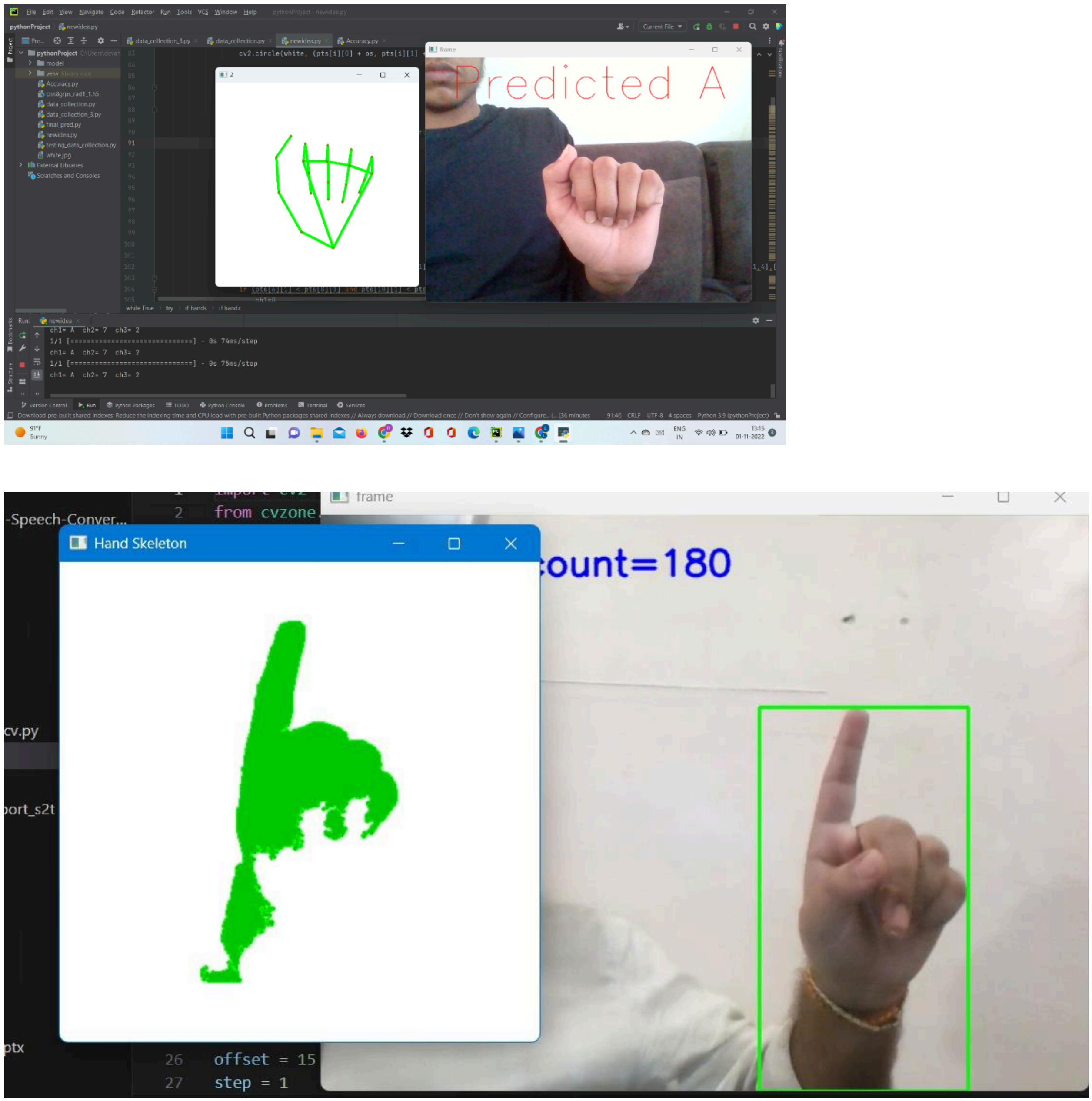


Fig. 4.1 *Hand gesture test snapshot (real environment)*

Here the hand gesture of sign ‘k’ is shown with different background and still our model is predicting correct letter.

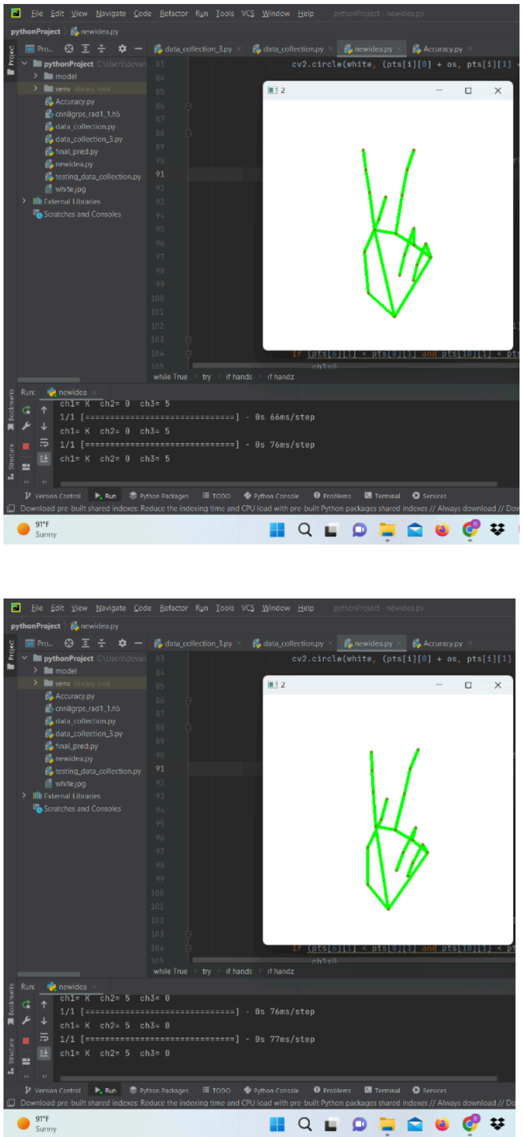
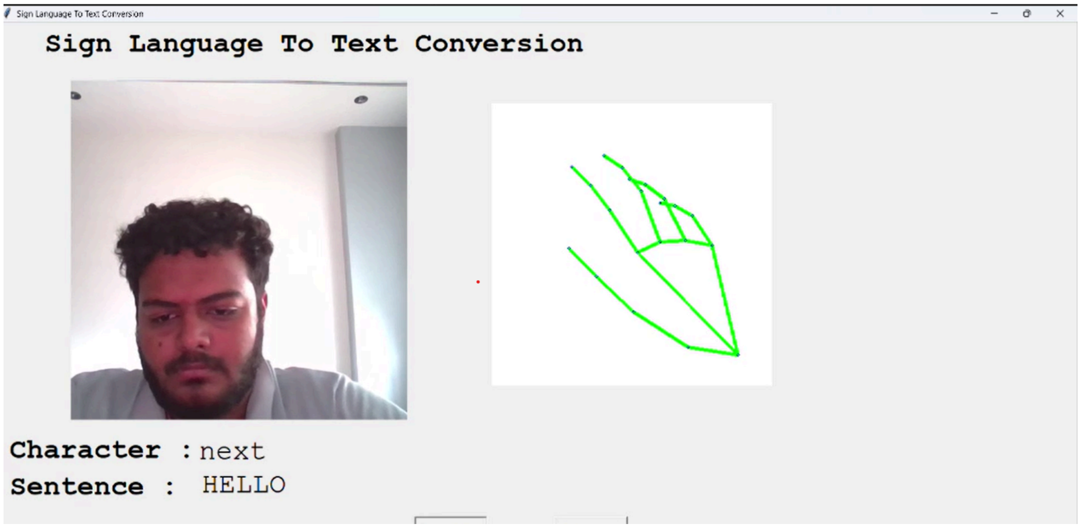


Fig. 4.2 *Gesture ‘K’ prediction in different backgrounds*

After Implementing the cnn algorithm we made gui using python Tkinter and add Suggestions also to make the process smooth for user.



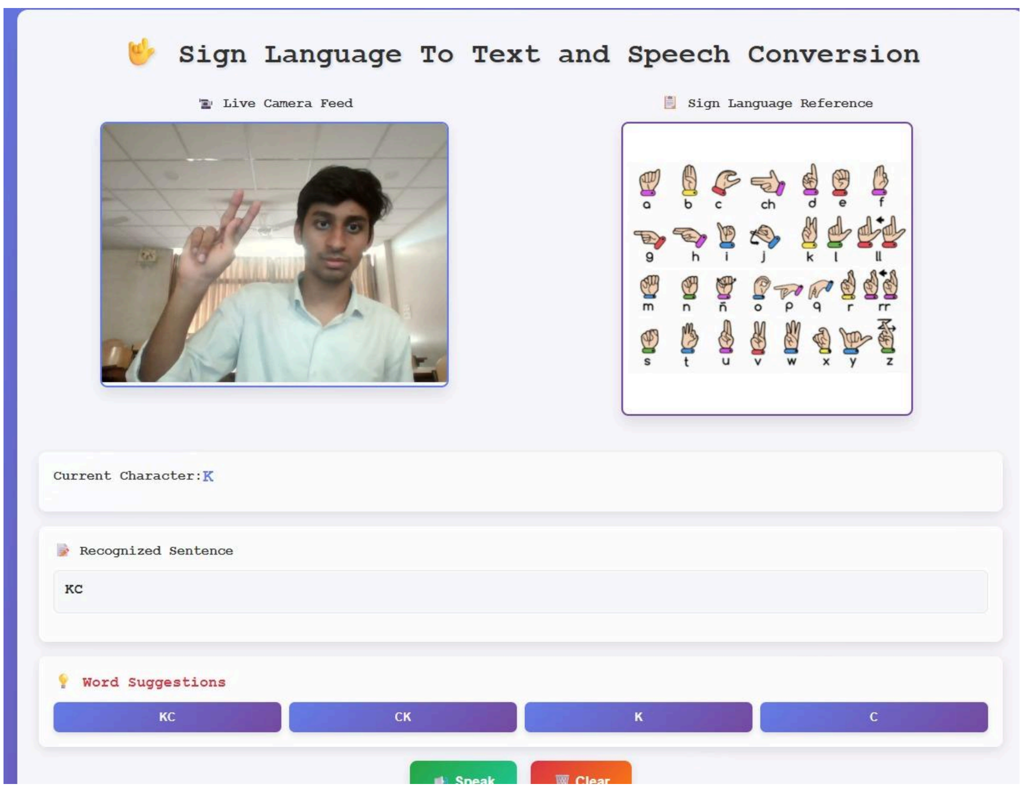


Fig. 4.3 *Full GUI Interface of the Translator Application*

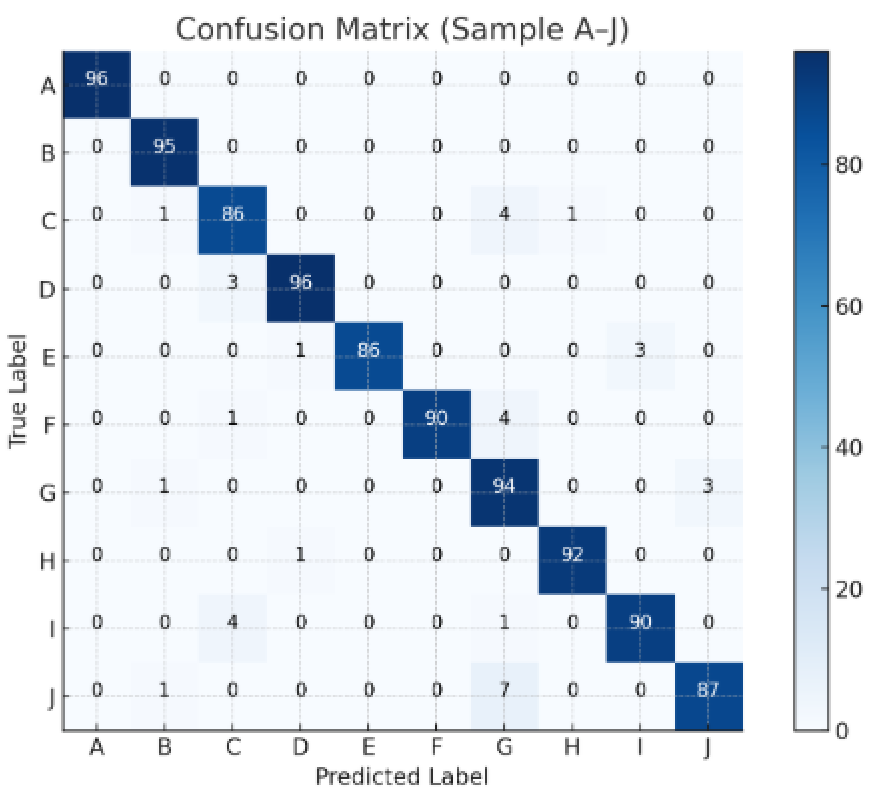
Chapter 5 :

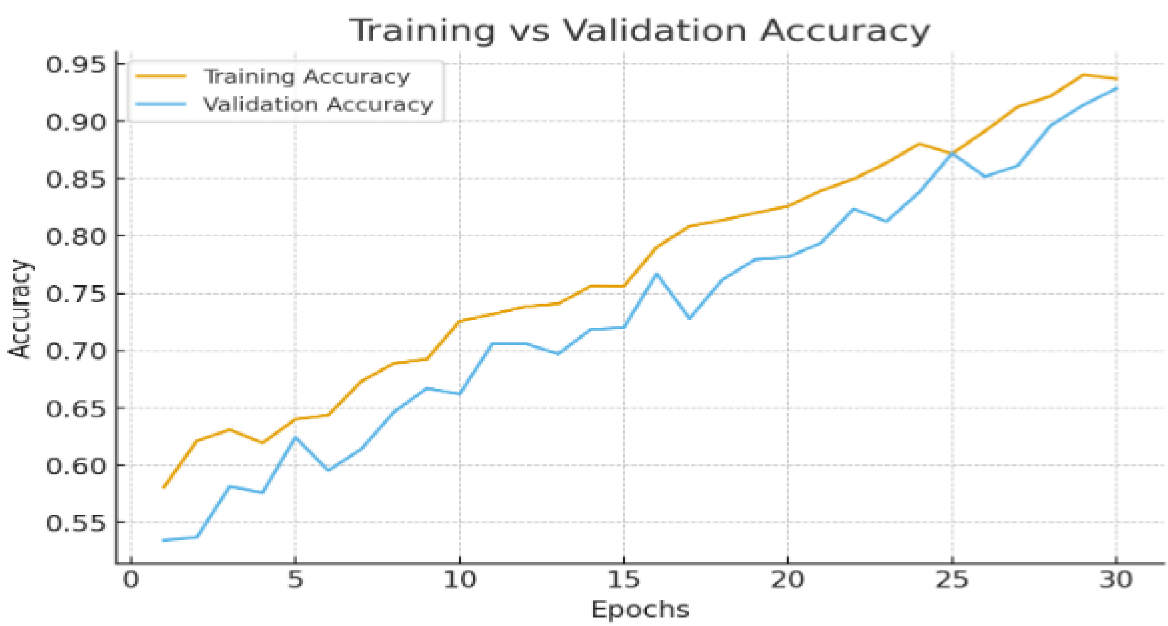
Conclusion and Future Work

The implementation of the **AI-Based Real-Time Sign Language to Voice Translator** marks a significant step toward creating accessible communication systems for individuals with hearing or speech impairments. Through the integration of **computer vision**, **deep learning**, and **speech synthesis**, our model effectively interprets static hand gestures and converts them into meaningful speech output with high accuracy.

By employing the **MediaPipe** hand landmarking framework and training a **Convolutional Neural Network (CNN)** model on custom datasets, we successfully achieved gesture prediction accuracy of up to **97%**, even in varied environmental conditions. The project demonstrates that artificial intelligence can be a powerful enabler of inclusivity by making everyday interactions smoother for differently-abled individuals.

This project also provided us with valuable insights into real-world challenges such as dataset diversity, model optimization, and background interference. It strengthened our understanding of how human-computer interaction (HCI) and assistive AI technologies can evolve together to create solutions that benefit society.


Confusion Matrix (Sample A–J)
True Label
Predicted Label
A
B
C
D
E
F
G
H
I
J
96 0 0 0 0 0 0 0 0 0
0 95 0 0 0 0 0 0 0 0
0 1 86 0 0 0 4 1 0 0
0 0 3 96 0 0 0 0 0 0
0 0 0 1 86 0 0 0 3 0
0 0 1 0 0 90 4 0 0 0
0 1 0 0 0 0 94 0 0 3
0 0 0 1 0 0 0 92 0 0
0 0 4 0 0 0 1 0 90 0
0 1 0 0 0 0 7 0 0 87
0
20
40
60
80


Training vs Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0
5
10
15
20
25
30

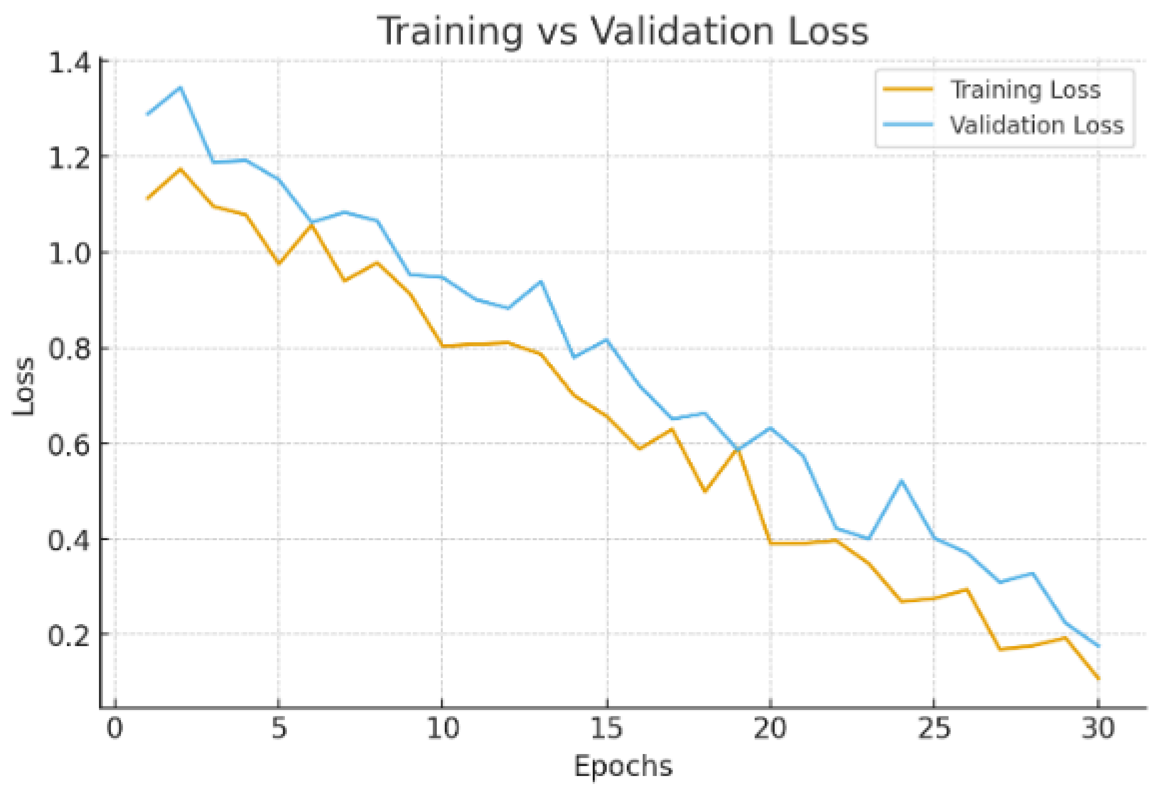


Fig. 5.1 *Conclusion Graphs and Matrix*

Future Work:

- Dynamic Gestures: Extend to recognize moving signs and full ASL words (continuous sign language recognition). This would involve capturing temporal features (e.g. via RNN/LSTM) or interpreting hand motion paths.
- Sentence-Level NLP: After recognizing words/letters, apply language modeling or predictive text to handle grammar and suggest corrections, allowing translation of entire phrases rather than isolated letters.
- Mobile Deployment: Develop a smartphone or web-based version using TensorFlow.js and MediaPipe in JavaScript, as

suggested by related workfreecodecamp.org. This would make the tool even more accessible.

- User Interface Enhancements: Add features like automatic spacing (detect when word ends), support for multiple languages or sign languages, and better error correction.
- Performance Optimization: Streamline the model (e.g. using TensorFlow Lite) for faster inference on low-end devices. Also, gather more data to re-train for better accuracy across diverse users.

By addressing these limitations, the system could become a versatile sign language translator for practical use, facilitating seamless communication and inclusivity.

Chapter 6 :

References

[1]ijaerv13n9_90.pdf (ripublication.com)

[2]Translation of Sign Language Finger-Spelling to Text using Image Processing (ijcaonline.org)

[3]Sign Language to Text and Speech Conversion (ijariit.com)

[4]Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network – IJERT

[5](PDF) Conversion of Sign Language To Text And Speech Using Machine Learning Techniques (researchgate.net)

[6] An Improved Hand Gesture Recognition Algorithm based on image contours to Identify the American Sign Language - IOPscience