

**TRIBHUVAN UNIVERSITY**

**Faculty of Humanities and Social Science**

**SIGN LANGUAGE RECOGNITION SYSTEM A PROJECT REPORT**

Submitted to

**Department of Computer Application Ed-mark College**

#### In partial fulfillment of the requirements for the Bachelors of Computer Application

Submitted by Anita Ghimire

Reg. No: 6-2-727-3-2019

Under the Supervision of

**Mr. Bhas Raj Pathak**

October, 2024



**TRIBHUVAN UNIVERSITY**

**Faculty of Humanities and Social Science Ed-mark College**

**SUPERVISOR’S RECOMMENDATION**

I hereby recommend that this project prepared under my supervision by **Anita Ghimire** entitled **SIGN LANGUAGE RECOGNITION SYSTEM** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

………………………

**SIGNATURE**

**Bhas Raj Pathak SUPERVISOR**

**Ed-mark College Kalanki, Kathmandu**



**TRIBHUVAN UNIVERSITY**

**Faculty of Humanities and Social Science Ed-mark College**

**LETTER OF APPROVAL**

This is to certify that this project prepared by **Anita Ghimire** entitled ―**SIGN LANGUAGE RECOGNITION SYSTEM**‖ in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| ……………………………. **Bhas Raj Pathak Supervisor**  **Kalanki, Kathmandu** | …………………………. **Sanjay Kumar Sah Coordinator**  **Kalanki, Kathmandu** |
| …………………………….  **Internal Examiner** | ……………………………  **External Examiner** |

# ABSTRACT

The communication between a people from the impaired community with a person who does not understand sign language could be tedious task. This is why sign language is introduced. Sign language is an art of conveying message using hand gesture. But for this people have to learn sign languages which take time and money. Developed software that will be a bridge between these people with the help of only a mobile phone. System classifies the hand gestures using AI and translates it into English alphabets. This is achieved by collecting the required image data, extracting features, feeding to the machine learning algorithms, and generating the classifier output.

The user will be prompted to display the hand gesture in front of the camera. Then the video from camera will be fed to our program. Through mediapipe it will obtain the hand landmarks which will be pre-processed to get only the necessary data. The processed data will then be fed to the pre-trained model which will classify the corresponding letter. The resulting alphabet will be shown on the screen.

#### Keywords: Hand Gesture, ML, AI

# ACKNOWLEDGEMENT

The success of this project would not have been possible without the kind support and assistance of many individuals and organizations, and I am immensely blessed to have got this all along with the duration of this project. We would like to extend my profound gratitude to each and every one of them.

I am highly indebted to ED-MARK COLLEGE for constant guidance and supervision, as well as for providing the entire necessary ICT infrastructure and a friendly environment for the successful completion of the project. We are also appreciative of the efforts of BCA coordinator **Mr. Sanjay Kumar Sah** without his supporting role, the project would have been nowhere near completion.

I would like to express my gratitude to project supervisor **Mr**. **Bhas Raj Pathak** who took a keen interest in my project and guided me throughout the project by providing all the necessary ideas, information, and knowledge for the development of a functional Mobile application.

I am thankful and fortunate enough to get constant support from every teaching staff of the BCA department which helped to successfully complete this project.

……………... Anita Ghimire BCA 8th Sem.

# TABLE OF CONTENT

[ABSTRACT iii](#_bookmark0)

[ACKNOWLEDGEMENT iv](#_bookmark1)

[TABLE OF CONTENT v](#_bookmark2)

[LIST OF FIGURES vii](#_bookmark3)

[LIST OF TABLES viii](#_bookmark4)

[LIST OF ABBREVIATIONS ix](#_bookmark5)

[CHAPTER 1](#_bookmark6) [INTRODUCTION 1](#_bookmark7)

* 1. [Introduction 1](#_bookmark8)
  2. [Problem Statement 1](#_bookmark9)
  3. [Objective 2](#_bookmark10)
  4. [Scope and Limitation 2](#_bookmark11)
  5. [Development Methodology 2](#_bookmark12)
  6. [Report Organization 3](#_bookmark14)

[CHAPTER 2 BACKGROUND STUDY AND LITERATURE REVIEW 5](#_bookmark15)

* 1. [Background Study 5](#_bookmark16)
  2. [Literature Review 5](#_bookmark17)

[CHAPTER 3](#_bookmark18) [SYSTEM ANALYSIS AND DESIGN 7](#_bookmark19)

* 1. [System Analysis 7](#_bookmark20)
     1. [Requirement Analysis 7](#_bookmark21)
     2. [Feasibility Analysis 8](#_bookmark23)
     3. [Object Modeling: Object & Class Diagram 9](#_bookmark25)
     4. [Dynamic Modeling: State & Sequence diagram: 10](#_bookmark27)
     5. [Process modeling: Activity Diagram 11](#_bookmark29)
  2. [System Design: 11](#_bookmark31)
     1. [Refinement of class and object 11](#_bookmark32)
     2. [Component diagrams 12](#_bookmark34)
     3. [Deployment diagram 13](#_bookmark36)
  3. [Algorithm details 13](#_bookmark38)

[CHAPTER 4](#_bookmark39) [IMPLEMENTATION AND TESTING 16](#_bookmark40)

* 1. [Implementation 16](#_bookmark41)
     1. [Tools used 16](#_bookmark42)
     2. [Implementation details of modules 16](#_bookmark43)
  2. [Testing 17](#_bookmark44)
     1. [Test cases for Unit testing 17](#_bookmark45)

[CHAPTER 5](#_bookmark51) [CONCLUSION AND FUTURE RECOMMENDATIONS 20](#_bookmark52)

* 1. [Conclusion 20](#_bookmark53)
  2. [Future Recommendations 20](#_bookmark54)

[REFERENCES 21](#_bookmark55)

[APPENDICES 22](#_bookmark56)

# LIST OF FIGURES

[Figure 1. 1 Agile Methodology 3](#_bookmark13)

[Figure 3. 1 Use Case Diagram of Sign Language Recognition](#_bookmark22) 7

[Figure 3. 2 Class Diagram of Sign Language Recognition 9](#_bookmark26)

[Figure 3. 3 Sequence diagram of Sign Language Recognition 10](#_bookmark28)

[Figure 3. 4 Activity Diagram of Sign Language Recognition 11](#_bookmark30)

[Figure 3. 5 Refinement Diagram of Sign Language Recognition 12](#_bookmark33)

[Figure 3. 6 Component diagram of Sign Language Recognition 13](#_bookmark35)

[Figure 3. 7 Deployment Diagram of Sign Language Recognition 13](#_bookmark37)

# LIST OF TABLES

[Table 3. 1 Schedule Feasibility 9](#_bookmark24)

[Table 4. 1 Unit Testing for loading model 17](#_bookmark46)

[Table 4. 2 Unit testing for converting video to frame 18](#_bookmark47)

[Table 4. 3 Unit testing for recognizing Hand Gesture 18](#_bookmark48)

[Table 4. 4 Unit Testing for Training and Saving the Model 19](#_bookmark49)

[Table 4. 5 System testing for Classifying the Sign Language 19](#_bookmark50)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| ICT | Information and Communication Technology |
| ML | Machine Learning |
| RTA | Road Traffic Accidents |
| SDLC | Software Development Life Cycle |
| SLR | Sign Language Recognition |

* 1. **Introduction**

# CHAPTER 1 INTRODUCTION

Sign Language is mainly used by deaf (hard hearing) and dumb people to exchange information between their own community and with other people. It is a language where people use their hand gestures to communicate as they can't speak or hear. Sign Language Recognition (SLR) deals with recognizing the hand gestures acquisition and continues till text is generated for corresponding hand gestures. It allows them to convey complex ideas, emotions, and information through a series of hand gestures. However, the barrier between sign language users and those who do not understand it can limit effective communication in various settings such as education, healthcare, and everyday interactions. This is where a Sign Language Detection System comes into play. This system aims to bridge the communication gap between sign language users and non- users, fostering inclusivity and accessibility.

This might be challenging for the general people to understand the gestures used in sign language. This sign language can be translated into a form that is easily understood by the general public. This model will be more efficient and hence communicate for the deaf (hard hearing) and dump people will be easier.

## Problem Statement

Over 100 million people - more than 1% of the world’s population - are unable to hear. Being deaf from birth or childhood, many of these people use sign language as their primary form of communication[1].

There are several hundred sign languages around the world and these also have their own language. One of the most common of these is American Sign Language (ASL). More than 500,000 people use ASL in the US alone, and millions more use it worldwide. Most hearing people don’t know that written English is only the second language of people who are born deaf[1].

According to Nepal Census, 2078 2.2% of the Nepali Population have some form of disability. (Deaf 7.85%), Hard of Hearing (7.87%), Deaf and blind (1.56%), Speech

impairment (6.36%).

Dumb people use hand signs to communicate, hence normal people face problem in recognizing their language by signs made. Hence there is a need of the systems which recognizes the different signs and conveys the information to the normal people.

## Objective

* + - To recognize the hand gestures using machine learning model and translate the hand gesture movement into English Alphabet Letters and words.

## Scope and Limitation

**Scope**

The major scope of Sign Language Recognition (SLR) systems is to provide an efficient and accurate way to convert sign language into text for the hearing impaired or enabling very young children to interact with computers (recognizing sign language), among others. In this project, a vision-based system is able to interpret static hand gestures from the American Sign Language alphabet. We can use this model to understand sign language and it acts as a bridge between the people who use sign language and those who can't understand sign language.

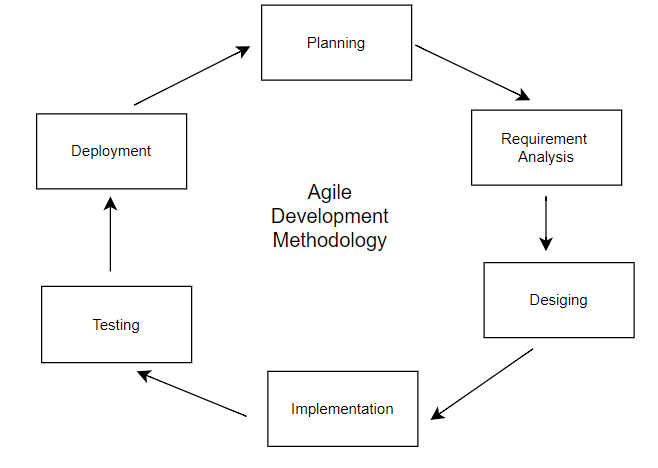
**Limitation**

Since the system is based on Isolated sign language technique, sequence of gestures that generate a meaningful sentence is not considered here and the system only recognizes the single alphabet and some defined simple Symbols like hello, bye etc. detected by the system and it only focuses on American alphabets.

## Development Methodology

Agile is a methodology focused on iterative development, flexibility, and continuous improvement. For the sign language detection system, Agile is employed to manage the project's development in a dynamic and adaptive manner. The process begins with defining project objectives and creating a prioritized backlog of features and tasks. The project is then divided into time-boxed sprints, during which specific features are developed, tested, and refined. Daily stand-ups ensure regular communication and prompt issue resolution, while sprint reviews involve stakeholders to gather feedback and validate

progress. This feedback drives adjustments and refinements to the system. User stories guide the development, ensuring that features align with actual user needs.



**Figure 1. 1 Agile Methodology**

## Report Organization

Report of this project consists of multiple sections and chapters illustrating the in-depth briefing of the project they are:

**Chapter 1:** Introduction covers all the general outline of the project including the generic briefing of the developed project, how it is developed and what was the outcome.

Problem statement covers the negative points of the current situation and explains why this matters.

Objective covers what is the main purpose of the project. Scope and Limitations explains what are the scope of the project and what are the limitations of the project. This part explains what methodology the project followed to complete the project.

**Chapter 2:** Background Study involves a critical evaluation of the context for a study. This may range from situational analysis to the historical evaluation of the gap where a study fits. Literature review refers to a study of relevant ideas that you may use to compare and contrast the concepts under investigation.

**Chapter 3:** System analysis covers the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their

purpose. Whereas, System Design focuses on how to accomplish the objective of the system.

**Chapter 4:** This section involves the tools used in the project, the testing process in the system and the bugs that we faced while making the system. This process serves the dual purpose of verifying that the specification is implementable in practice, and that implementations conform to the specification. It also includes the result analysis and classification reports

**Chapter 5:** Conclusion part has summarized the key findings, outcomes or information in our report. It highlights the big achievements. Whereas, the future recommendation part explains how the project can be improvised in the future.

**CHAPTER 2**

**BACKGROUND STUDY AND LITERATURE REVIEW**

## Background Study

The existing Sign Language Detection System relies primarily on manual interpretation and translation by human sign language interpreters. This traditional approach lacks real- time responsiveness and may not always be readily available, leading to communication challenges for the deaf and hard of hearing community. It often involves face-to-face or remote interpretation services, which can be expensive and logistically challenging. The absence of an automated system limits the accessibility and inclusivity of sign language communication in various contexts. [2]

The Sign Language Classification (SLC) system, which is required to recognize sign languages, has been widely studied for years. The studies are based on various input sensors, gesture segmentation, extraction of features and classification methods. People around the world communicate using sign language as distinct from spoken language everyday, a visual language that uses a system of manual, facial and body movements as the means of communication[3].

Sign language is not an universal language, and different sign languages are used in different countries, like the many spoken languages all over the world. Some countries such as Belgium, the UK, the USA or India may have more than one sign language. Hundreds of sign languages are in used around the world, for instance, Japanese Sign Language, British Sign Language (BSL), Spanish Sign Language, Turkish Sign Language.[4]

## Literature Review

Sign language recognition system: It uses webcam-captured video frames, processed with OpenCV and MediaPipe Holistic, and analyzed with an LSTM network to translate gestures into written text. The system aims to enhance communication between deaf and non-deaf individuals in real-time. Challenges included dataset creation and LSTM training, which were addressed by optimizing hyperparameters and implementing regularization techniques. Future work will focus on expanding the dataset and combining CNN with LSTM for improved accuracy. [5]

Vision-based sign language recognition system: Presents a real-time vision-based system for recognizing static hand gestures in Portuguese Sign Language (PSL). Using a Kinect camera for data acquisition, the system extracts hand features and employs Support Vector Machine (SVM) classifiers to achieve high accuracy in gesture recognition. Compares two feature sets centroid distance and hand depth image values—finding that both achieve similar high accuracy, with centroid distance being computationally simpler. The system shows potential for broader applications in human-computer interaction and can be extended to recognize the entire PSL alphabet and dynamic gestures. [6]

Deep learning-based sign language recognition system for static signs: Sign language is crucial for human communication, and research in computer vision for Indian Sign Language (ISL) recognition is progressing. This study focuses on robust modeling of 100 static ISL signs using deep learning-based CNNs, with 35,000 images collected from various users. Evaluating 50 CNN models, the proposed system achieved a highest training accuracy of 99.72% on colored and 99.90% on grayscale images. Performance metrics like precision, recall, and F-score confirm its effectiveness over earlier methods that recognized only a few signs. [7]

ML Based Sign Language Recognition System: This paper reviews steps in developing an automated sign language recognition (SLR) system using vision-based isolated hand gesture detection and recognition. Using a convex hull for feature extraction and KNN for classification, the model was evaluated with 4 candidates in a controlled environment, achieving 65% accuracy. This system aims to bridge the communication gap for the speech and hearing impaired. [8]

Towards an Automatic Sign Language Recognition System Using Subunits: This paper explores automatic recognition of German continuous sign language using a single color video camera. It employs a statistical approach based on Bayes decision rules and focuses on subunits, called fenones, rather than whole signs to minimize training material and expand vocabulary. The fenones are defined using the K-means algorithm. Current experiments are evaluating the software prototype. [9]

**CHAPTER 3**

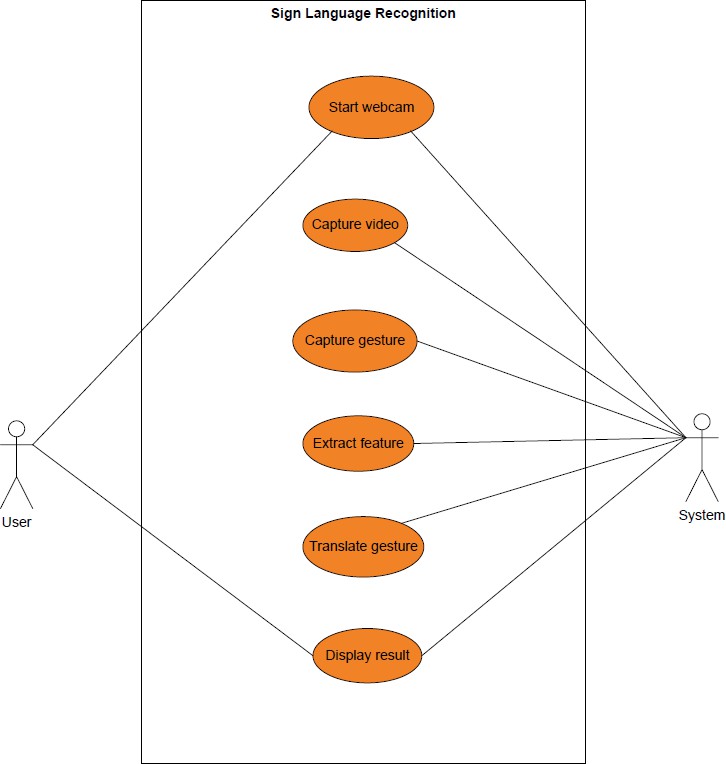
**SYSTEM ANALYSIS AND DESIGN**

## System Analysis

The major objectives of system analysis are to find answers for each business process, what is being done, how is it being done, who is doing it, when it is being done, why is it being done and how can it be improved? It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints.

### Requirement Analysis

* + - 1. **Functional requirements**



**Figure 3. 1 Use Case Diagram of Sign Language Recognition**

* + - 1. **Non-functional requirements**
* **Reliability**: The application always generates an output.
* **Robustness**: The application doesn’t terminate under any circumstances.
* **Availability**: The service provided by the system can be used by all the users.
* **Usability**: The system is secure as no data which could be used to identify individuals or be considered confidential are being collected or stored.

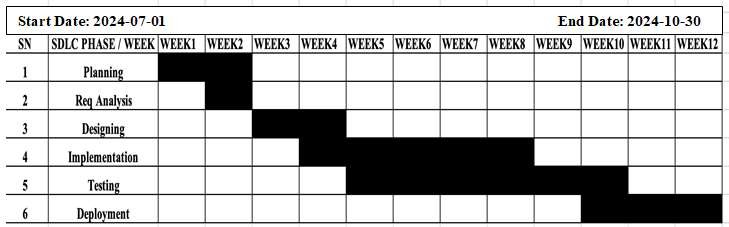
### Feasibility Analysis

A feasibility Analysis is a crucial step in determining whether a sign language recognition system is technically, operationally, and economically feasible. Here is a brief analysis of each feasibility aspect:

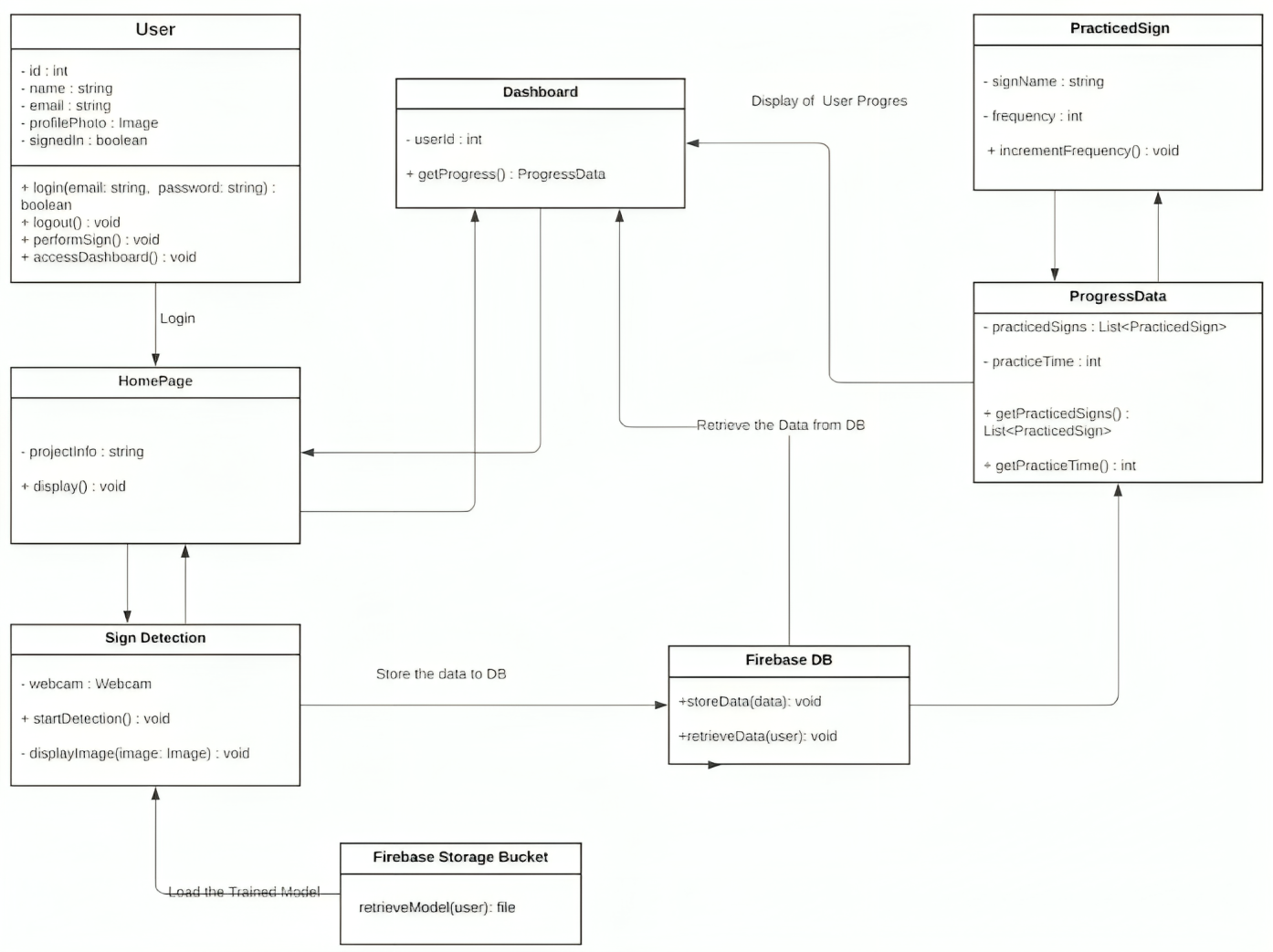
1. **Technical:** The purpose of the course of action can be performed with the existing hardware, software and the technical expertise. In short, it does not require any additional Hardware and Software. The project is feasible in technical remarks. The proposed system can run on any machine however, works best on the software and hardware that had been used while developing the system. Thus, it would be feasible in all technical terms of feasibility.
2. **Operational:** Operational feasibility is the ability to utilize, support, and as well as to perform the necessary tasks of the system. This project will be operationally feasible because it will be easy and efficient for an user to understand the alphabets using hand gestures. Also, the application will be able to operate with few human resources.
3. **Economic:** The system that will be developed is economical because the tools that will be used are open source, so you will not need to purchase additional software or hardware. It is also cost-effective because it will eliminate the paperwork, and the project can be run. Also, the software and hardware resources required to run the project are already available to us. No new system should be deployed. Thus, it is highly economical.

**iii. Schedule:**

**Table 3. 1 Schedule Feasibility**



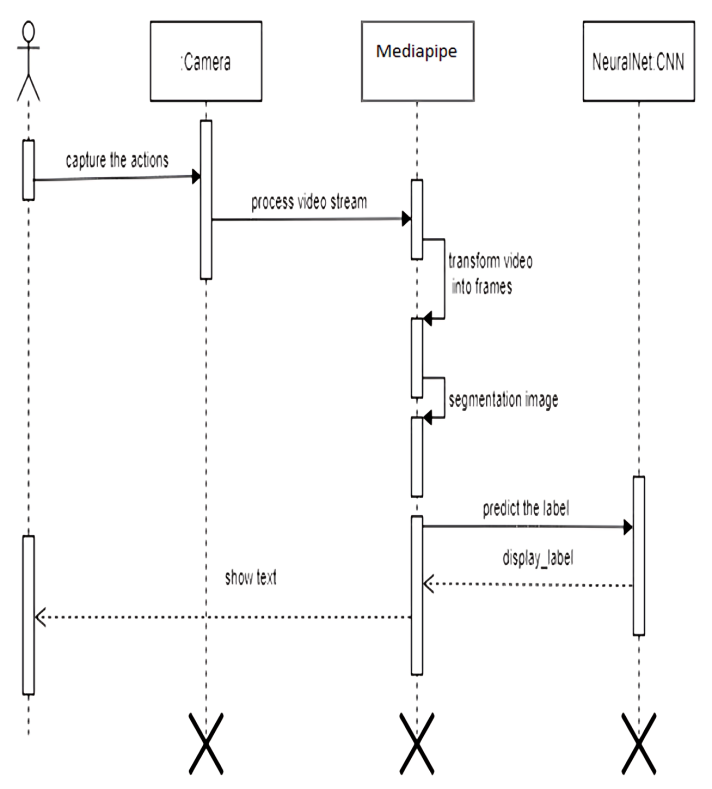
### Object Modeling: Object & Class Diagram

****

**Figure 3. 2 Class Diagram of Sign Language Recognition**

### Dynamic Modeling: State & Sequence diagram:

The Sequence Diagram illustrates the interactions and message passing between objects during a specific scenario.



**Figure 3. 3 Sequence diagram of Sign Language Recognition**

### Process modeling: Activity Diagram

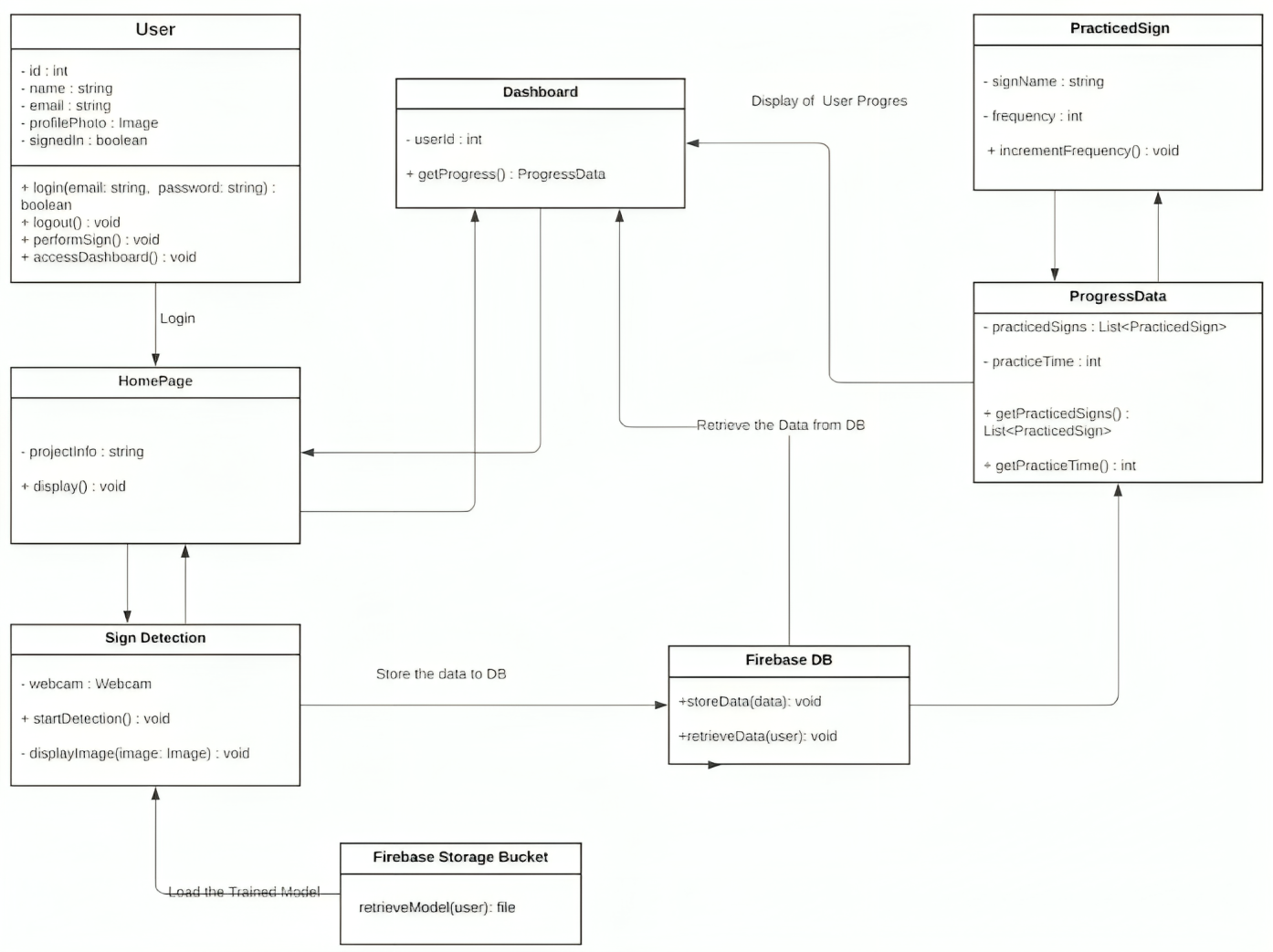
### 

**Figure 3. 4 Activity Diagram of Sign Language Recognition**

## System Design:

System design shows the overall design of the system. In this section, we discuss in detail the design aspects of the system.

### Refinement of class and object

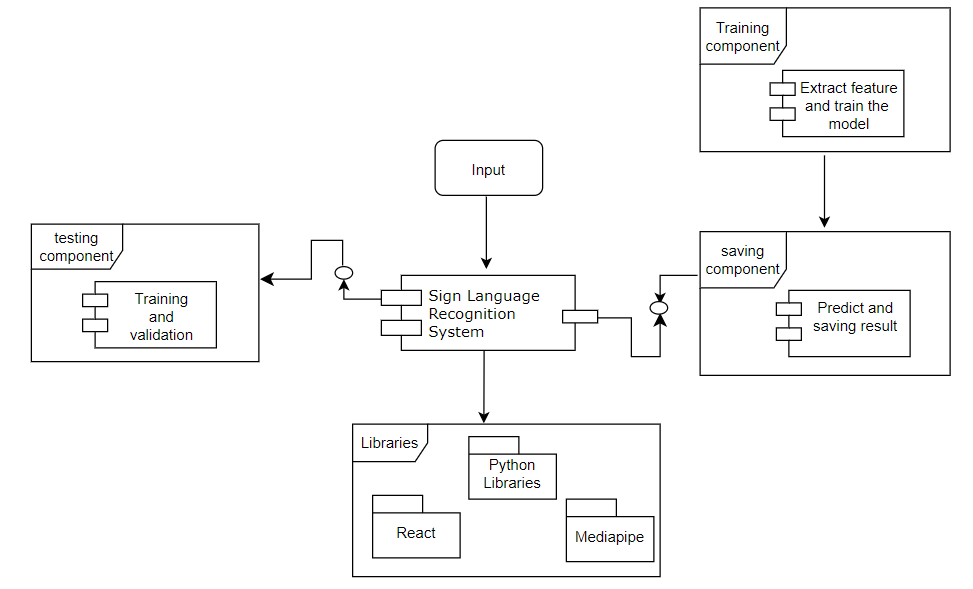
****

**Figure 3. 5 Refinement Diagram of Sign Language Recognition**

This is how the whole system of SLC works . Users provide gestures as input in camera and after getting video stream OpenCV will segment the image and send it for feature extraction. After extracting the features and training it for classification it is ready to classify the gesture.

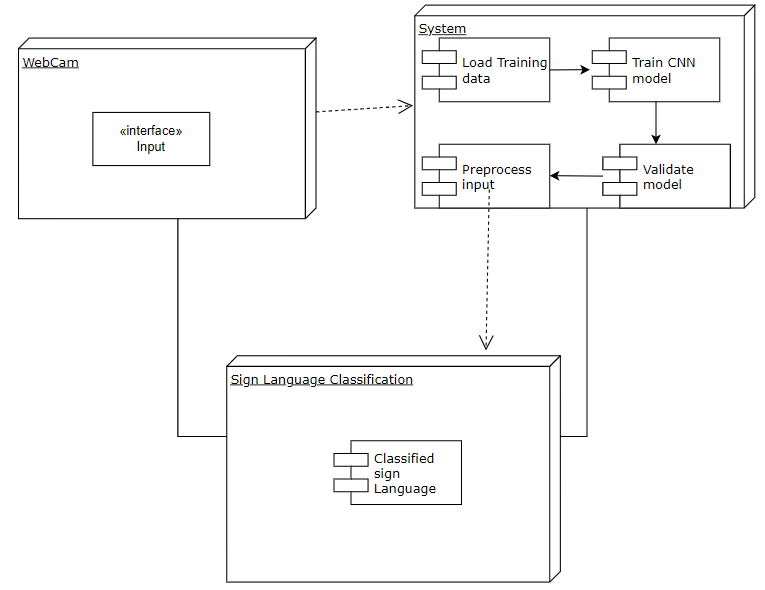
### Component diagrams

This Component diagram/UML diagram shows the implementation view of the system and shows the internal working of the system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



**Figure 3. 6 Component diagram of Sign Language Recognition**

### Deployment diagram

****

**Figure 3.** 7 **Deployment Diagram of Sign Language Recognition**

## Algorithm details

The sign language detection system employs Convolutional Neural Networks (CNNs) to recognize hand gestures. Initially, images or videos of hand gestures are collected and

annotated, creating a labeled dataset for training. In the pre-processing phase, images are resized and cleaned to ensure consistent input for the CNN.

Segmentation isolates the hand from the background, focusing the CNN on the relevant parts of the image. The CNN, consisting of convolutional, pooling, and fully connected layers, extracts features and classifies gestures based on training data. Training involves optimizing the network using techniques like Adam or Stochastic Gradient Descent.

Once trained, the CNN classifies new gestures and outputs the results as text or audio. Post-processing techniques, such as smoothing predictions, enhance accuracy, especially in video data. The user interface displays the recognized gestures in real-time, and the system continuously improves by retraining the CNN with new data and incorporating user feedback.

* **Convolutional Layer**: This layer applies convolution operations to the input image using filters (kernels).

(I∗K)(i, j)=∑ ∑I (i+m, j+n) \*K (m, n)

m n

Where I is the input image, K is the kernel, and i, j are the spatial coordinates. The kernel slides over the input image to produce a feature map.

* **Activation Function**: Introduces non-linearity into the model.

ReLU(*x*) =max (0*, x*)

* **Pooling Layer**: Reduces the spatial dimensions of the feature maps to decrease the number of parameters and computation in the network.

P (i, j) = max A (i\*k+m, j \*k+n) 0≤m<k, 0≤n<k

Where 𝑘 is the pooling window size.

* **Fully Connected Layer**: Flattens the input and passes it through one or more dense layers.

Z=W\*X+*b*

Where X is the input, W is the weight matrix, b is the bias, and Z is the output.

* **Softmax Layer**: Converts the raw scores into probabilities.

Softmax (*zi*) =ezi/∑*jezj*

Where zi is the i element of the output vector Z.

**Pros of CNNs:**

1. **Feature Extraction:** Automatically detects features (like edges, textures) without needing manual extraction.
2. **Spatial Hierarchy:** Learns hierarchical patterns (low-level to high-level) which is good for image tasks.
3. **Parameter Sharing:** Uses fewer parameters compared to fully connected networks, making it more efficient.
4. **Translation Invariance:** Better at recognizing objects in different positions and orientations.

**Pros of CNNs:**

* **Feature Extraction:** Automatically detects features (like edges, textures) without needing manual extraction.
* **Spatial Hierarchy:** Learns hierarchical patterns (low-level to high-level) which is good for image tasks.
* **Parameter Sharing:** Uses fewer parameters compared to fully connected networks, making it more efficient.
* **Translation Invariance:** Better at recognizing objects in different positions and orientations.

**Cons of CNNs:**

* **Data Requirement:** Needs a large amount of labeled data to perform well.
* **Computational Cost:** Requires significant computational power and memory.
* **Complexity:** Can be complex to design and tune, especially deep networks.
* **Overfitting Risk:** Can overfit if not properly regularized or if the dataset is too small.

# CHAPTER 4 IMPLEMENTATION AND TESTING

## Implementation

* + 1. **Tools used Python**

Python is a high-level, general-purpose programming language. It is dynamically types and supports multiple programming paradigms, including structured, object-oriented, and functional programming. Python is extensively applied in data analysis, machine learning, data engineering and other fields. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

**VS code**

Visual Studio Code is a lightweight but powerful source code editor made by Microsoft with Electron Framework for Windows, Linux, and Mac-OS.

**MediaPipe**  
MediaPipe is a framework developed by Google for building multimodal machine learning pipelines, especially useful for real-time video processing and gesture recognition. It provides pre-trained models and solutions for tasks like hand tracking, face detection, and pose estimation.

**React**

React is a popular JavaScript library for building user interfaces, particularly single-page applications. Developed by Facebook, React enables developers to create reusable UI components and manage application state efficiently, making it a key technology for front-end development.

**Draw.io**

Draw.io is an open source technology stack for building diagramming applications, and the world’s most widely used browser-based end-user diagramming software.

**Firebase**  
Firebase is a platform developed by Google for building mobile and web applications. It offers features such as Google Authentication for user sign-in, real-time databases, and cloud storage. Firebase is also commonly used for model storage in machine learning applications, allowing seamless integration with mobile or web apps to deploy trained models and manage user authentication.

### Implementation details of modules

The implementation of the modules plays a crucial role in the process as sign language recognition is achieved through the combination of MediaPipe's real-time hand tracking and a Convolutional Neural Network (CNN) to recognize the specific gestures or signs. Below is a detailed explanation of the key components and functions used to achieve this:

**MediaPipe Hands** - MediaPipe's hand-tracking solution is used to detect and track the hand landmarks in real-time. The hand landmarks provide 21 points for each detected hand, which serves as the main input to the CNN model. The landmarks are the coordinates of the fingers and palm, offering a high-level representation of hand poses.

**Landmark Extraction** - After detecting the hand, the landmark extraction module collects the 3D coordinates (x, y, z) of the hand landmarks. These coordinates are normalized and flattened to be passed into the next stage of the model, reducing the need for raw image inputs while preserving key spatial information.

**Sequential()** - The Sequential model is used to build a linear stack of layers. In this case, it takes the flattened hand landmarks as inputs rather than pixel data from the images, processing the gestures through the neural network's various layers.

**Conv2D Layer (optional)** - If working with image inputs, Conv2D layers can be used in the preprocessing stage, where filters are applied to capture important spatial features from the image. However, when using MediaPipe landmarks, this step can be skipped as the landmarks serve as high-level feature inputs.

**model.add(Dense())** - The Dense layers take the normalized hand landmark points and create mappings between the hand's position and a specific sign. These layers help recognize patterns in the hand poses that correspond to various signs in the language.

**Plot()** - This function can be used to plot key points in the gesture recognition pipeline, such as the landmarks detected by MediaPipe or the accuracy/loss graphs during the training of the CNN model. For example, parameter 1 could be the 3D points of the hand in the x-axis, while parameter 2 could be the corresponding y-axis values for visualization.

**MaxPooling2D() (if using image data)** - When image data is involved, MaxPooling is used after the convolutional layers to reduce the spatial size of the representation and to focus on the most important features. However, if MediaPipe landmarks are directly used, pooling isn't required.

**Dropout()** - Dropout is still applied to prevent overfitting. During training, some neurons are randomly dropped in the dense layers to ensure the model does not memorize the hand positions for specific signs, improving its generalization capability.

**Flatten()** - This function is used to flatten the MediaPipe hand landmark coordinates from 3D to 1D before passing them into the Dense layers, similar to how it flattens the output from CNN layers when using image inputs.

**Final Dense Layer** - This layer uses the learned features from previous layers (whether it’s image features from Conv2D or hand landmarks from MediaPipe) and maps them to the final output class, i.e., the sign language label. For example, the output will be one of the predefined hand signs such as "hello," "thank you," or specific letters of the alphabet.

.

## Testing

### Test cases for Unit testing

**Table 4. 1 Unit Testing for loading model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Test case** | **Input description** | **Expected output** | **Result** | **Test status** |
| 1 | Loading model | Initializing trained model and load it. | Loaded model without errors | Model was loaded without any error. | pass |

**Table 4. 2 Unit testing for converting video to frame**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.N** | **Test**  **case** | **Input**  **description** | | **Expected**  **output** | | **Result** | **Test status** |
| 1 | Conve rting video to  frames | Capturing video converting into frames | and  it | Image frames captured video  stream | of | Frames of images were not generated from the video. | Fail |
| 2 | Conve rting video to  frames | Capturing video converting into frames | and  it | Image frames captured video  stream | of | Frames of images were generated from the video. | pass |

**Table 4. 3 Unit testing for recognizing Hand Gesture**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N** | **Test case** | **Input**  **description** | **Expected**  **output** | **Result** | **Test**  **status** |
| 1 | Recognize hand gesture | Image frame that contains hand object | Shows the result according to hand gesture provides as  input | gestures were not recognized correctly | Fail |
| 2 | Recognize hand gesture | Image frame that contains hand object | Shows the result according to hand gesture provides as  input | gestures were recognized correctly | pass |

**Table 4. 4 Unit Testing for Training and Saving the Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Test case** | **Input description** | **Expectedoutput** | **Result** | **Test status** |
| 1 | Training and saving the model | Training code for the model was implemented  along with saving it in a  Local file. | Model will be trained without any errors and will be saved  into the local file System. | Model was trained and  was found saved on  local file system | Pass |

**Table 4. 5 System testing for Classifying the Sign Language**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N** | **Test case** | **Input**  **description** | **Expected**  **output** | **Result** | **Test**  **status** |
| 1 | Classify | Hand gestures | Shows the | gestures were | pass |
|  | the sign | were provided | result | recognized |  |
|  | language |  | according to | correctly |  |
|  | correctly |  | hand gesture |  |  |
|  |  |  | provides as |  |  |
|  |  |  | input |  |  |

* 1. **Result Analysis**

Initially, a Convolutional Neural Network (CNN) model was developed and evaluated using a static sign language dataset, which resulted in remarkably high accuracy. However, when the same CNN was applied to dynamic sign language sequences processed as temporal data, it struggled to learn the gestures effectively. Instead, the model exhibited a tendency to focus on recognizing individuals and facial features rather than the specific sign language gestures.

To address this challenge, various hyperparameters were tested to enhance the model's ability to learn the gestures. Unfortunately, these adjustments led to a significant decline in performance. This degradation was further exacerbated by noise in the data, as unintended gestures often overlapped with the target gestures within the sequences.

Following a thorough cleaning and filtering of the dataset to isolate only the relevant sign language gestures, some improvements were observed. The model demonstrated a better capacity to identify the gestures, indicating that focused data preprocessing played a critical role in enhancing the effectiveness of the CNN for sign language recognition.

**CHAPTER 5**

**CONCLUSION AND FUTURE RECOMMENDATIONS**

## Conclusion

Hand gestures are a powerful way for human communication, with lots of potential applications in the area of human computer interaction. Vision-based hand gesture recognition techniques have many proven advantages compared with traditional devices. However, hand gesture recognition is a difficult problem and the current work is only a small contribution towards achieving the results needed in the field of sign language recognition. This vision-based system is able to interpret static hand gestures from the American Sign Language. The report describes the implementation of a system for recognizing sign language. The input consists of a video of the gesture to be recognized. It is recorded and preprocessed which is then sent to the model. The model returns the meaning of the sign contained in the video as output. Such a system will be helpful to many institutions who depend solely on manual translators.

## Future Recommendations

The proposed sign language recognition system used to recognize sign language letters can be further extended to recognize gestures and facial expressions. Instead of displaying letter labels it will be more appropriate to display sentences as more appropriate translation of language. This also increases readability. The scope of different sign languages can be increased. More training data can be added to detect the letter with more accuracy. This project can further be extended to convert the signs to speech.

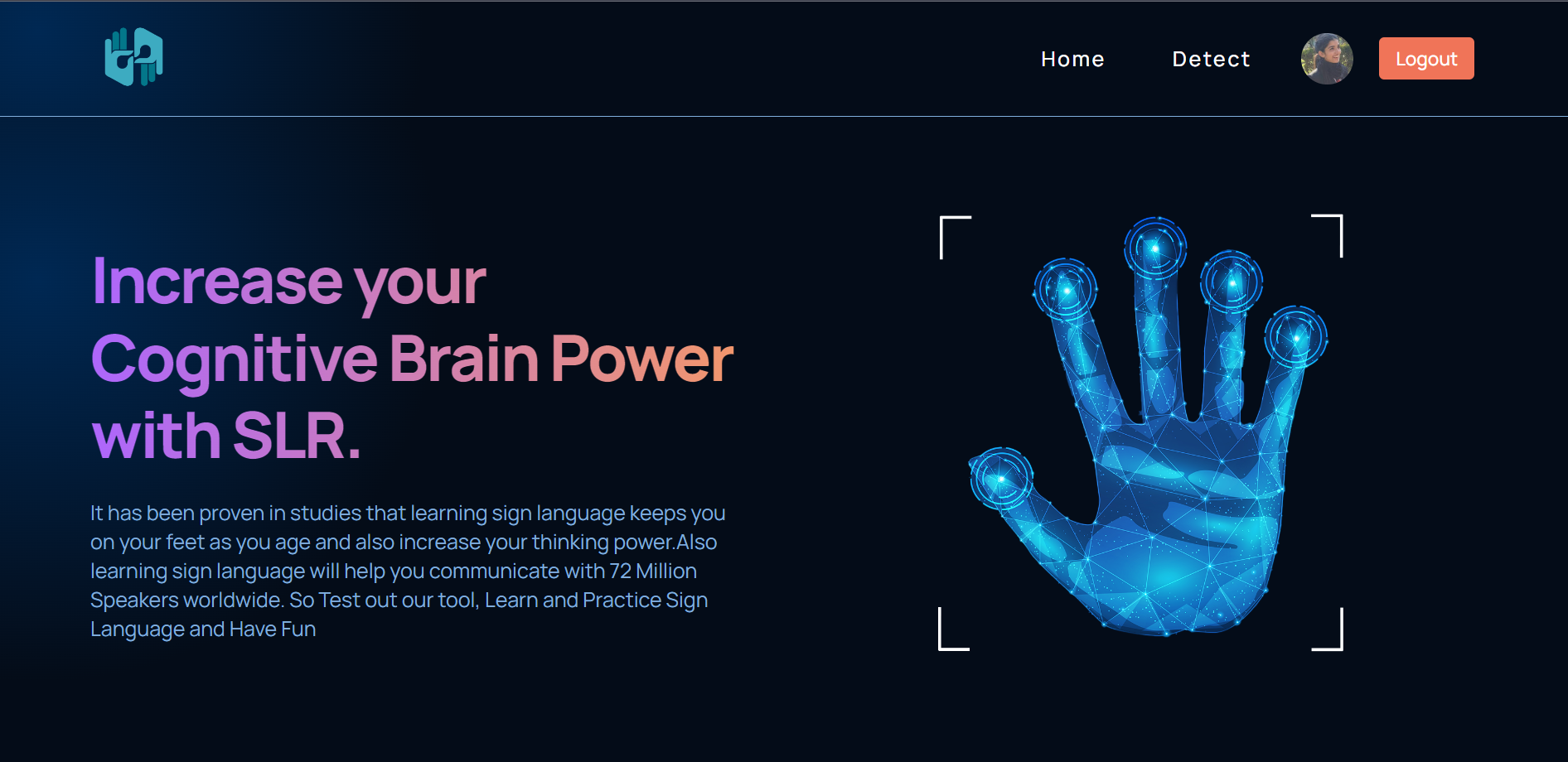
# REFERENCES

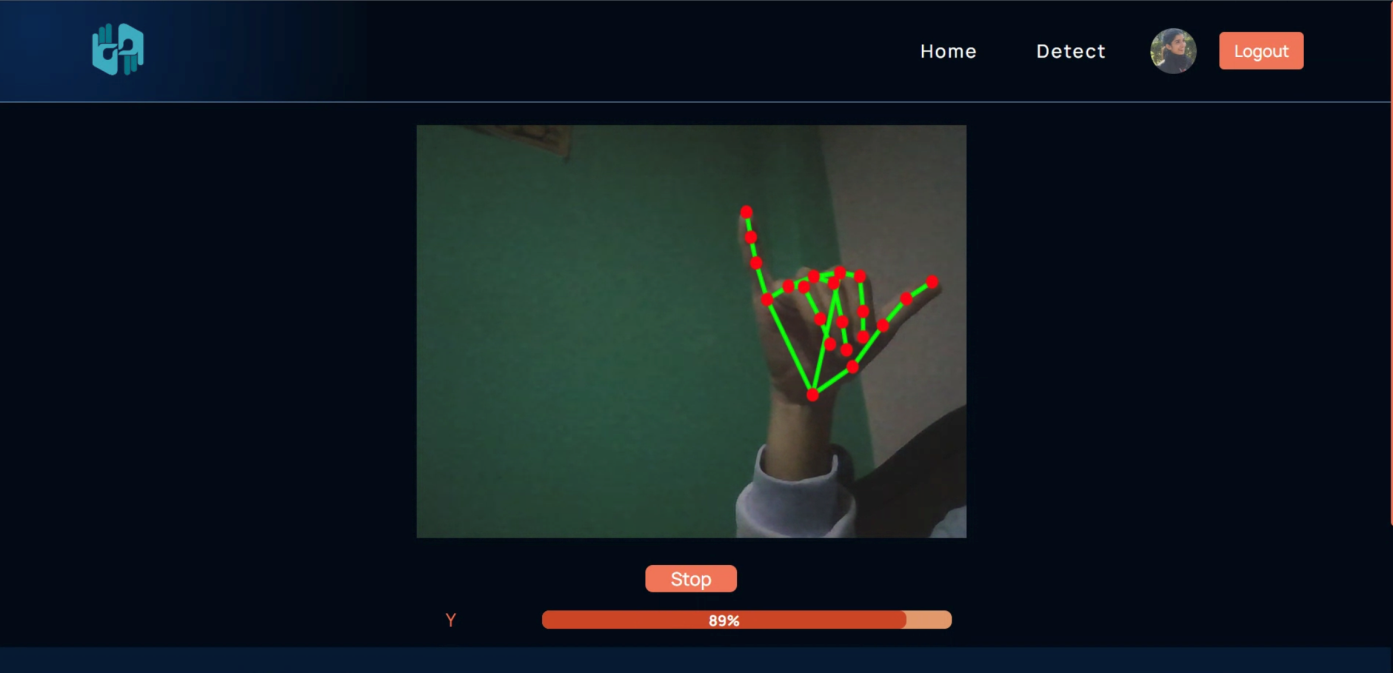
1. Papastratis, I.; Chatzikonstantinou, C.; Konstantinidis, D.; Dimitropoulos, K.; Daras,

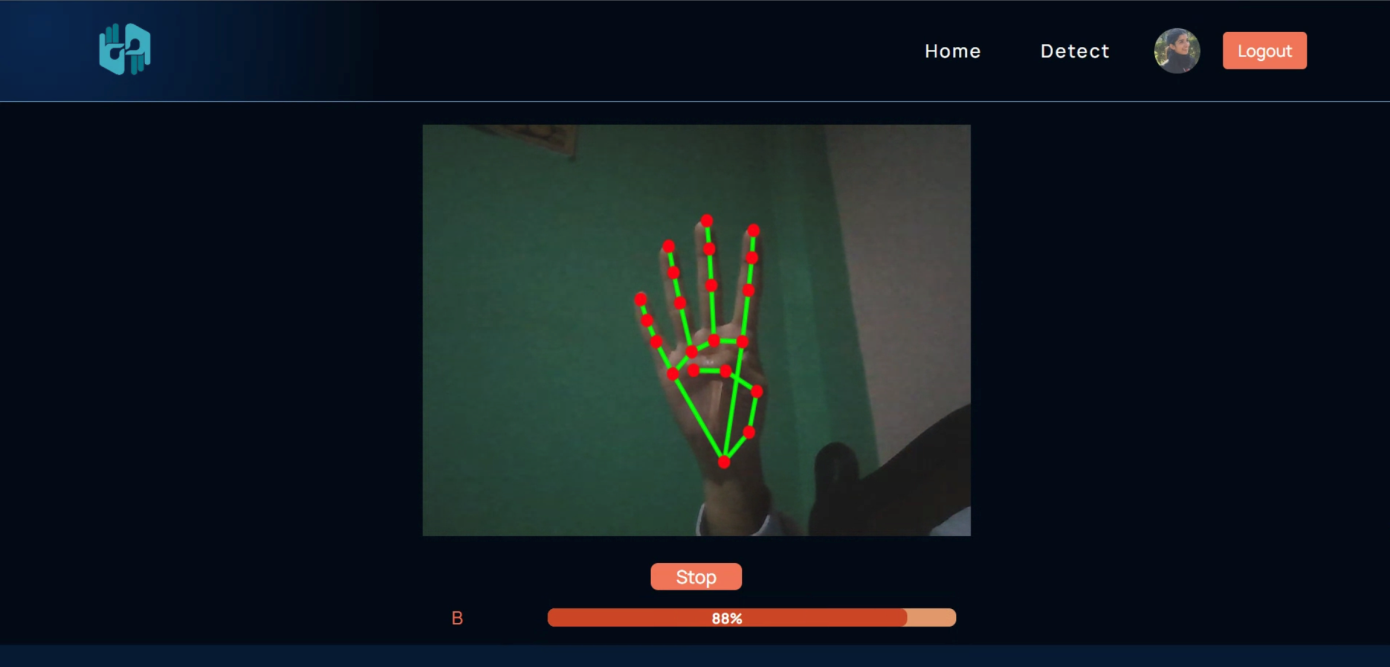
P. Artificial Intelligence Technologies for Sign Language. Sensors 2021, 21, 5843. [CrossRef] [PubMed]

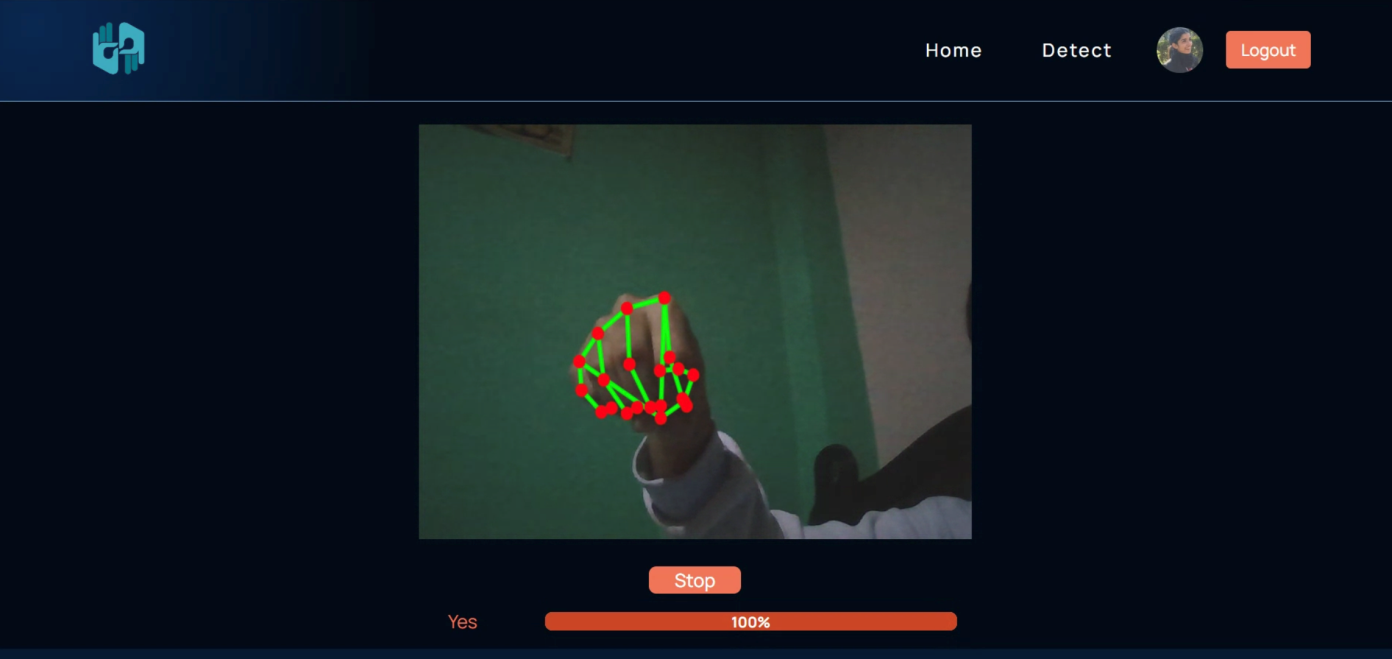
1. S. Bele and K. Sharma, "SIGN LANGUAGE RECOGNITION SYSTEM USING MACHINE LEARNING," *IJIRM*, vol. 7, no. 1, 2023.
2. Nandy, A.; Prasad, J.; Mondal, S.; Chakraborty, P.; Nandi, G. Recognition of Isolated Indian Sign Language Gesture in Real Time.Commun. Comput. Inf. Sci. 2010, 70, 102– 107.
3. Mekala, P.; Gao, Y.; Fan, J.; Davari, A. Real-time sign language recognition based on neural network architecture. In Proceedings of the IEEE 43rd Southeastern Symposium on System Theory, Auburn, AL, USA, 14–16 March 2011.
4. M. Muchuda, "Sign Language Recognition System," *ResearchGate*, 2023. [Online]. Available:[https://www.researchgate.net/publication/374199430\_Sign\_Language\_Recogni](https://www.researchgate.net/publication/374199430_Sign_Language_Recognition_System) [tion\_System.](https://www.researchgate.net/publication/374199430_Sign_Language_Recognition_System) [Accessed: 27-Jul-2024].
5. F. Ribeiro and P. Trigueiros, "Vision-based Portuguese Sign Language Recognition System," *CORE*. [Online]. Available: <https://core.ac.uk/download/pdf/55638597.pdf>. [Accessed: 27-Jul-2024].
6. A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural Computing and Applications*, vol. 32, pp. 7957–7968, Jun. 2020, doi: 10.1007/s00521-019-04691-y.
7. K. Amrutha and P. Prabu, "ML Based Sign Language Recognition System," *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, Kottayam, India, 2021, pp. 1-5, doi: 10.1109/ICITIIT51526.2021.9399594.
8. B. Bauer and K.-F. Kraiss, "Towards an Automatic Sign Language Recognition System Using Subunits," Gesture and Sign Language in Human-Computer Interaction, I. Wachsmuth and T. Sowa, Eds. Berlin, Germany: Springer-Verlag, 2002, vol. 2298.

# APPENDICES

****



****

****