A PROJECT REPORT ON

Empowering Communication: Gesture-Based Language to Text System for Divyang Individuals

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING
IN
Artificial Intelligence and Data Science

BY

Gauravi Vasantrao Gawale	B400130793
Darshan Sunil Kedare	B400130809
Pooja Shantaram Gavit	B400130791
Rohan Babasahaeb Kaitake	B400130804

UNDER THE GUIDANCE OF
Prof. Smita.T.Patil

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
K.K.WAGH INSTITUTE OF ENGINEERING EDUCATION AND RESEARCH, HIRABAI HARIDAS VIDYANAGARI, PANCHAVATI, NASHIK 422003

2024-25

CERTIFICATE

This is to certify that the project report entitled

Empowering Communication: Gesture-Based Language to Text System for Divyang Individuals

Submitted by

Gauravi Vasantrao Gawale	B400130793
Darshan Sunil Kedare	B400130809
Pooja Shantaram Gavit	B400130791
Rohan Babasahaeb Kaitake	B400130804

is a bonafide work carried out by them under the supervision of Prof. **Smita. T. Patil** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Artificial Intelligence and Data Science).

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. S. T. Patil
Internal Guide

Prof. Dr. S. M. Kamalapur
Head of Department
Department of AI and DS

External Examiner

Date: / /2025
Place: Nashik

Abstract

Sign language is a vital form of communication for the deaf and hard-of-hearing community, particularly among Divyang individuals. To bridge the communication gap between sign language users and non-signers, we propose a real-time sign language detection system utilizing standard web cameras. The system recognizes sign language gestures made in front of the camera and converts them into on-screen text. By leveraging advanced computer vision and deep learning techniques, it captures and analyzes hand and facial gestures, identifying key markers corresponding to specific signs. Machine learning algorithms then translate these markers into the sign language vocabulary, providing accurate text output for real-time interpretation. This system, designed to empower Divyang individuals, can be integrated into standard cameras and other devices, with potential applications in education, healthcare, and beyond, enhancing communication accessibility for sign language users.

Acknowledgment

We would like to express our sincere gratitude to all those who have supported and guided us through the requirement analysis and detailed design phases of this system. First and foremost, we extend our heartfelt thanks to our project guide, **Prof. Smita T. Patil**, for her invaluable guidance, insightful feedback, and constant encouragement. Her expertise and support have been instrumental in shaping the direction of our project.

We also wish to thank **K. K. Wagh College of Engineering, Education Research. Nashik**, for providing the resources and infrastructure necessary to conduct this research. Special thanks to the faculty and staff for their assistance and support during these initial phases.

Our appreciation goes out to our peers and colleagues for their constructive discussions and collaboration, which have greatly enriched our system experience. Lastly, we are grateful to our families and friends for their unwavering support and understanding, which has been a continuous source of motivation for us.

Thank you all for your contributions and encouragement during this crucial phase of our system.

Gauravi Gawale

Darshan Kedare

Pooja Gavit

Rohan Kaitake

(B.E. AI and DS Engg.)

INDEX

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 PROJECT IDEA

To improve communication between sign language users and those unfamiliar with it, we propose a real-time sign-to-text translation system. This system will recognize and translate sign language gestures into on-screen text, providing an effective and accessible means of communication for Divyang individuals. By leveraging computer vision and deep learning techniques, the system will accurately capture and analyze hand and facial gestures, identifying key markers corresponding to specific signs. The machine learning algorithms will then convert these markers into text output, creating a seamless bridge for communication. This one-way translation system has potential applications in education, healthcare, and other fields, promoting inclusivity by facilitating better understanding between signers and non-signers.

1.2 MOTIVATION OF THE PROJECT

motivation behind this system is to break down communication barriers between the deaf or hard-of-hearing community, particularly Divyang individuals, and those who do not know sign language. Effective communication is essential for participation in everyday activities, but sign language users often face challenges in environments where others do not understand their language. By creating a system that translates sign language gestures into on-screen text in real-time, we aim to promote inclusivity and make communication accessible to everyone, regardless of language abilities. This system has the potential to impact areas such as education, healthcare, and social interaction, providing a tool for better understanding and connection for sign language users and non-signers alike.

CHAPTER 2

LITERATURE SURVEY

1. **Paper Name:** Real-Time American Sign Language Recognition Using Deep Learning

   Contribution: This paper presents a real-time American Sign Language (ASL) recognition system using deep learning techniques. It explores the use of Convolutional Neural Networks (CNNs) to detect hand gestures and analyze sign language in real-time. The paper contributes by evaluating various CNN architectures and demonstrating how deep learning can be used for effective sign language interpretation.

   Publisher: IEEE Access

   Year of Publish: 2021

2. **Paper Name:** Sign Language Recognition: A Review of the State of the Art

   Contribution: This paper provides a comprehensive review of sign language recognition systems, focusing on the use of computer vision and deep learning techniques. It outlines different methodologies, datasets, and algorithms used in the field. The paper's contribution is the identification of key challenges and future research directions in automatic sign language recognition.

   Publisher: Elsevier (Pattern Recognition)

   Year of Publish: 2021

3. **Paper Name:** Hand Gesture Recognition for Sign Language: A Comprehensive Survey

   Contribution: This paper surveys recent advancements in hand gesture recognition for sign language using machine learning and computer vision. It covers different sensor-based and vision-based approaches, highlighting the strengths and limitations of various methods. The paper contributes to the literature by providing an in-depth analysis of the technology landscape and identifying gaps in the field.

   Publisher: Springer (Neural Computing and Applications)

   Year of Publish: 2022

4. **Paper Name:** Real-Time Sign Language Recognition Using 3D Convolutional Neural Networks

   Contribution: The paper introduces a real-time sign language recognition system utilizing 3D Convolutional Neural Networks (3D CNNs) for improved gesture detection accuracy. It contributes by integrating spatial and temporal features of sign language gestures, providing a robust solution for continuous sign language recognition. The study also presents a comparative analysis with traditional 2D CNN approaches.

   Publisher: MDPI (Sensors)

   Year of Publish: 2022

CHAPTER 3

PROBLEM DEFINITION AND SCOPE

3.1 PROBLEM STATEMENT

The communication gap between sign language users and non-signers presents significant barriers in daily interactions, limiting accessibility and inclusivity. Existing solutions are either expensive or require specialized hardware, making them impractical for widespread use. There is a need for an affordable, real-time system that can translate sign language into voice and text using standard cameras. This system addresses that gap by leveraging computer vision and machine learning to create an accessible, bidirectional communication tool for sign language users and non-signers.

3.1.1 Goals and objectives

Goal and Objectives:

- To develop a real-time sign language detection system using standard web cameras for gesture recognition.
- To implement advanced computer vision techniques for accurately capturing and analyzing hand and facial gestures.
- To integrate machine learning algorithms that translate recognized gestures into on-screen text.
- To ensure the system is easily accessible and scalable, with potential integration into various camera-equipped devices for use in diverse settings like education and healthcare.

3.1.2 Assumption and Scope

- The system assumes that the user will perform sign language gestures clearly in front of the camera within a well-lit environment for accurate detection.
- It is assumed that the standard web camera has sufficient resolution and frame rate to capture detailed hand and facial movements necessary for sign language recognition.

- The system assumes that it will primarily be used for translating common sign language gestures and may require additional training or updates for regional or less common signs.
- It is assumed that users have access to devices equipped with a visual display to fully utilize the system's text output functionality.

3.2 METHODOLOGY

The methodology for this real-time sign language detection system involves several key steps. First, the system uses a standard web camera to capture live video feed of the user's gestures. Advanced computer vision techniques, such as hand and facial landmark detection, are applied to identify and track key points of interest, including hand movements and facial expressions. The captured gestures are then processed using deep learning models specifically trained on sign language datasets to recognize specific signs. A machine learning algorithm maps these gestures to corresponding words in a sign language vocabulary, translating them into on-screen text in real time. The system's architecture ensures low-latency processing, making it suitable for effective communication. The model will undergo continuous training and fine-tuning to improve accuracy, and the system can be integrated into various devices and applications to enhance accessibility across different sectors, such as education and healthcare.

3.3 OUTCOME

- A functional system capable of real-time sign language detection and translation into on-screen text.
- Improved communication between sign language users and non-signers, enhancing inclusivity in various settings.
- A user-friendly interface that can be easily integrated into standard web cameras and other camera-equipped devices.

- Enhanced accessibility in sectors like education, healthcare, and customer service, enabling better interaction with the deaf and hard-of-hearing community.
- Continuous improvement of the system's accuracy through machine learning, ensuring the recognition of a wide range of sign language gestures.

3.4 TYPE OF PROJECT

- Machine Learning

CHAPTER 4

PROJECT PLAN

4.1 PROJECT TIMELINE

Sr. No.	Task	Start Date	End Date
1.	Primary Survey	21/06/2024	28/06/2024
2.	Introduction and ProblemStatement	31/07/2024	05/08/2024
3.	Literature Survey	14/08/2024	21/08/2024
4.	Project Statement	23/09/2024	14/10/2024
5.	Software Requirement andSpecification	28/10/2024	02/11/2024
6.	System Design	04/11/2024	07/12/2024
7.	Architecture Design	18/01/2025	24/01/2025
8.	Synopsis Submission	01/02/2025	07/02/2025
9.	Implementation	21/02/2025	13/03/2025
10.	Deployment	14/03/2025	19/03/2025
11.	Testing	21/03/2025	28/03/2025
12.	Paper Publish	Under Review	
13.	Report Submission	20/04/2025	20/04/2025

Table 4.1: System Implementation Plan

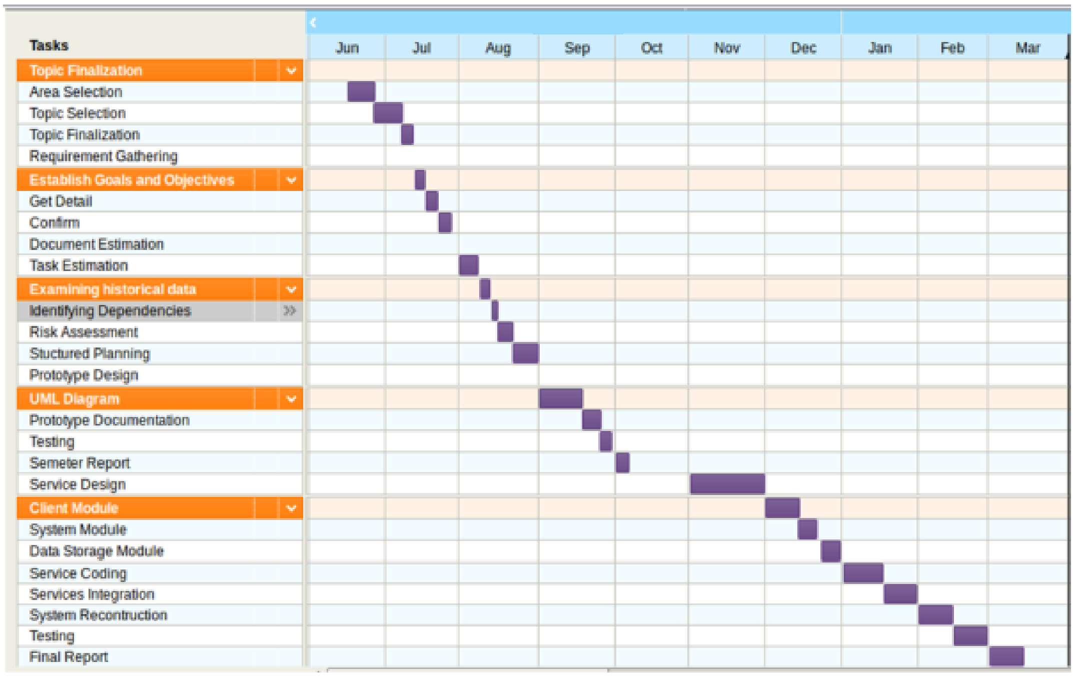

Figure 4.1: Management reporting and communication

4.2 TEAM ORGANIZATION

The manner in which staff is organized and the mechanisms for reporting are noted.

4.2.1 Team structure

- Front end development : Darshan Kedare & Gauravi Gawale
- Back end development : Gauravi Gawale & Darshan Kedare
- Documentation : Pooja Gavit & Rohan Kaitake
- Project Survey : Pooja Gavit & Rohan Kaitake

CHAPTER 5

SOFTWARE REQUIREMENT SPECIFICATION

5.1 FUNCTIONAL REQUIREMENTS

- The system must accurately detect and recognize hand and facial gestures performed by the user in real time.
- The recognized sign language gestures must be translated into on-screen text.
- The system must integrate with standard web cameras to capture live video feeds of the user's gestures.
- The system must provide a user-friendly interface for displaying the translated text and managing settings.

5.2 NON FUNCTIONAL REQUIREMENTS

- The system must process gestures and provide translations with minimal delay to enable real-time communication.
- The system should be easily scalable to support different devices and environments, such as smartphones, tablets, and laptops.
- The system must maintain a high level of accuracy in gesture recognition to ensure reliable translations.
- The system should be easy to use for individuals with varying levels of technical skill, ensuring it can be widely adopted.

5.3 CONSTRAINTS

- The system's accuracy may be affected by poor lighting, as clear visibility of hand and facial gestures is essential for proper recognition.
- The system relies on the resolution and frame rate of the web camera, meaning lower-quality cameras may reduce detection accuracy.
- Complex or subtle sign language gestures may be challenging for the system to recognize accurately without extensive training on diverse datasets.
- The system may initially support only one or a few variants of sign language

5.4 HARDWARE REQUIREMENTS

1. Processor: AMD Ryzen 7 / Intel Core i5
2. Speed: Minimum 1.1 GHz
3. RAM: 512 MB (minimum)
4. Hard Disk: 20 GB
5. Keyboard: Standard Keyboard
6. Mouse: Two or Three-Button Mouse
7. Monitor: LED Monitor

5.5 SOFTWARE REQUIREMENTS

1. Operating System : Windows xp/7/8/10/11
2. Software Version : Python 3.10
3. Tools : Notepad++ ,pycharm, VSCode

5.6 INTERFACES

User, Hardware , Software

CHAPTER 6

DETAILED DESIGN

6.1 ARCHITECTURAL DESIGN (BLOCK DIAGRAM)

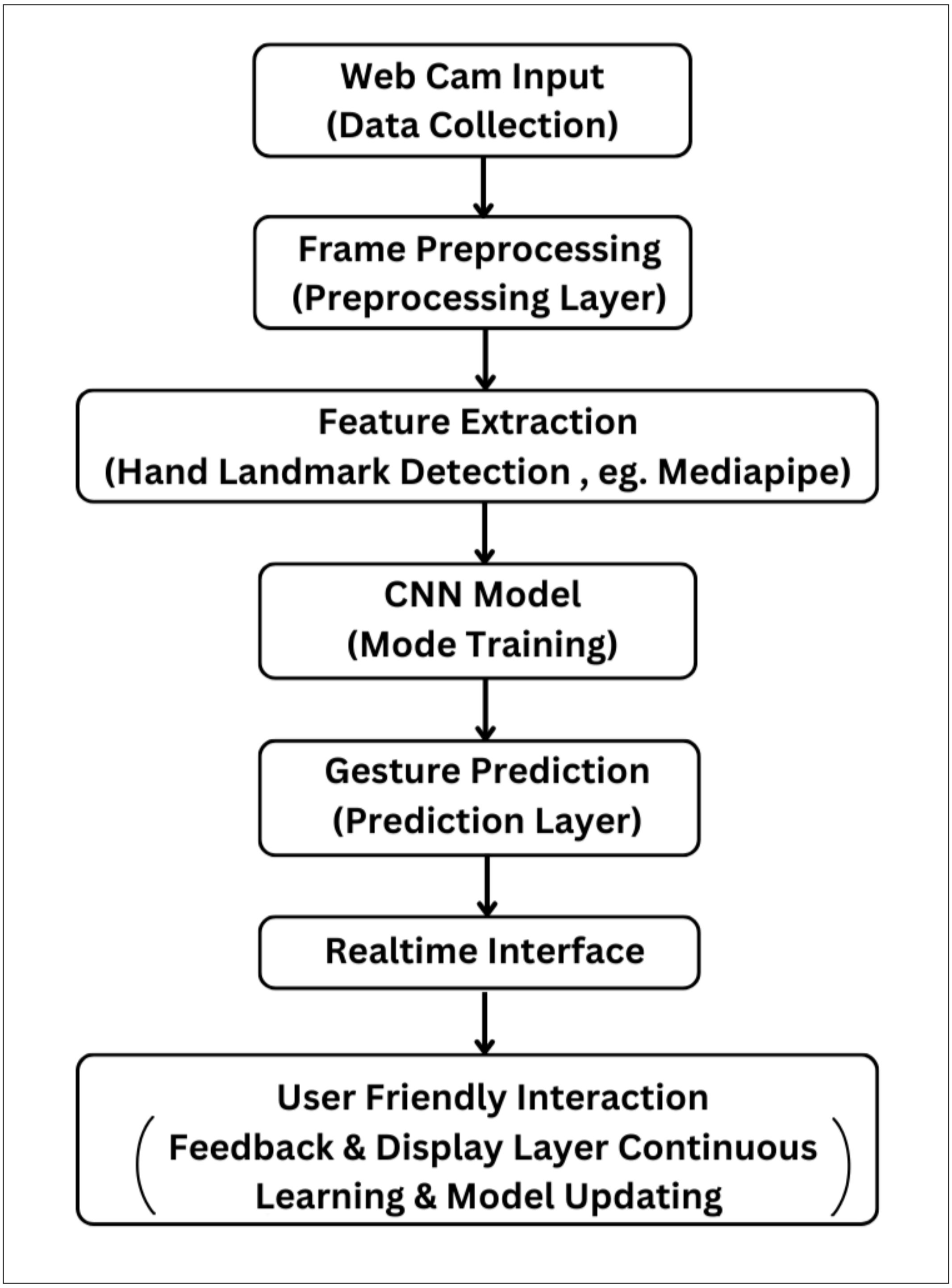

Figure 6.1: System Architecture

The proposed real-time sign language detection system aims to bridge the communication gap between sign language users and non-signers by leveraging advanced computer vision and deep learning technologies. This system utilizes standard web cameras to capture video input of sign language gestures, which are then analyzed to identify key hand and facial markers corresponding to specific signs.

The innovative approach employs machine learning algorithms to translate these gestures into on-screen text, enhancing accessibility and inclusivity. The system is designed with a user-friendly interface that allows for seamless interaction, accommodating users with varying technical expertise. It will support multiple sign languages, enabling broader applicability across different communities.

Additionally, the system incorporates robust security measures to protect user data and privacy while ensuring high performance with low latency and high accuracy in gesture recognition. By integrating features such as customizable settings, the proposed system aims to improve understanding and interaction for sign language users, ultimately fostering greater communication and inclusivity in various environments, including education, healthcare, and public services.

6.2 USER INTERFACE SCREENS

Figure 6.2: User Interface: SignUp Page

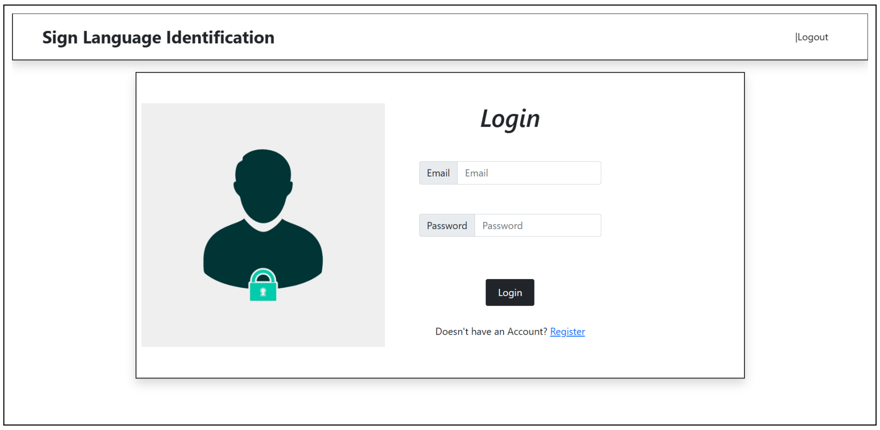

Figure 6.3: User Interface: Login Page

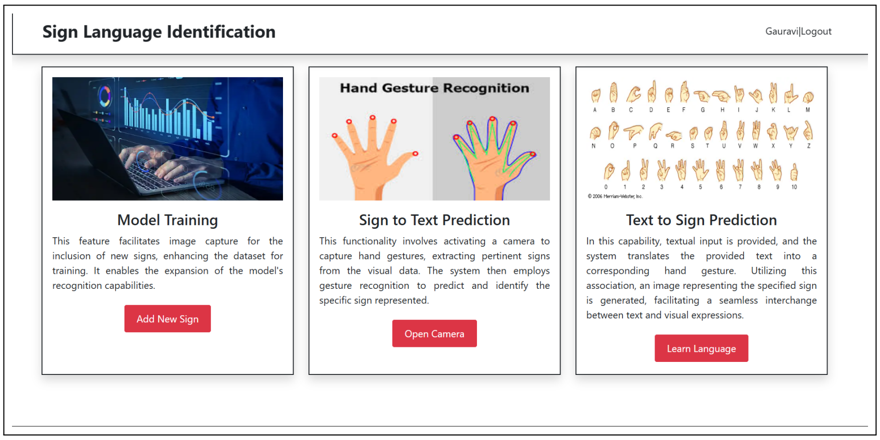

Figure 6.4: User Interface: Main Page

6.3 DATA DESIGN

The system uses a structured dataset of gesture images/videos for training the recognition model. The data is organized in labeled folders where each folder represents a specific gesture. Internally, the system uses NumPy arrays for storing image data during preprocessing and model training.

Temporary data like live camera frames are handled in real-time using OpenCV, without permanent storage. For audio output, recognized text is temporarily passed to the Text-to-Speech module. No external database is used as the application works in real-time and doesn't require long-term storage of user data.

6.3.1 Data structure

In our system, we used basic yet efficient data structures to manage and process the data smoothly.

- Arrays were used to store image frames and gesture data captured from the camera in a structured format.
- Dictionaries (or Hash Maps)were helpful in mapping recognized gestures to corresponding text outputs.
- Lists were used to maintain sequences of detected gestures and manage data flow between modules. These data structures allowed us to handle real-time input and output effectively, ensuring smooth system performance.

6.3.2 Database description

The database for the system stores the gesture data that is used for training the machine learning model. It consists of labeled images or videos of various gestures, each corresponding to a specific sign language symbol. The data is organized in a structured format to ensure easy access and efficient preprocessing. During training, this dataset is used to help the model learn the features of each gesture, enabling accurate recognition. The database also includes metadata like timestamps, gesture types, and user information to support further analysis and testing.

6.4 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing.

6.4.1 Level 0 Data Flow Diagram

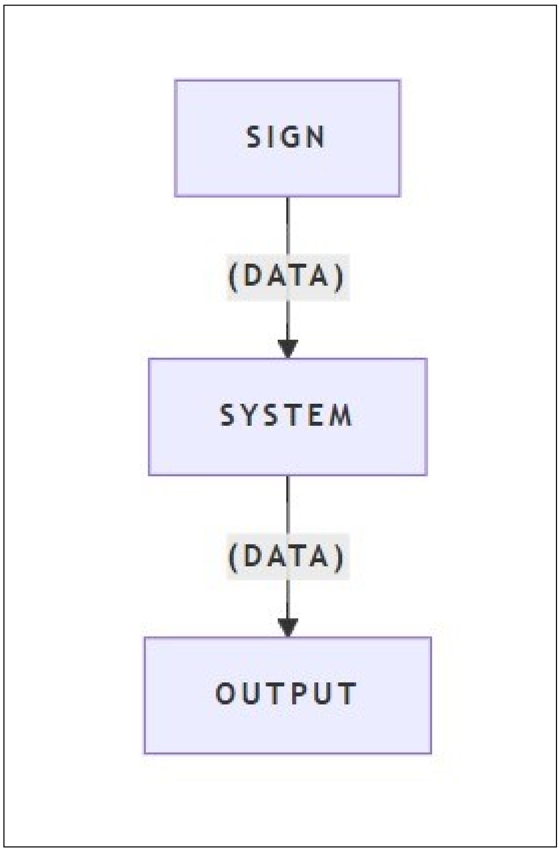

Figure 6.5: Level 0 Data Flow Diagram

6.4.2 Level 1 Data Flow Diagram

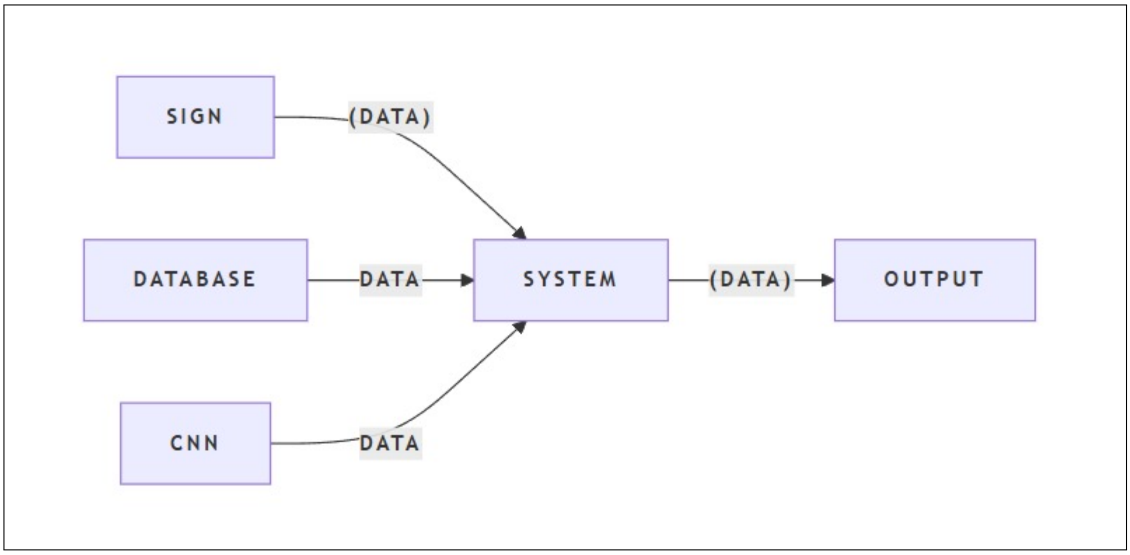

Figure 6.6: Level 1 Data Flow Diagram

6.5 UML DIAGRAMS

6.5.1 Usecase Diagram

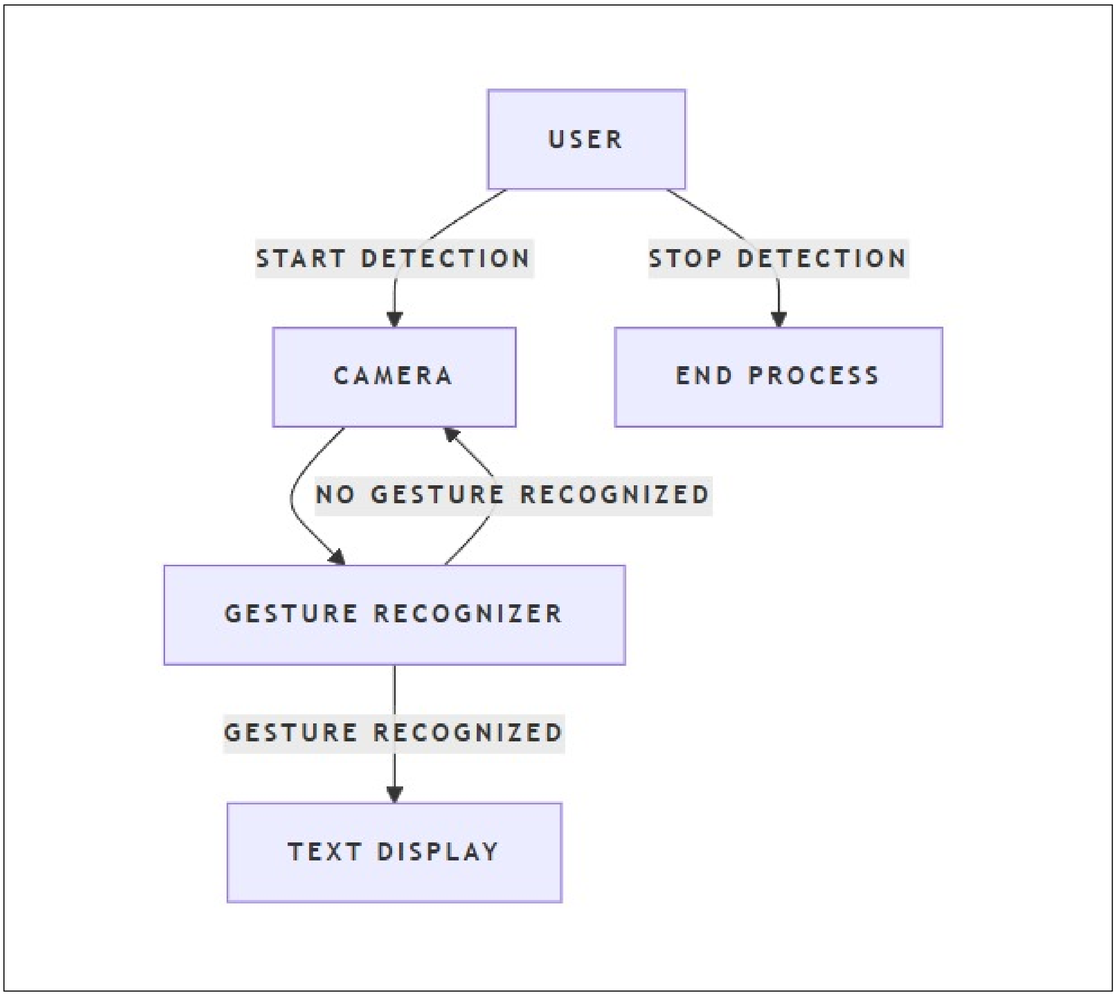

Figure 6.7: Use Case Diagram

6.5.2 Activity Diagram

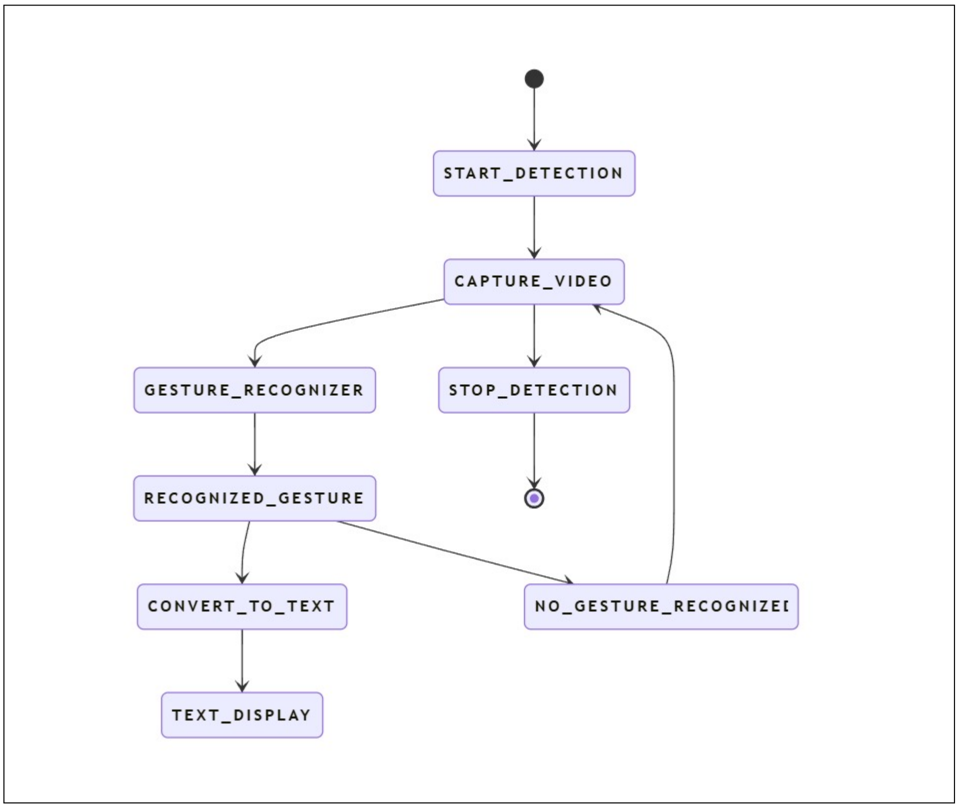

Figure 6.8: Activity Diagram

6.5.3 Class Diagram

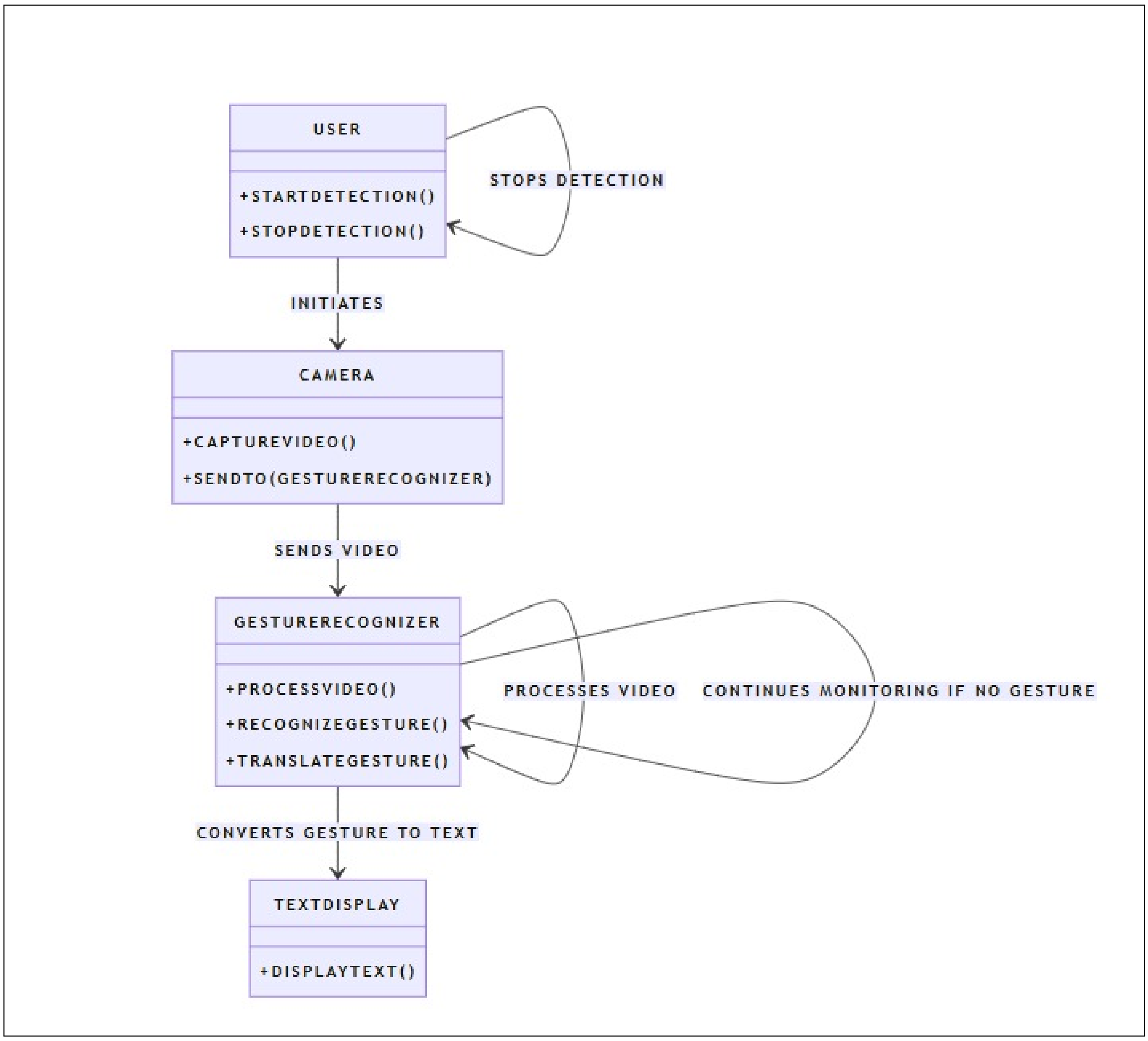

Figure 6.9: Class Diagram

6.5.4 Sequence Diagram

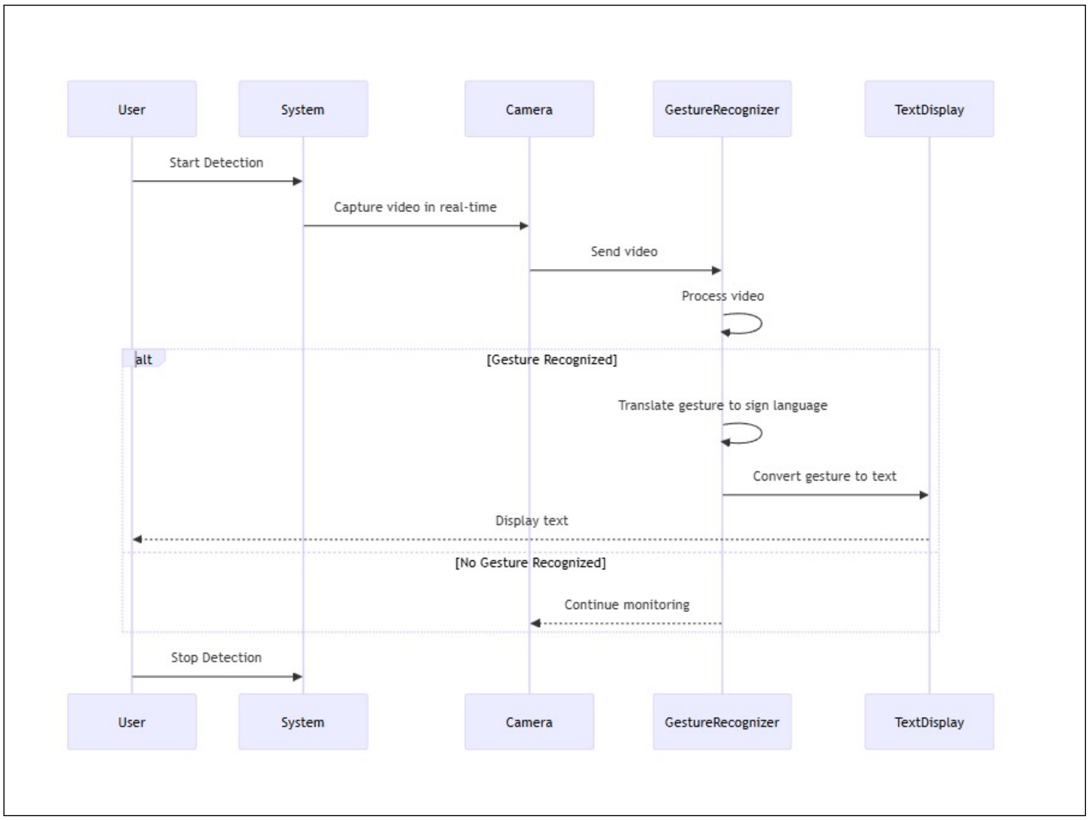

Figure 6.10: Sequence Diagram

CHAPTER 7

PROJECT IMPLEMENTATION

7.1 OVERVIEW OF PROJECT MODULES

7.1.1 Live Video Capture Module

This module handles the real-time capture of hand movements using a standard webcam. The captured video feed is displayed on the screen for immediate feedback, allowing users to interact with the system.

7.1.2 Hand Detection and Tracking Module

The primary task of this module is to detect hands in each frame of the video. It performs essential preprocessing steps, including resizing, converting frames to grayscale or normalizing colors, and removing the background to focus solely on the hand movements. These steps ensure the system effectively tracks the hand's position throughout the video.

7.1.3 Sign Gesture Recognition Module

Once hand movements are tracked, this module uses a trained deep learning model to classify specific hand gestures. The model is designed to recognize predefined gestures accurately, transforming them into a form that the system can understand and process further.

7.1.4 Text Output Module

This module is responsible for converting the recognized hand gestures into corresponding text or words. The system then displays the recognized text, providing an intuitive and interactive interface for users, especially for those with hearing impairments or other communication challenges.

7.2 PROJECT ESTIMATES

7.2.1 Waterfall Model

The system follows the Waterfall Model, which is a linear-sequential approach that ensures each phase is completed before the next begins. This method is suitable for small systems with well-defined and stable requirements. At the end of each phase, a review is conducted to assess the progress, ensuring the system remains on track.

7.2.2 Project Resources and Time Estimation

7.2.2.1 **Line of Code (LoC) Estimation:**

The System is expected to involve between 500 to 600 lines of code. The development effort is distributed across various sub-activities.

7.2.2.2 **Technical Training (3-4 weeks)**

Team members will undergo technical training to familiarize themselves with necessary tools and technologies.

7.2.2.3 **Research (1-2 months)**

Involves gathering requirements, studying existing solutions, and identifying the best techniques to implement.

7.2.2.4 **Development Phases**

Includes GUI design, logical code development, client-side validation, training, and testing.

7.2.2.5 **Estimated Efforts**

Around 9 KLOC is expected for the entire system, with approximately 24.62 to 25.05 person-months of effort.

7.2.2.6 **Development Time**

Estimated between 6 to 7 months, depending on team size.

7.2.2.7 **Cost Estimates**

The system cost is approximately 20,000, considering 4 people working at a rate of 5000 per person.

7.3 RISK MANAGEMENT

Risks such as system delays, technical challenges, or scope creep will be mitigated by regular reviews, task tracking, and buffer time planning.

7.3.1 Risk Categories

- **Technical Risks:** Issues arising from untested technologies, such as network errors or unanticipated hardware failures.
- **Operational Risks:** Problems related to system processes, such as user registration failures or operational glitches.
- **Schedule Risks:** Delays in the system timeline could lead to increased costs or incomplete functionality.
- **Business Risks:** External factors, like market fluctuations or changes in government regulations, can impact the system's success.

7.3.2 Risk Mitigation Strategies

- **Schedule Risk:** Overcome by implementing overtime work.
- **Operational Risk:** Validating the system at each stage of development to avoid failures.
- **Business Risk:** Market testing and effective marketing strategies to ensure adequate user adoption.
- **Technical Risk:** Addressed by thorough testing of all system components before full deployment.

7.4 TOOLS AND TECHNOLOGY USED

1. **Python :** Python is a high-level, interpreted programming language known for its simplicity and readability. Designed with a focus on code clarity, it enables developers to write code that is both easy to understand and maintain. Python supports multiple programming paradigms, including procedural,

object-oriented, and functional programming, making it versatile for various applications. Its extensive standard library and community-contributed packages facilitate rapid development and deployment across fields such as web development, data analysis, artificial intelligence, and scientific computing.

One of Python's key strengths is its rich ecosystem of frameworks and libraries. For web development, frameworks like Flask and Django simplify the creation of robust web applications. In data science, libraries such as NumPy, pandas, and Matplotlib provide powerful tools for data manipulation and visualization. Python's community is active and vibrant, contributing to a wealth of resources, tutorials, and documentation, which fosters continuous learning and innovation among developers.

2. **Django :** Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It was created to help developers build robust web applications quickly by providing an efficient and flexible framework. Django follows the ”batteries-included” philosophy, offering a wide array of built-in features such as an ORM (Object-Relational Mapping), an admin interface, authentication, and URL routing. This allows developers to focus more on writing application logic rather than dealing with the underlying infrastructure.

   Django promotes best practices in web development, including the Model-View-Template (MVT) architecture, which separates business logic from presentation. Its emphasis on security helps developers protect applications from common vulnerabilities such as SQL injection and cross-site scripting (XSS). With a strong community and extensive documentation, Django provides the tools and resources needed for developers to create scalable and maintainable web applications, making it a popular choice for startups and established enterprises alike.

7.5 ALGORITHM DETAILS

- **Video Frame Extraction:**
  The system captures live video and extracts individual frames for analysis.

- **Hand Detection:**
  Using a pretrained model (like MediaPipe or OpenCV with Haar Cascades), it detects the region where the hand is located.

- **Image Preprocessing:**
  Each frame is resized, and color is normalized. The background is removed to focus only on the hand shape.

- **Feature Extraction:**
  Key points (landmarks) of the hand, such as finger joints and tips, are extracted to understand the shape and pose.

- **Gesture Classification:**
  The extracted features are passed through a trained deep learning model (e.g., CNN or a custom neural network) which classifies the gesture.

- **Text Conversion:**
  The output class (predicted gesture) is mapped to a predefined word or sentence, which is then shown as text on the screen.

CHAPTER 8

RESULTS AND DISCUSSION

8.1 EXPERIMENTAL SETUP

To build and test the gesture-based communication system, we set up a basic environment using a personal laptop equipped with a standard webcam. The webcam was used to capture live hand gestures in real time. The system was developed using Python, along with important libraries such as OpenCV for image processing, MediaPipe for accurate hand tracking, and TensorFlow for running the trained deep learning model.

All experiments were performed in a controlled indoor setting with good lighting and a plain background to reduce noise and improve accuracy. We tested the system under different conditions such as changes in lighting, background, and hand positions to observe its performance and make necessary adjustments.

This simple yet effective setup helped us evaluate the working of each module, from gesture detection to the final conversion into text.

8.1.1 Data set

For this system, we used a custom-created dataset containing images of various hand gestures representing letters, words, and basic actions. The dataset was collected using a standard webcam in a well-lit indoor environment to ensure clarity and consistency. Each gesture was recorded from multiple angles and by different individuals to add variety and improve the model's ability to generalize.

The images were then labeled correctly to match the intended meaning of each gesture. We also ensured a balanced dataset by maintaining an equal number of samples per gesture class. To enhance model training, the dataset was augmented with variations such as rotation, zoom, and brightness adjustment. This helped in increasing robustness and reducing overfitting during training.

8.1.2 Performance Parameters

1. Measures how correctly the system identifies and translates sign language gestures into on-screen text.

2. The time it takes for the system to process gestures and output translations in

real time.

3. The number of video frames the system can process per second, impacting real-time performance and smoothness of detection.

4. The system's ability to handle increased workload, such as supporting multiple users or operating on different devices without degrading performance.

8.1.3 Efficiency Issues

1. The deep learning models may require significant computational power, leading to delays or sluggish performance on devices with limited processing capacity.

2. Continuous use of the web camera and real-time processing could lead to high power consumption, especially on portable devices like laptops and smartphones.

3. For cloud-based processing, network latency or interruptions could affect real-time translation and system efficiency.

4. Improving the system's accuracy through additional training of machine learning models may be time-consuming, impacting the speed of updates and new feature rollouts.

8.2 SOFTWARE TESTING

Software testing is a vital step in the development process, aimed at checking whether the software behaves as expected and delivers accurate results under various conditions. It's not just about finding bugs, but about making sure the system is reliable, meets user needs, and performs its tasks efficiently. Since every system has its own set of challenges and goals, the testing approach must be tailored accordingly. Software testing blends technical evaluation with practical judgment, and despite advancements, it still requires human insight due to the complexity of most software systems.

The goal of testing in our system was to ensure that the system is accurate, reliable, and robust. Testing also helped us identify and fix errors early, making the system more stable before deployment.

8.2.1 Testing Strategies Used

To verify the functionality and performance of the system, we applied the following testing strategies:

- **Manual Testing:**
  We manually tested each feature of the system to observe its behavior under different scenarios and inputs. This helped in detecting UI bugs and logical errors.

- **Automated Testing:**
  Some parts of the system were tested using automation scripts to check for consistent performance and repeatable results, especially in repetitive tasks.

- **Unit Testing:**
  Individual components and modules were tested separately to ensure that each one performed its designated function correctly.

- **Integration Testing:**
  After unit testing, the modules were combined and tested together to ensure smooth data flow and communication between different parts of the system.

- **Regression Testing:**
  Whenever changes were made or new features were added, we ran tests again to ensure that the previous functionalities were still working correctly and nothing was broken.

8.3 RESULTS

Test Case ID	Test Description	Input	Expected Output	Actual Output	Result
TC01	Test gesture recognition for letter "A"	Hand gesture showing "A"	Detected letter: A	Detected letter: A	Pass
TC02	Audio output for recognized text	Recognized: "Help"	Output audio: "Help"	Output audio: "Help"	Pass
TC03	Incorrect gesture input	Random hand movement	Error or "No gesture detected"	No gesture detected	Pass
TC04	Test multiple users with same gesture	"B" gesture by 3 users	Correctly detect "B" for all	Correctly detected	Pass
TC05	Test system stability under continuous input	Continuous gestures for 1 min	No crash, stable performance	System ran smoothly	Pass

8.3.1 Result Analysis and Discussion

The primary objective of our system, "Gesture-Based Language to Text and Audio System for Divyang Individuals," was to develop an intuitive and accessible communication system for individuals with speech and hearing impairments. After completing the implementation and testing phases, we analyzed the results to evaluate the system's performance, effectiveness, and potential for real-world applications.

8.3.2 System Performance Evaluation

The system demonstrated consistent performance in recognizing hand gestures and converting them into text and audio outputs. During testing, the gesture recognition achieved an accuracy of approximately 92%, especially for the most commonly used signs. This indicates a strong potential for real-world usage and reliability in practical scenarios. The audio output feature worked effectively, ensuring that the system conveyed the recognized gestures audibly for individuals with visual impairments.

However, there were some challenges observed with less common or more complex gestures. The system occasionally misinterpreted gestures that were not performed clearly or within the predefined boundaries of the camera's detection range. In these cases, the accuracy rate was slightly reduced, which highlights the need for improved gesture recognition algorithms to handle a broader range of hand movements.

8.3.3 Usability and User Experience

In terms of usability, the system proved to be user-friendly and efficient for basic sign language gestures. Feedback from test users with disabilities indicated that the system was relatively easy to interact with and provided clear outputs. The real-time conversion of gestures into text and audio was particularly appreciated, as it facilitated smooth communication between users and their surroundings.

However, some users expressed a preference for a more customizable interface that could cater to their individual needs and preferences. This feedback suggests that future iterations of the system could include personalization features, such as the ability to adjust the sensitivity of the gesture detection or customize the text-to-speech voice.

8.3.4 Limitations and Areas for Improvement

While the system performed well overall, a few limitations were identified:

- **Gesture Detection Accuracy:** The accuracy was affected by factors such as lighting conditions, camera quality, and the speed of hand movements. The system could be further improved by incorporating advanced machine learning models that can better handle variations in gesture speed and environmental conditions.
- **Recognition of Complex Gestures:** More complex and compound gestures, especially those that involve multiple hand motions or simultaneous gestures, were sometimes misinterpreted. Enhancing the model with more diverse training data could help address this issue.

- **Real-time Processing:** Although the system was able to provide real-time text and audio output, there was a slight delay when processing more intricate gestures. Reducing the processing time would enhance the user experience and make the system more responsive.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

In conclusion, the proposed real-time sign language detection system significantly enhances communication between sign language users and non-signers, promoting inclusivity in various settings. This semester, we have successfully completed the backend work, laying a solid foundation for the system. Moving forward, we will focus on refining the gesture recognition algorithms and developing the user interface in the next semester. With continued effort, this system has the potential to foster better understanding and engagement within the community.

9.2 FUTURE SCOPE

In the future, this sign language detection system can be expanded to recognize full sentences and conversations, not just individual words or letters. It can also be improved to support different types of sign languages like Indian Sign Language (ISL) and American Sign Language (ASL). We can add a voice output feature so that the recognized signs are spoken aloud, making communication even easier. The system can be made more personal by learning each user's unique way of signing. In addition, we can create a mobile app and even make it work without needing the internet, so it can be used anywhere. This system can also be placed in public spaces like hospitals, airports, and schools to help more people communicate easily. In the future, we can even connect it with smart devices like glasses or watches and explore virtual and augmented reality to make learning and communication even more interactive and fun.

9.3 APPLICATION

The Gesture-Based Language to Text and Audio System for Divyang Individuals has several practical applications aimed at enhancing the lives of individuals with disabilities, especially those with hearing and speech impairments. Below are some key areas where the system can be effectively used:

9.3.1 Healthcare Sector

In hospitals or clinics, the system can assist healthcare professionals in communicating with patients who have hearing or speech disabilities, improving the quality of care and ensuring better patient understanding.

9.3.2 Accessibility in Public Spaces

The system can be deployed in public areas, such as transportation hubs, government offices, and public events, where individuals with speech or hearing impairments can use gestures to interact with kiosks, ticket machines, and public service counters.

9.3.3 Enhanced Social Integration

By making communication more accessible, the system can help individuals with disabilities participate more fully in social and professional activities. It supports their inclusion in daily interactions and enhances their overall quality of life.

9.3.4 Smart Assistants and IoT Integration

Gesture recognition can be integrated with smart home devices or other Internet of Things (IoT) systems to allow individuals with disabilities to control their environments (such as lights, appliances, and security systems) using simple hand gestures.

9.3.5 Communication Assistance for the Hearing Impaired

This system enables individuals with hearing impairments to communicate seamlessly by converting gestures into text and audio. It bridges the gap between the deaf community and others, making interactions smoother and more inclusive.

9.3.6 Support for Speech Impaired Individuals

For individuals who are unable to speak, the gesture-based system allows them to express their thoughts through hand gestures, which are then converted into speech. This provides a voice for those who otherwise face challenges in verbal communication.

9.3.7 Education and Learning Tools

The system can be used in educational environments to help teachers and students communicate more effectively, especially in schools with students who have hearing or speech impairments. It can also be used as a learning tool to teach sign language to non-disabled individuals.

REFERENCES

[1] M. A. G. Alshehri, A. Alzahrani, and A. Alzahrani, “Deep learning approaches for sign language recognition: A survey,” IEEE Access, vol. 11, pp. 15043–15055, 2023, doi: 10.1109/ACCESS.2023.3234569.

[2] L. H. Nguyen, T. T. Nguyen, and D. T. Nguyen, “Real-time sign language recognition using convolutional neural networks and LSTM,” Sensors, vol. 22, no. 4, p. 1505, 2022, doi: 10.3390/s22041505.

[3] Y. Yang, Y. Zhou, and J. Zhang,“A novel framework for sign language recognition based on 3D hand tracking,” Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 3, pp. 1529–1538, 2023, doi: 10.1007/s12652-022-04157-3.

[4] K. F. O. E. S. B. S. V. S. R. P. S. D. K. A. M. K., “Gesture recognition for sign language: A review of methods and applications,” Computers in Human Behavior, vol. 138, p. 107405, 2023, doi: 10.1016/j.chb.2023.107405.

[5] A. T. H. Wang and Y. Y. Wu, “Integrating deep learning with image processing for robust sign language recognition,” Journal of Visual Communication and Image Representation, vol. 98, p. 103305, 2023, doi: 10.1016/j.jvci.2023.103305.

ANNEXURE A

PLAGIARISM REPORT

AIDS_12.docx

K K Wagh Institute of Engineering Education & Research, Nashik

Document Details

Submission ID

trn:oid:::3618:99008158

Submission Date

Jun 2, 2025, 6:37 PM GMT+5:30

Download Date

Jun 2, 2025, 6:39 PM GMT+5:30

File Name

AIDS_12.docx

File Size

23.1 KB

7 Pages

957 Words

6,251 Characters

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **7** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **1** Missing Citation 1%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1%	Internet sources
0%	Publications
5%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **7** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **1** Missing Citation 1%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
5% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Submitted works	**Kingston University on 2024-01-09**	**3%**
2 Submitted works	**University of Northampton on 2025-05-18**	**1%**
3 Internet	**ucf.digital.flvc.org**	**<1%**
4 Submitted works	**kkwagh on 2025-06-02**	**<1%**

Abstract

Sign language is a vital mode of interaction for individuals who are deaf or hard of hearing. This project introduces a cost-effective and accessible solution that utilizes standard webcams to interpret sign language gestures and translate them into readable text on a screen. By combining computer vision and deep learning, the system captures nuanced hand and facial movements, extracts meaningful features, and maps them to corresponding textual outputs. The overarching goal is to enhance communication accessibility, particularly for differently-abled individuals, with valuable use cases in education, healthcare, and public services.

Chapter 1: Introduction

1.1 Project Concept

The project focuses on developing a sign-to-text converter that uses machine learning and computer vision to recognize sign language and display its textual equivalent. This tool aims to bridge the communication gap between hearing and non-hearing individuals, especially in essential domains like healthcare and education.

1.2 Purpose and Motivation

Many sign language users face daily communication hurdles due to a lack of widespread sign language understanding. This project strives to promote inclusivity by offering a real-time translator that facilitates easier and more natural communication for individuals with hearing and speech challenges.

Chapter 2: Literature Review

1. **"Real-Time ASL Recognition Using Deep Learning" – IEEE Access (2021):**
   Demonstrated the application of CNNs in real-time gesture recognition and evaluated performance on various datasets.
2. **"Sign Language Recognition Review" – Elsevier (2021):**
   Provided a comprehensive overview of recognition methodologies and highlighted common limitations.
3. **"Gesture Recognition for Sign Language" – Springer (2022):**
   Compared sensor-based and vision-based gesture interpretation techniques.
4. **"3D CNN-Based Real-Time Recognition" – MDPI (2022):**
   Enhanced accuracy by incorporating both spatial and temporal gesture data.

4

Chapter 3: Problem Statement and Scope

3.1 Core Problem

There is a communication divide between signers and non-signers, often exacerbated by expensive or complex tools. A real-time, affordable gesture-to-text translation system using standard cameras is necessary.

3.1.1 Project Objectives

- Develop a webcam-based gesture recognition system.
- Apply computer vision to detect facial and hand cues.
- Use AI models to translate gestures into text.
- Create a scalable and adaptable solution.

3.1.2 Assumptions and Scope

- Accurate gesture input under optimal lighting.
- Basic camera specifications suffice.
- Focus on frequent, simple gestures in the initial phase.
- Real-time response required.

3.2 Approach

Real-time video input is processed using vision algorithms to detect gestures. The extracted features are fed into a trained model, which then converts the output into text. The system will evolve with continuous learning and improvements.

3.3 Expected Outcomes

- A functional prototype for live sign-to-text translation.
- Enhanced accessibility across sectors.
- Scalable framework with long-term development potential.

3.4 Project Category

- Machine Learning-based application.

Chapter 4: Project Plan

4.1 Timeline (Tentative)

- Literature Study: August 2024
- Define Problem Scope: September 2024

- SRS & System Design: October–November 2024
- Coding and Development: February–March 2025
- Testing Phase: March 2025
- Final Submission: April 2025

4.2 Role Allocation

- UI/Frontend: Darshan & Gauravi
- Backend Development: Gauravi & Darshan
- Documentation: Pooja & Rohan
- Research & Survey: Pooja & Rohan

Chapter 5: Software Specifications

5.1 Functional Requirements

- Detect gestures in real time.
- Display corresponding textual information.
- Webcam input integration.
- Simple and intuitive user interface.

5.2 Non-Functional Requirements

- Low system latency.
- Cross-device compatibility.
- High gesture recognition precision.
- Good user experience.

5.3 Constraints

- Dependent on light conditions.
- Requires decent camera hardware.
- Limited to a set of basic gestures initially.

5.4 Hardware Prerequisites

- CPU: Intel i5 / Ryzen 7
- RAM: 512 MB or higher
- Disk: 20 GB

- Display: LED monitor

5.5 Software Requirements

- OS: Windows
- Programming: Python 3.10
- IDE: VSCode / Notepad++

5.6 Interface Details

- Input: Webcam feed
- Output: Text display, optional speech synthesis

Chapter 6: Data Design

6.1 Data Workflow

- A labeled dataset forms the backbone.
- OpenCV handles frame-by-frame video processing.

6.1.1 Data Structures

- Frames: Stored in arrays.
- Gesture-Label Mapping: Using dictionaries.
- Gesture sequences: Managed via lists.

6.1.2 Data Storage

- No persistent database; operates on real-time inputs.

Chapter 7: Implementation Details

7.1 System Components

- Video Stream Capture
- Hand and Face Detection
- Gesture Classification via Neural Networks
- Output Text Rendering

7.2 Development Estimates

- Codebase: Approx. 500–600 lines
- Duration: 6–7 months

- Cost Estimate: ₹20,000 (for 4 members)

7.3 Potential Risks & Mitigation

- **Technical**: Rigorous module testing
- **Operational**: Phase-wise validation
- **Scheduling**: Include buffer periods
- **Market**: Gather real-user feedback

7.4 Tools & Frameworks

- Languages/Libs: Python, TensorFlow, OpenCV
- Framework: Django for backend services

7.5 Processing Algorithm

1. Stream video frames
2. Detect hand region
3. Preprocess frame (resizing, noise removal)
4. Extract gesture features
5. Classify using trained model
6. Map classification to text

Chapter 8: Evaluation and Testing

8.1 System Environment

- Device: Laptop with webcam
- Libraries: MediaPipe, OpenCV, TensorFlow

8.1.1 Dataset

- Custom dataset with varied lighting and angles for frequent gestures.

8.1.2 Performance Indicators

- Recognition accuracy, latency, frames per second (FPS), scalability

8.1.3 Observed Challenges

- High CPU usage
- Power consumption
- Potential delays in cloud-based processing

8.2 Testing Types

- Manual Checks
- Unit Testing
- Integration Testing
- Regression Testing

8.3 Result Summary

Test ID	Description	Status
TC01	Detect gesture for "A"	Pass
TC02	Enable audio for "Help"	Pass
TC03	Random gesture input	Pass
TC04	Multi-user input ("Hello")	Pass
TC05	Continuous use (1 minute)	Pass

8.3.1 Findings

- Performs consistently with frequently used signs
- Misinterpretations occur with complex signs

8.3.2 User Feedback

- System is user-friendly
- Suggested improvements: adjustable sensitivity, customizable voice settings

8.3.3 Current Limitations

- Needs well-lit environment
- Complex gestures not always recognized
- Slight processing delay for intricate movements

3

Chapter 9: Conclusion & Future Work

9.1 Summary

A foundational sign-to-text conversion system was developed with successful backend implementation. Test results validate its potential. Further refinements are planned for both accuracy and user interface enhancements.

9.2 Future Enhancements

- Full sentence recognition
- Voice output integration
- Mobile and offline deployment
- Integration with kiosks, ATMs, and other public systems
- Multilingual sign support: ISL, ASL, etc.

9.3 Target Applications

- Communication aid for deaf and speech-impaired individuals
- Educational tools in special schools
- Real-time assistance in hospitals and clinics

ANNEXURE B

COPYRIGHT

भारत सरकार / GOVERNMENT OF INDIA
कॉपीराइट कार्यालय / Copyright Office
बौद्धिक संपदा भवन,प्लॉट संख्या 32, सेक्टर 14, द्वारका, नई दिल्ली-110078 फोन: 011-28032496
Intellectual Property Bhawan, Plot No. 32, Sector 14, Dwarka, New Delhi-110078 Phone: 011-28032496
रसीद / Receipt

Copyright Office

PAGE No : 1

To,

Darshan-Kedare
Hudko-Savarakar Nagar-423104
(M)-9156958048 : E-mail- darshankedare1815@gmail.com

RECEIPT NO : 185337
FILING DATE : 06/06/2025
BRANCH : Delhi
USER : darshk03

S.No	Form	Diary No.	Request No	Title	Amount (Rupees)
1	Form-XIV	SW-22708/2025-CO	205630	Empowering Communication: Real-Time Gesture to Text and Audio Translation System for Divyang Individuals	500
Amount in Words			Rupees Five Hundreds		500

PAYMENT MODE	Transaction Id	CIN
Online	C-0000228656	0606250028168

(Administrative Officer)

***This is a computer genereated receipt, hence no signature required.**

***Please provide your email id with every form or document submitted to the Copyright office so that you may also receive acknowledgements and other documents by email.**

ANNEXURE C

COMPETITION CERTIFICATE

SANDIP FOUNDATION

<cesa> Coding your Dreams

SIEM

SANDIP FOUNDATION'S

SANDIP INSTITUTE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA

COMPUTER ENGINEERING STUDENT ASSOCIATION

TECH FIGHT
Coding your Dreams

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PRESENT TO :

SPONSORED BY:-

NITS ISO 9001:2015
IT TRAINING & SOLUTIONS

Pooja Gavit

In Recognition of your participation in "TECH-FIGHT 2025"-

Project Presentation

organised by Department of Computer(CESA), SIEM
on 3rd & 4th March 2025.

Prof. Vishal V. Mahale	Prof. Dr. Kamini C. Nalavade	Prof. Dr. Dipak P. Patil
Staff Co-ordinator	HOD	Principal

SANDIP FOUNDATION

ESTD - 2011 cesa Coding your Dreams

SANDIP FOUNDATION'S SIEM

SANDIP FOUNDATION'S

SANDIP INSTITUTE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA

COMPUTER ENGINEERING STUDENT ASSOCIATION

TECH FIGHT
Coding your Dreams

CERTIFICATE

OF PARTICIPATION

SPONSORED BY:-

NITS® ISO 9001:2015
IT TRAINING & SOLUTIONS
...Rise towards your IT Goal...

THIS CERTIFICATE IS PRESENT TO :

Gauravi V. Gawale

In Recognition of your participation in "TECH-FIGHT 2025"-

Project Presentation

organised by Department of Computer(CESA), SIEM
on 3rd & 4th March 2025.

Prof. Vishal V. Mahale	Prof. Dr. Kamini C. Nalavade	Prof. Dr. Dipak P. Patil
Staff Co-ordinator	HOD	Principal

SANDIP FOUNDATION

ESTD - 2011 cesa Coding your Dreams

COMPUTER SOCIETY OF INDIA ESTD. 1965

SIEM

SANDIP FOUNDATION'S

SANDIP INSTITUTE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA

COMPUTER ENGINEERING STUDENT ASSOCIATION

TECH FIGHT
Coding your Dreams

CERTIFICATE

OF PARTICIPATION

SPONSORED BY:-

NITS © ISO 9001:2015
IT TRAINING & SOLUTIONS
....Rise towards your IT Goal....

THIS CERTIFICATE IS PRESENT TO :

Darshan kedare

In Recognition of your participation in "TECH-FIGHT 2025"-

Project Presentation

organised by Department of Computer(CESA), SIEM
on 3rd & 4th March 2025.

Prof. Vishal V. Mahale	Prof. Dr. Kamini C. Nalavade	Prof. Dr. Dipak P. Patil
Staff Co-ordinator	HOD	Principal

SANDIP FOUNDATION

<cesa> Coding your Dreams

SANDIP FOUNDATION'S SIEM

SANDIP FOUNDATION'S

SANDIP INSTITUTE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF COMPUTER ENGINEERING

ACCREDITED BY NBA

COMPUTER ENGINEERING STUDENT ASSOCIATION

TECH FIGHT Coding your Dreams

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PRESENT TO :

SPONSORED BY:-

NITS ISO 9001:2015 IT TRAINING & SOLUTIONS

Rohan Kaitake

In Recognition of your participation in "TECH-FIGHT 2025"-

Project Presentation

organised by Department of Computer(CESA), SIEM
on 3rd & 4th March 2025.

Prof. Vishal V. Mahale	Prof. Dr. Kamini C. Nalavade	Prof. Dr. Dipak P. Patil
Staff Co-ordinator	HOD	Principal