UrduGest: Urdu Sign Language Recognition

Ahmad Hassan Raja (BSAI-006)

Taha Farooq (BSAI-002)

Muhammad Qasim Khan (BSAI-028)

Supervised By

Respected Dr. **Moiz Ullah Ghouri**

Submitted for the partial fulfillment of BS Artificial Intelligence degree to the Faculty of Engineering & Computing

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF MODERN LANGUAGES

ABSTRACT

This project is focused on building a real-time recognition system for Urdu Sign Language to support communication for people who are deaf or hard of hearing. While many technologies already exist for popular sign languages like American Sign Language, Urdu Sign Language is still largely ignored in modern tools. To solve this issue, we developed a system that can recognize hand gestures for Urdu using a regular camera and simple machine learning techniques.

We first created our own datasets by recording videos of hand gestures using a webcam. These datasets included three types of signs: the 37 letters of the Urdu alphabet, the numbers 0 to 9, and 7 commonly used action gestures. To extract useful information from each gesture, we used a library called Mediapipe that helps identify the exact positions of hand landmarks, like finger joints and hand tips, from the video. This data was stored and used to train a machine learning model known as the Random Forest Classifier. The model learned how to tell different signs apart and reached an accuracy of around 85%.

The system works in real time. When a person performs a sign in front of the camera, the system detects it, shows a box around the hand, and displays the identified Urdu character or action directly on the screen using an Urdu font. We also built a complete graphical interface using a framework called Streamlit, so users can interact with the system easily. It has a login and registration system, and allows users to choose between recognizing alphabets, numbers, or actions. There's also an advanced section where users can combine gestures to form full words or sentences. Once a sentence is formed, the system can also read it out loud using a text-to-speech tool and translate the Urdu sentence into English using an online translation service. These features make the system not only useful for sign recognition but also for everyday communication and learning.

This project is a step forward in making technology more inclusive for the Urdu-speaking deaf community. It combines live camera input, gesture recognition, text display, translation, and voice—all in one system. In the future, we plan to improve the system's speed and accuracy, and extend support to other regional or international sign languages.

CERTIFICATE

Dated: _____

Final Approval

It is certified that project report titled "**UrduGest: Urdu Sign Language Recognition**" submitted by **Ahmad Hassan Raja**, **Taha Farooq**, and **Muhammad Qasim Khan**, for the partial fulfillment of the requirement of "**Bachelor's Degree in Artificial Intelligence**" is approved.

COMMITTEE

Dr. Noman Malik
Dean Engineering & CS Signature: _____

Dr. Fazli Subhan
HoD Engineering Signature: _____

Malik Muhammad Shahid Ali
Head Project Committee Signature: _____

Dr. Moiz Ullah Ghauri
Supervisor Signature: _____

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine/original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade and it may result in withdrawal of our Bachelor's degree.

Group Members	**Signature**
1. Ahmad Hassan Raja	_____
2. Taha Farooq	_____
3. Muhammad Qasim Khan	_____

PLAIGRISM CERTIFICATE

This is to certify that the project entitled "**UrduGest: Urdu Sign Language Recognition**", is being submitted here for the award of the "**Degree of Bachelor**" in "**Artificial Intelligence**". This is the result of the original work by **Ahmad Hassan Raja**, **Taha Farooq**, and **Muhammad Qasim Khan**, under my supervision and guidance. The work embodied in this project has not been done earlier for the basis of the award of any degree or compatible certificate or similar title of this for any other diploma/examining body or university to the best of my knowledge and belief.

Turnitin Originality Report

Processed on 0-May-2025 00:00 PKT

ID: XXXXXXXX

Word Count: 9999

Similarity Index

X%

Similarity by Source

Internet Sources: X%

Publications: X%

Student Papers: X%

Date: 20/05/2020

Dr. Moiz Ullah Ghauri

Table of Contents

List of Figures

List of Tables

CHAPTER 1:

INTRODUCTION

1.1 Introduction

Language is a fundamental tool for communication, bridging the gap between individuals and societies. However, for millions of people with hearing or speech impairments, traditional communication methods may not be viable. Sign language serves as an essential medium for these individuals to express themselves and engage with others. Urdu, being the national language of Pakistan, has its own unique sign language system, known as Urdu Sign Language (USL). Despite its importance, USL recognition systems are underdeveloped, limiting the accessibility and inclusivity of deaf and mute individuals in educational, professional, and social contexts.

The advent of artificial intelligence (AI) and machine learning (ML) has revolutionized various fields, offering innovative solutions to complex problems. Leveraging these technologies, **UrduGest** aims to develop an automated Urdu Sign Language recognition system. This system utilizes real-time gesture detection and classification to enable effective communication for Urdu-speaking individuals with hearing impairments. Through this project, we address a critical gap in the domain of accessible technology and strive to promote inclusivity in communication.

1.2 Motivation

Pakistan is home to approximately 10 million individuals facing hearing and speech impairments [1]. These challenges often lead to social isolation and limited opportunities in education, employment, and personal growth. Existing sign language recognition systems predominantly focus on widely spoken languages, leaving Urdu speakers at a disadvantage.

This disparity inspired the development of **UrduGest**, a project aimed at creating an automated system to recognize Urdu Sign Language gestures. By providing a real-time solution, this system aspires to empower individuals with hearing impairments, enabling them to communicate effectively and participate actively in society. Our motivation stems from the potential of this project to bridge communication gaps and foster inclusivity for the marginalized deaf and mute community in Pakistan.

1.3 Problem Statement

The lack of robust Urdu Sign Language recognition systems creates significant barriers for individuals with hearing and speech impairments in Pakistan. Current methods rely heavily

on human interpreters, which are scarce, expensive, and prone to errors. Automated systems developed for other sign languages often fail to capture the unique variations of Urdu Sign Language, resulting in misinterpretation and ineffective communication.

This gap necessitates the development of a reliable, real-time Urdu Sign Language recognition system. Such a system should accurately interpret hand gestures and translate them into Urdu text or speech, while also offering translation capabilities into English for broader accessibility. Addressing this issue is crucial for ensuring equal opportunities and effective communication for the deaf and mute population.

1.4 Goals and Objectives

The main aim of the UrduGest project is to build a smart and real-time Urdu Sign Language recognition system to help individuals with hearing and speaking difficulties communicate more easily. Recognizing the lack of proper tools for Urdu Sign Language, this project focuses on using camera-based input to detect and interpret hand gestures. The system was trained on a custom-built dataset of Urdu alphabet signs, numbers, and common actions. By using a Random Forest Classifier along with landmark detection from the Mediapipe library, the model is able to identify hand gestures with high accuracy.

One of the major goals of this project was real-time gesture recognition. The system uses a webcam to continuously monitor hand movements and immediately responds by displaying the corresponding Urdu character or action on the screen. It also allows users to form complete words or sentences by combining letters through the interface buttons. The recognition output is shown using an Urdu font, making it visually native and easy to read. A user-friendly interface was designed using Streamlit, offering login and registration, model selection (alphabets, numbers, common actions), adjustable confidence levels, and feedback options, so users with any background can use the system without difficulty.

Beyond recognition, the system includes additional helpful features. Once a word or sentence is formed using signs, the system can convert the Urdu text into speech to help users communicate more naturally. It also provides a translation feature that converts the Urdu sentence into English using a Hugging Face translation model. These combined features make the system useful not only for communication but also for learning, teaching, and bridging the gap between the deaf community and the rest of society.

1.5 Scope of the Study

This project is designed for individuals who have hearing or speech impairments and rely on Urdu Sign Language for communication. It aims to provide them with a helpful tool that improves their ability to express themselves in different environments.

The system uses technologies like computer vision and machine learning to recognize hand gestures in real time. It also includes features like converting Urdu text to speech and translating Urdu into English, making the system more useful in daily life. The complete setup includes modules for gesture recognition, voice output, and translation. These features are especially useful in schools, workplaces, and public spaces where communication is often a challenge for the deaf community.

While the system currently focuses only on Urdu Sign Language, its performance depends on the accuracy of the hand gestures and the quality of the training data. Still, the project offers a strong step toward greater accessibility and aims to make communication easier and more inclusive for everyone.

1.6 Process Model

The "UrduGest" project utilizes the Iterative Development Model, a methodology well-suited for projects requiring constant refinement and adaptability. Given the project's focus on implementing advanced AI and machine learning techniques, the iterative model offers several advantages. It enables the incremental delivery of components, such as gesture recognition and text-to-speech, allowing for continuous testing and improvement based on feedback. This approach facilitates the easier incorporation of new functionalities, such as support for numbers and Urdu words, while minimizing risks associated with large-scale development. This iterative approach ensures the system evolves efficiently and effectively.

1.7 Nature of the Project

UrduGest is primarily an AI and Machine Learning-based application with a focus on real-time computer vision and natural language processing (NLP). The project incorporates AI/ML to enable gesture recognition using Random Forest model, leverages Mediapipe and OpenCV for hand pose estimation, and utilizes Streamlit for building a user-friendly web application. This multi-disciplinary approach aims to improve accessibility for the hearing and speech impaired, resulting in an innovative and user-centric solution.

1.8 Overview/Organization of the Report

This progress report is organized into the following chapters:

- **Chapter 1: Introduction**

  Covers the background, motivation, problem statement, goals, objectives, scope, and methodology.

- **Chapter 2: Background and Existing Work**

  Discusses the application domain, existing systems, and their limitations, helping to identify the need for the proposed solution.

- **Chapter 3: Requirements Specification**

  Details both functional and non-functional requirements. It includes use case models and feasibility studies to validate the proposed system's practicality.

- **Chapter 4: System Modeling**

  Describes the design and structure of the system through interface mockups, architecture diagrams, and entity-relationship models that guide the implementation.

- **Chapter 5: Implementation**

  Provides a comprehensive explanation of how the system was developed, including the tools, libraries, and machine learning models used. It discusses each module and the step-by-step development process.

- **Chapter 6: Testing, Results, and Validation**

  Explains how different parts of the system were tested using structured testing methods. It presents test cases, results, and performance evaluations to assess system accuracy and reliability.

- **Chapter 7: Conclusion and Future Work**

  Summarizes the project outcomes and compares them with the original goals. It highlights key achievements, discusses any limitations, and presents suggestions for extending the system, such as supporting other sign languages or improving accuracy.

CHAPTER 2:

BACKGROUND AND EXISTING SYSTEM

2.1 Introduction

The field of sign language recognition has gained significant attention over the years, particularly with advancements in artificial intelligence (AI) and machine learning (ML) technologies. These innovations have paved the way for automated systems that can recognize sign language gestures in real-time, offering a promising solution to communication barriers for individuals with hearing impairments. Sign languages, such as American Sign Language (ASL) [2] and British Sign Language (BSL), have been the focus of several research projects, but the lack of recognition systems tailored to regional languages like Urdu remains a gap in current solutions.

Urdu, the national language of Pakistan, has its own sign language system, known as **Urdu Sign Language (USL)**. However, there is a lack of effective and widespread tools for recognizing this language. The need for an automated Urdu sign language recognition system is critical, as it would significantly enhance communication and inclusion for millions of individuals with hearing impairments in Pakistan. This chapter explores the key concepts, constructs, and research related to sign language recognition, with a focus on Urdu Sign Language.

2.2 Explanation of Important Constructs of the Application Domain

Sign language recognition (SLR) is a subfield of computer vision and natural language processing (NLP), aimed at enabling machines to interpret human gestures as language. In the case of Urdu sign language, the system needs to recognize and interpret hand gestures representing Urdu letters, numbers, and commonly used phrases. The main constructs in this domain include:

- **Hand Gesture Recognition:**

  Hand gesture recognition is a vital component of sign language recognition systems. It involves identifying hand movements, and positions to determine the meaning of a gesture. The challenge lies in differentiating between various gestures, accounting for

factors such as hand orientation, finger positioning, and speed of movement. Computer Vision technologies like OpenCV and Mediapipe have made significant advancements in real-time hand gesture recognition by detecting and tracking key landmarks on the hands.

- **Machine Learning for Gesture Classification:**

Machine learning plays a crucial role in building accurate sign language recognition systems. After preprocessing hand gesture data, machine learning algorithms are used to classify gestures into specific categories (in this case, Urdu alphabets, numbers or common actions). Machine learning algorithm used is Random Forest Classifier. The choice of algorithm depends on the complexity of the gestures and the accuracy requirements of the system.

- **Feature Extraction:**

Feature extraction is the process of identifying relevant patterns or landmarks from raw input visual data, to represent the gesture in a more useful form for machine learning models. In the case of sign language recognition, this might involve extracting data points related to hand position, orientation and fingers movement. Mediapipe provides hand landmark detection models that extract key features from the hand to support accurate gesture recognition.

- **Real-Time Processing:**

Real-time sign language recognition involves capturing, processing, and interpreting gestures. It presents unique challenges in terms of processing speed, model accuracy, and responsiveness. Technologies like TensorFlow, OpenCV allow for efficient real-time processing on devices with limited computational power.

- **Multimodal Communication:**

Sign language recognition systems often benefit from multimodal communication capabilities. These integrate additional features such as Text-to-Speech or Speech-to-Text. For Urdu sign language, converting recognized signs into spoken Urdu or written text can enhance communication, enabling non-sign language users to understand the

conversation. Integrating translation between languages, such as from Urdu sign language to English, can broaden the system's usability.

2.3 Existing Studies/Systems

Research on Urdu Sign Language (USL) recognition has been steadily growing, though it remains less extensive compared to research on sign languages for more widely spoken languages like American Sign Language (ASL) or British Sign Language (BSL).

Early research in USL recognition explored simpler approaches, often focusing on isolated gestures. These studies employed traditional machine learning techniques such as K-Nearest Neighbors (KNN), which classifies gestures based on their similarity to existing data points in the training set; Bag-of-Words (BoW), which represents gestures as a collection of visual features, similar to how text is represented as a collection of words; and Support Vector Machines (SVM), which aim to find an optimal hyperplane to separate different gesture classes.

More recent research has leveraged the power of deep learning, particularly Convolutional Neural Networks (CNNs), to achieve higher accuracy in USL recognition. CNNs have proven effective in extracting relevant features from image data, making them well-suited for analyzing hand gestures. Studies like "UrSL-CNN Approach to Urdu Sign Language Translation for Hearing-Impaired Individuals" [3] have demonstrated the effectiveness of CNNs for high-accuracy sign language classification. Some studies have explored hybrid approaches that combine traditional machine learning techniques with deep learning models. For example, "A computer vision-based system for recognition and classification of Urdu sign language dataset" [4] utilized multiple classifiers, including Random Forest and other techniques, to improve recognition performance.

Key research findings emphasize the importance of high-quality and diverse training data for achieving accurate USL recognition. Researchers have addressed challenges such as variations in signing style, lighting conditions, and background clutter through techniques such as data augmentation and robust feature extraction methods. Recent research has increasingly focused on developing real-time USL recognition systems for practical applications, such as real-time communication and assistive technologies.

2.4 Comparison of Existing Systems

The comparison of existing systems is given in Table 2.1. It highlights various sign language recognition solutions in terms of their techniques, and Language support.

Table 2. 1 Comaprison of Existing Systems

ef	Article/APP/System	Year	Features	Techniques	Limitations
[1]	UrSL-CNN Approach to Urdu Sign Language Translation for Hearing-Impaired Individuals	2024	High-accuracy, multilingual, image-based sign language classification.	CNN, SVM, Naïve Bayes, Random Forest	No Graphical User Interface
[2]	A computer vision-based system for recognition and classification of Urdu sign language dataset	2022	Multiple classifiers-based system for recognition of Urdu sign language gestures.	Bag of words, KNN, Pattern recognition, Random Forest	No Real-Time Sign Detection
[3]	ML Based Sign Language Recognition System	2021	Vision-based isolated hand gesture detection and recognition	KNN, Bayesian KNN	No support for Urdu Language
[4]	Urdu Sign Language Reorganization via Artificial Neural Networks	2021	Neural Networks based system for Urdu sign language recognition. Achieved high accuracy in recognizing	ANN, KNN, SVM	No Real-Time Sign Detection

			isolated Urdu alphabets.		

2.5 Summary

This chapter has provided a comprehensive overview of the key concepts and existing research in the field of sign language recognition. While significant advancements have been made in recognizing widely spoken sign languages, research on Urdu Sign Language recognition is relatively limited. This gap presents a critical opportunity for the "UrduGest" project to contribute to the field by developing a robust and accessible system for Pakistani deaf community. The insights gained from existing research will guide the development of the "UrduGest" system, addressing the challenges and limitations of current approaches.

CHAPTER 3:

REQUIREMENT SPECIFICATION

3.1 Introduction

This chapter outlines the detailed requirements for the "UrduGest" project, covering both functional and non-functional aspects. The requirements are defined with a strong focus on the needs and expectations of the target users, ensuring the system is accessible, usable, and effective for its intended purpose. Key considerations include user-centric design and technical feasibility, ensuring that the requirements can be realistically implemented within the scope of the project and with the available resources. These requirements will guide the development process and ensure that the final system meets the needs of the target users – individuals with hearing impairments. This chapter will provide a comprehensive understanding of the project's requirements, serving as a foundation for the subsequent design and development phases.

3.2 Interface Requirements

Interface requirements define the necessary interactions between different components of the system and with the external environment. For the "UrduGest" project, these requirements encompass both hardware and software interfaces.

3.2.1 Hardware Interface Requirements

- **Developer End**

  The development environment should include a suitable PC with at least an Intel Core i5 or i7 / AMD Ryzen 5 or 7 with sufficient processing power and at least 8GB 3200MHz memory to handle the demands of real-time video processing, hand pose estimation, and machine learning model training.

- **User End**

  The system is designed to be compatible with a wide range of devices including desktop PCs, laptops, and mobile devices, that has an integrated or dedicated camera sensor. This will allow user to detect sign language using their device.

3.2.2 Software Interface Requirements:

- **Operating System:** The system will be deployed as a web application, accessible through any modern web browser (Chrome, Opera, Edge, etc.) on various operating systems (Windows, macOS, Linux, iOS, Android).
- **Programming Languages:** The system will be developed using Python.
- **Libraries:** The system will utilize libraries such as OpenCV, MediaPipe, and Random Forest classifier for computer vision tasks, hand pose estimation, and machine learning model training, respectively.
- **User Interface:** The system will employ the Streamlit framework for creating a user-friendly and interactive graphical user interface (GUI). Streamlit allows the developer to design visually attractive and user-friendly GUI.

3.3 Functional Requirements

Without sacrificing data or user security, these functional requirements guarantee that the Urdu Sign Language Recognition System operates smoothly and provides accurate, real-time gesture detection.

3.3.1 Real-time Gesture Landmarking

The system must accurately and efficiently detect and track 21 hand landmarks as illustrated in Figure 3.1 in real-time using computer vision techniques. This includes accurate hand detection, ensuring reliable detection and tracking of the user's hands within the video stream. The system should be robust to variations in lighting conditions, hand position (e.g., distance from the camera, hand orientation), and background clutter. Furthermore, it must process video frames and detect hand landmarks with minimal latency to ensure a smooth and responsive user experience.

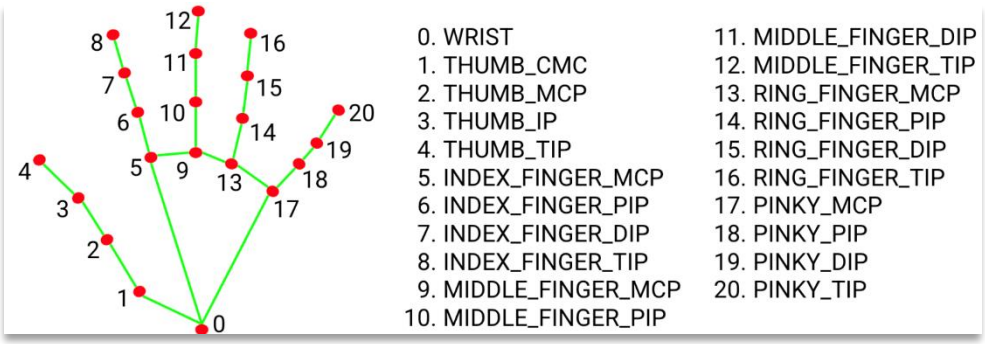

Figure 3. 1 Hand Landmarks

3.3.2 Urdu Alphabets Recognition

The system must accurately recognize and classify individual Urdu alphabet signs with high precision and recall. It should be able to handle variations in signing styles, including differences in hand shapes, speeds, and individual signing preferences. Furthermore, the system should be robust to noise and interference, such as background movement or slight hand tremors, that may affect gesture recognition accuracy.

3.3.3 Number Recognition

The system must accurately recognize and classify numerical signs (0-9) with high accuracy.

3.3.4 Common Actions Recognition

The system should be able to recognize and interpret a few commonly used actions and phrases by accurately identifying individual signs.

3.3.5 Words/Sentence Formation

The system should allow users to construct complete words or sentences by recognizing individual Urdu alphabet signs. Users can interact with buttons to add characters, insert spaces, save or clear the sentence, and quit recognition. The formed sentence should be displayed, converted to Urdu speech, and translated to English.

3.3.6 Result Display

The system must display the recognized sign/word in Urdu font, right above the hand performing sign. The display should be easy to read and understand for users with varying levels of technical expertise.

3.3.7 Urdu Font Support

The system must accurately render Urdu text using appropriate font such as Jameel Noori Nastaleeq that support the correct display of Urdu characters.

3.3.8 Text-to-Speech Conversion

The system should convert the recognized text into high-quality Urdu speech using a suitable text-to-speech model. The synthesized speech should be clear, natural-sounding, and easily understandable to the user.

3.3.9 English Translation (for words and sentences)

The system should accurate and appropriate English translations for recognized Urdu words and sentences. The translated text should be grammatically correct and maintain the real meaning.

3.3.10 Graphical User Interface

The system should have a user-friendly web-based graphical user interface (GUI). The GUI should be easy to navigate and use, even for users with limited technical expertise. The GUI should also provide clear and informative visual feedback to the user throughout the interaction process, such as displaying the recognized sign/word, providing progress updates, and indicating any errors.

3.3.11 Web Compatibility

The system should be accessible and functional across multiple platforms, including web browsers (Chrome, Opera, Edge, etc.) and mobile browsers (Chrome, Safari, etc.) running different operating systems (Windows, macOS, Linux, iOS, Android).

3.3.12 Security and Privacy

The system should handle user data, including video recordings and personal information, securely and confidentially. The system should comply with relevant data privacy regulations and ensure user data is protected from unauthorized access or disclosure.

3.3.13 Error Handling

The system should effectively detect and handle potential errors, such as incorrect gesture inputs, network connectivity issues, or system malfunctions. In case of errors, the system should provide clear and informative error messages to the user.

3.4 Use Case Diagram

The use case diagram in Figure 3.2 illustrates the process of recognizing Urdu sign language gestures. The user initiates the recognition process by signing an Urdu alphabet, number, or word in front of the camera. The system continuously captures the video stream from the camera and processes it in real-time. Utilizing computer vision techniques, the system accurately detects and tracks the user's hand gestures, identifying key landmarks and movements. Subsequently, the system analyzes the captured hand gestures to identify the corresponding Urdu sign, whether it's an individual alphabet, a number, or a

combination of signs representing a word. Finally, the system displays the recognized sign (alphabet, number, or word) clearly on the screen for the user's reference.

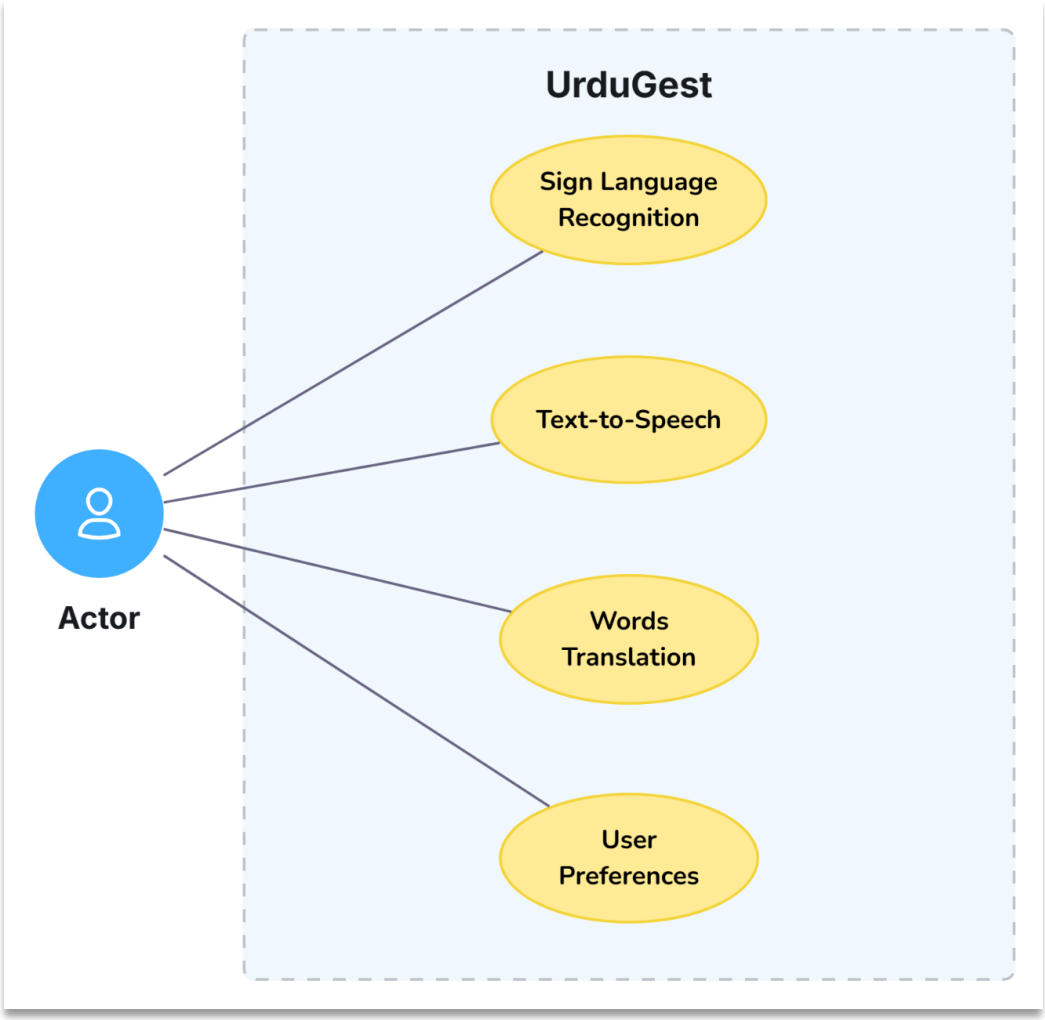

Figure 3. 2 Use Case Diagram

3.5 Use Cases

3.5.1 Use Case: Sign Language Recognition

Table 3.5.1 outlines the process through which the system captures and interprets hand gestures using the webcam and classifies them into corresponding Urdu alphabet signs through the trained model.

Table 3.5. 1 Use Case: Sign Language Recognition

Attribute	Description
Use Case Name	Sign Language Recognition
Scope	Recognizing and interpreting Urdu sign language gestures.
Level	User Goal
Primary Actor	User (individual with hearing or speech impairment)
Stakeholders & Interests	User: Effective communication, improved accessibility. Developer: System accuracy, user satisfaction.
Pre-conditions	System is properly installed and configured. Camera is connected and functioning correctly.
Success Guarantee	The system accurately recognizes the user's sign language gesture and displays the corresponding Urdu text.
Main Success Scenario	1. User signs an Urdu alphabet, number, or common action. 2. System initializes the video stream. 3. System detects and tracks hand gestures. 4. System analyzes gestures and identifies the corresponding Urdu sign. 5. System displays the recognized sign on the screen.

3.5.2 Use Case: Text-to-Speech Conversion

Table 3.5.2 highlights how the system converts the recognized Urdu text into urdu speech, enabling effective verbal communication for non-sign language users.

Table 3.5. 2 Use Case: Text-to-Speech Conversion

Attribute	Description
Use Case Name	Text-to-Speech Conversion
Scope	Converting recognized text into spoken Urdu.

Level	User Goal
Primary Actor	User
Stakeholders & Interests	User: Improved accessibility, easier understanding. Developer: Accurate speech synthesis, user satisfaction.
Pre-conditions	Recognized Urdu text is available.
Success Guarantee	The system accurately converts the recognized Urdu text into high-quality, natural-sounding spoken Urdu.
Main Success Scenario	1. Recognized Urdu text is available. 2. System utilizes a text-to-speech engine. 3. System synthesizes and plays the spoken Urdu output.

3.5.3 Use Case: English Translation

Table 3.5.3 represents the translation use case, showing how the recognized Urdu text is translated into English, facilitating bilingual communication and wider accessibility.

Table 3.5. 3 Use Case: English Translation

Attribute	Description
Use Case Name	English Translation
Scope	Translating recognized Urdu words and numbers into English.
Level	User Goal
Primary Actor	User
Stakeholders & Interests	User: Enhanced communication with non-sign language users. Developer: Accurate and fluent translations.
Pre-conditions	Recognized Urdu text is available.
Success Guarantee	The system accurately translates the recognized Urdu words and numbers into grammatically correct and contextually appropriate English.

Main Success Scenario	1. Recognized Urdu text is available. 2. System utilizes a translation API. 3. System displays the English translation along with the Urdu text.

3.5.4 Use Case: User Preferences

Table 3.5.4 explains how the user can manage personal preferences, such as enabling/disabling features like speech output or translation, to customize their interaction with the system.

Table 3.5. 4 Use Case: User Preferences

Attribute	**Description**
Use Case Name	User Preferences
Scope	Adjusting system preferences.
Level	User Goal
Primary Actor	User
Stakeholders & Interests	User: Personalization, improved usability. Developer: User control over system behavior.
Pre-conditions	System is running.
Success Guarantee	The user successfully adjusts the desired system settings.
Main Success Scenario	1. User accesses the settings menu. 2. User selects and modifies desired settings (e.g. select between desired recognition modes, switch between light and modes, adjusting minimum confidence, submit feedback). 3. System applies the changes and updates the user interface accordingly.

3.6 Non-functional Requirements

Non-functional requirements define the quality attributes and constraints that the "UrduGest" system must meet to be considered successful.

3.6.1 Performance:

The system should exhibit strong performance characteristics, including real-time processing capabilities. This involves processing video frames and recognizing gestures with minimal latency to ensure a smooth and responsive user experience. The system should also maintain a consistent frame rate for real-time video processing, typically above 24 frames per second, while efficiently utilizing system resources (CPU, memory) to minimize performance impact on the user's device.

3.6.2 Reliability:

The system must be reliable and robust in its operation. This includes ensuring system stability with minimal occurrences of crashes or unexpected errors. High accuracy in gesture recognition is crucial, minimizing false positives and false negatives to provide accurate and reliable communication.

3.6.3 Usability:

The system should be highly usable and accessible to all target users. This involves ensuring the system is user-friendly and easy to understand, even for users with limited technical expertise. The system should also be accessible to users with varying levels of visual and motor abilities.

3.6.4 Security:

The system must prioritize the security and privacy of user data. This includes implementing robust measures to protect user data, such as video recordings and personal information, from unauthorized access or disclosure.

3.6.5 Maintainability:

The system should be designed with maintainability in mind. This involves ensuring the code is well-documented and easy to maintain and update. Furthermore, the system should have a modular design, allowing for easy addition or modification of features as needed.

3.7 Resource Requirements

Resource requirements refer to the essential resources needed for the successful implementation of the "UrduGest" project. These include:

- **Hardware:**

- Computers with sufficient processing power (CPU and GPU) for model training and real-time processing.
- High-quality webcams or cameras for capturing clear video footage of sign language gestures.

- **Software:**
  - Anaconda as Python Development environment (Jupyter Notebook) with Python 3.12 installed and necessary libraries such as Tensorflow, OpenCV, Scikit-Learn, MediaPipe, Streamlit, LughaatNLP. Transformers, Pickle.
  - Urdu Text-to-speech API
  - Hugging Face Urdu Translation API

- **Human Effort:**
  - Skilled developers with expertise in Python, computer vision, machine learning, and web development.
  - Data collectors to acquire and label the necessary datasets of Urdu sign language gestures.

3.8 Database Requirements

The "UrduGest" project relies heavily on a robust dataset of Urdu sign language gestures for successful model training and evaluation. A substantial dataset [5] with a diverse range of sign language gestures is crucial, encompassing variations in lighting conditions, background environments, and individual signing styles to ensure the system's robustness and generalizability. High-quality clear and consistent images of sign language gestures [6] with minimal noise or interference. Accurate and consistent annotations of each image frame are crucial for effective model training and evaluation. Furthermore, a well-organized and efficient system for managing and storing the dataset is essential for data integrity, accessibility, and overall project success.

3.9 Project Feasibility

3.9.1 Technical Feasibility

The project is technically feasible, leveraging advancements in computer vision, machine learning, and natural language processing. The availability of libraries such as **OpenCV**,

MediaPipe Scikit-Learn, and **Streamlit** provides a solid foundation for the development of the system. The use of a web-based application allows for easy accessibility and deployment across multiple platforms.

3.9.2 Operational Feasibility

The proposed system aligns with the needs and requirements of the target users – individuals with hearing impairments. The iterative development approach allows for continuous evaluation and refinement of the system based on user feedback.

3.9.3 Economic Feasibility

The project can be developed with minimal financial investment, primarily relying on open-source software and commonly available hardware. The potential social and economic impact of the system, in terms of improved communication and inclusivity for the deaf community, justifies the development effort.

3.10 Summary

This chapter comprehensively outlines the requirements for the "UrduGest" project, covering interface specifications, functional capabilities, user interactions, and non-functional constraints. It also addresses resource needs, database requirements, and a feasibility study. These requirements will serve as a crucial foundation for the subsequent design and development phases, ensuring the final system effectively meets the needs of users with hearing impairments while adhering to technical and operational considerations.

CHAPTER 4:

SYSTEM MODELLING

This chapter focuses on the system design and analysis of the "**UrduGest**" project. The "UrduGest" system is an AI-powered application designed to recognize Urdu Sign Language gestures in real-time. It utilizes computer vision techniques, machine learning algorithms, and natural language processing to accurately interpret hand gestures, translate them into Urdu text and speech, and provide English translations.

This chapter will explain various system design and analysis diagrams, including use case diagrams, activity diagrams, data flow diagrams, sequence diagrams, and class diagrams, to visualize and understand the system's architecture and behavior. These diagrams will provide a comprehensive representation of the system's components, interactions, and data flow.

4.1 System Design and Analysis

System design and analysis is a crucial phase in the software development lifecycle. It involves a thorough examination of the system's requirements and the translation of these requirements into a detailed design specification.

This chapter focuses on the system design and analysis of the "UrduGest" project. The "UrduGest" system is an AI-powered application designed to recognize Urdu Sign Language gestures in real-time. It utilizes computer vision techniques and machine learning algorithms to recognize and interpret hand gestures, translate them into Urdu text and speech, and provide English translations.

This chapter will explore various system design and analysis diagrams, including use case diagrams, activity diagrams, data flow diagrams, sequence diagrams, and class diagrams, to visualize and understand the system's architecture and behavior. These diagrams will provide a comprehensive representation of the system's components, interactions, and data flow.

4.2 Design Approach

The "UrduGest" project employs a top-down design approach for its development. This approach involves breaking down the system into smaller, more manageable modules or subsystems, starting with a high-level overview and gradually progressing towards more detailed specifications for each component.

This approach is specifically suitable for the "UrduGest" project due to its complexity. By starting with a high-level view and gradually refining the design, the top-down approach provides a structured way to manage the complexity of the system, which involves multiple interconnected components such as real-time video processing and gesture detection, text-to-speech and English translation.

By starting with a shared high-level understanding of the system architecture, each team member can focus on a specific module or subsystem while maintaining an overall understanding of the system's design.

The top-down approach also allows for flexibility and adaptability during the development process. If changes or modifications are required, they can be implemented at the required level without impacting the overall system design.

4.3 Interface Design

The user interface (UI) of the "UrduGest" system plays a crucial role in its usability and accessibility. A well-designed interface ensures a seamless and intuitive user experience, making the system easy to use and understand, even for individuals with limited technical experience.

4.3.1 Login Page

The Login Page as illustrated in Figure 4.1, will authenticate/register users and control access to the system's features. It will include elements such as username/email fields, password fields, a login button and "Register" radio button for new users. The design will prioritize a clean and minimalist look with clear instructions and secure password handling using hashing method.

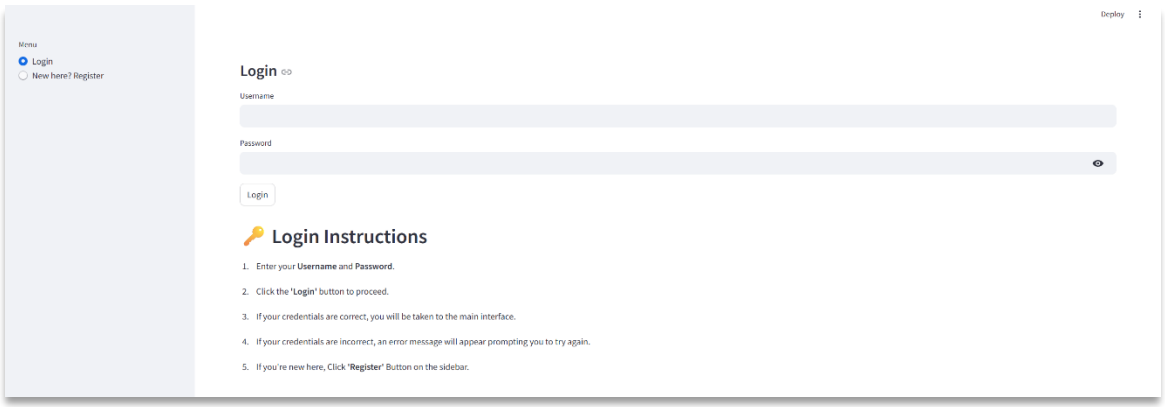

Figure 4. 1 Login Interface

4.3.2 Menu Page

The Menu Page as illustrated in Figure 4.2, will provide users with navigation options and access to different system functionalities. It will have a drop-down menu that will consist of main four sections i.e. Alphabets, Numbers, Common Actions, and Words/Sentence Formation. It will include elements such as a "**Start Recognition**" button (in all four sections) to initiate the real-time gesture recognition process, "**App Controls**" to allow users to adjust system preferences like "**Confidence**" section to adjust minimum detection confidence of the system model. The menu structure will be designed to be clear, concise, and easy to navigate through.

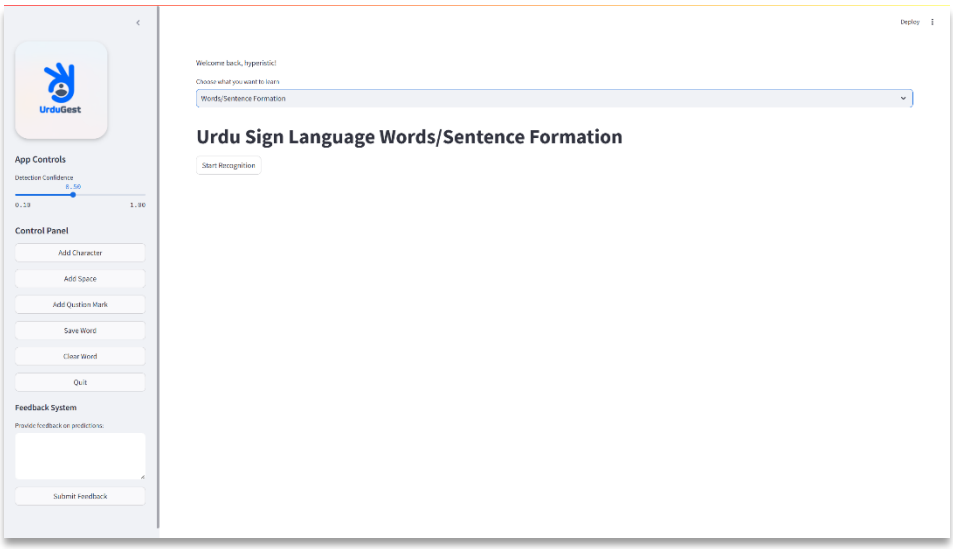

Figure 4. 2 Main Menu Interface

4.3.3 Sign Language Recognition Page

The Sign Language Recognition Page serves as the core interface for real-time sign language recognition. As illustrated in Figure 4.3, this page will display the live camera feed, visualize hand pose estimation with hand landmarks and bounding boxes, and display the recognized Urdu word/sentence in real-time. Moreover, it will provide its English translation and convert Urdu Text-to-speech. It will also include buttons to start recognition, Add Character/Space/Question mark, Save/Clear Word and adjust detection confidence slider. The design will prioritize a clear and concise display of information, minimal visual clutter, and intuitive controls.

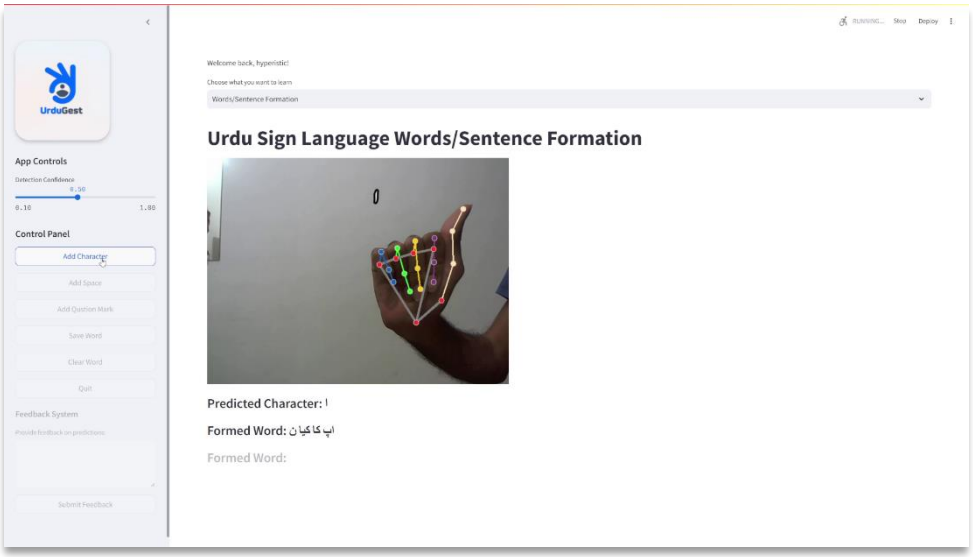

Figure 4. 3 Sign Language Recognition

4.4 Use Case Diagram

The use case diagram, as illustrated in Figure 4.4, provides a visual representation of the interactions between the user and the "UrduGest" system. The diagram depicts the primary actor, the "User," interacting with various use cases within the system. These use cases include:

- **Login/Register:** The user can interact with the system through a login or registration process, allowing for personalized access and potentially customized settings.
- **Sign Language Recognition:** The core functionality of the system, enabling the user to perform Urdu sign language gestures, which are then captured, processed, and recognized by the system.
- **Text-to-Speech Conversion:** The system can convert the recognized Urdu text into spoken Urdu, providing an auditory output for the user.
- **Words Translation:** The system can translate recognized Urdu words into English, facilitating communication with individuals who do not understand Urdu sign language.
- **Submit Feedback:** Users can provide feedback or suggestions to the system developers to improve the system's functionality and user experience.

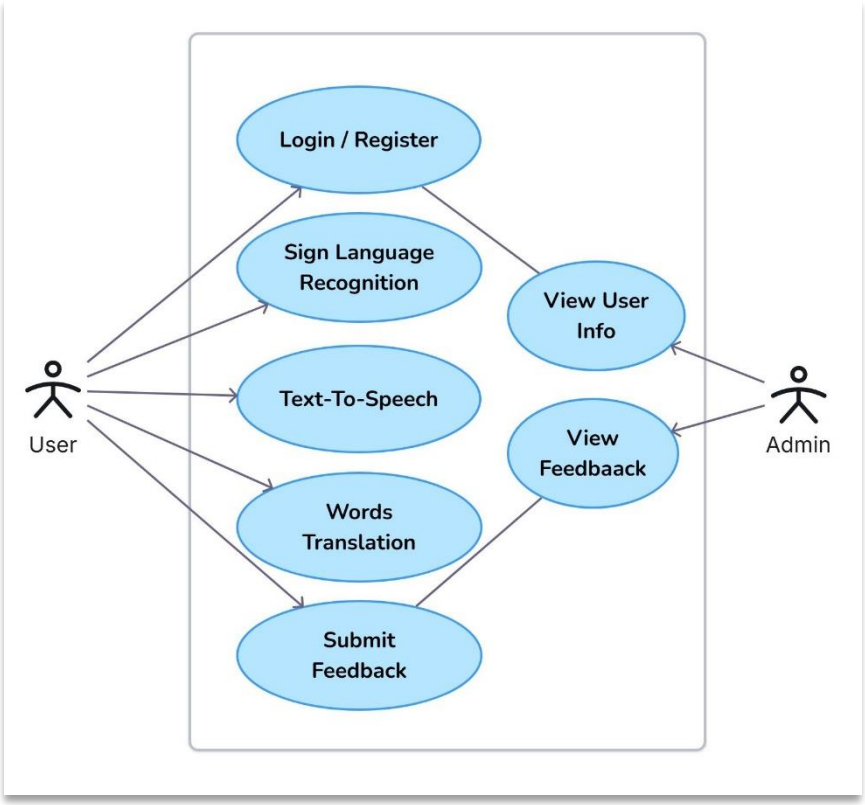

Figure 4. 4 Use Case Diagram

4.5 Data Flow Diagram

Data Flow Diagrams (DFDs) provide a graphical representation of the flow of data within a system. They are valuable for understanding the data inputs, processes, outputs, and data stores involved in the system's operation.

The Level 1 DFD as illustrated in Figure 4.5, provides a more detailed view of the system by breaking down the main process into smaller sub-processes. For the "UrduGest" system, the Level 1 DFD would decompose the main process into sub-processes such as video capture and processing, hand gesture recognition, text-to-speech conversion, translation, and user interface management. Figure 4.4 illustrates the Level 1 DFD for the "UrduGest" system, showing the decomposition of the main process into these key sub-processes and the flow of data between them.

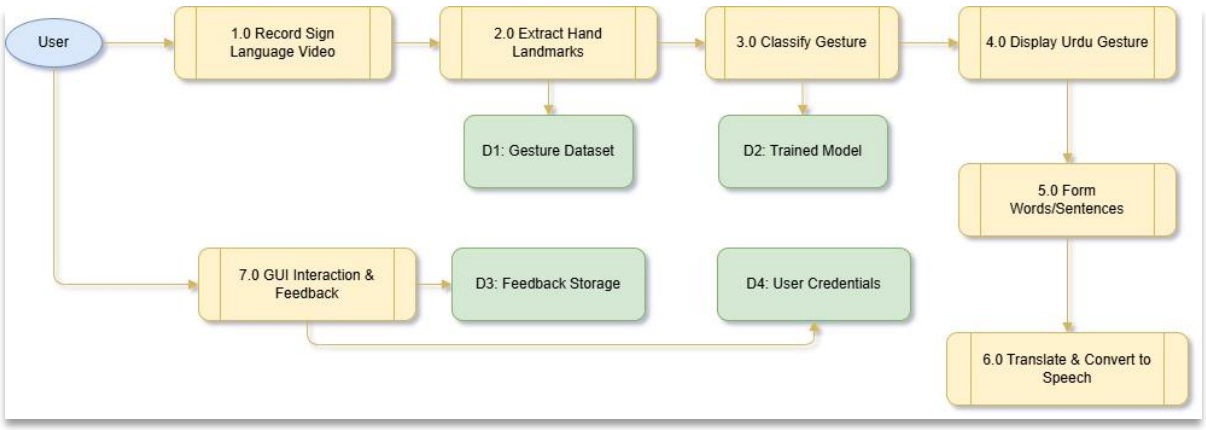

Figure 4. 5 Data Flow Diagram Level 1

4.6 System Sequence Diagram

The System Sequence Diagram as illustrated in Figure 4.6, provides a representation of the interactions of sign language recognition system, aiding in the understanding of system behavior.

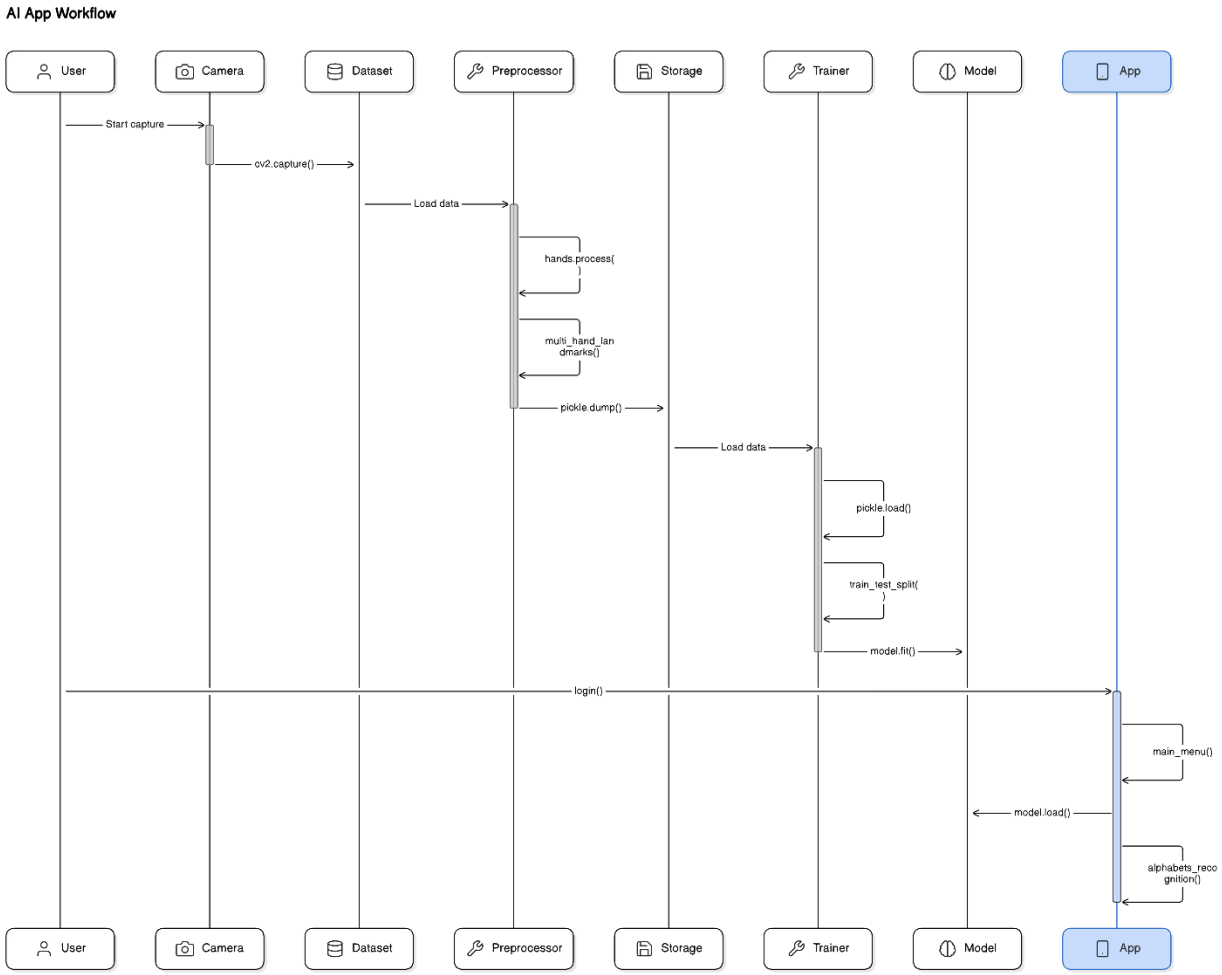

Figure 4. 6 System Sequence Diagram

4.7 Sequence Diagram

The Sequence Diagram as illustrated in Figure 4.7, provides a clear and concise representation of the interactions between the user and the system during the sign language recognition process, aiding in the understanding of system behavior and identifying potential design issues.

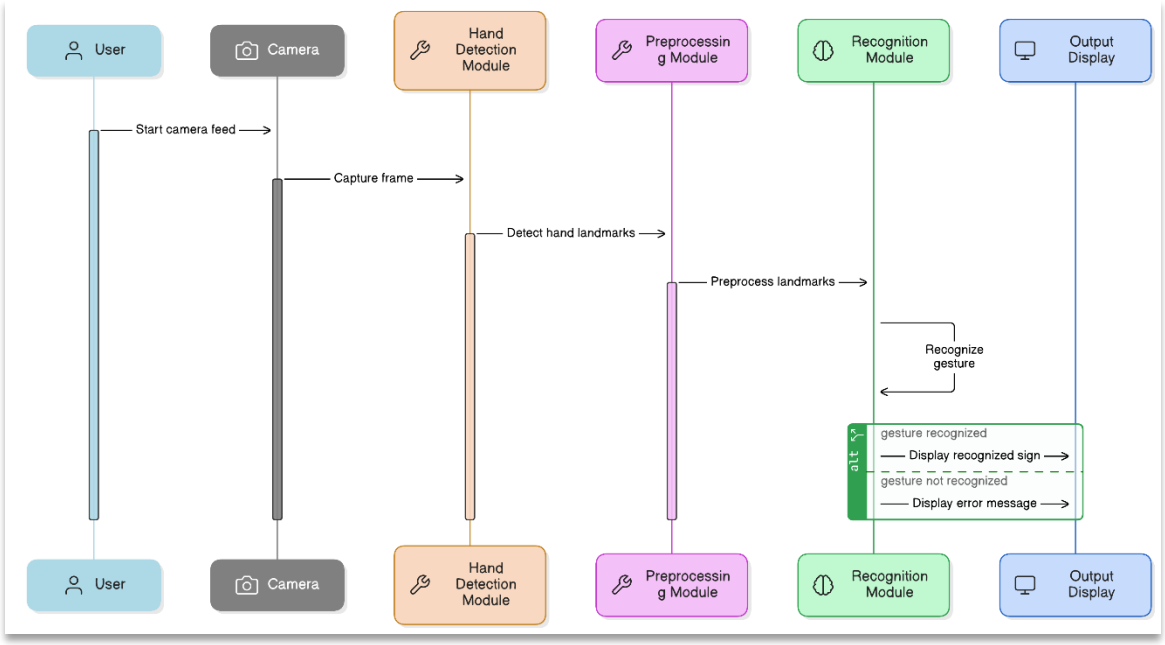

Figure 4. 7 Sequence Diagram

4.8 Activity Diagram

Figure 4.8 illustrates the main activities involved in the sign language recognition process within the "UrduGest" system. This diagram provides a clear and concise overview of the workflow, aiding in the understanding of the system's operational flow.

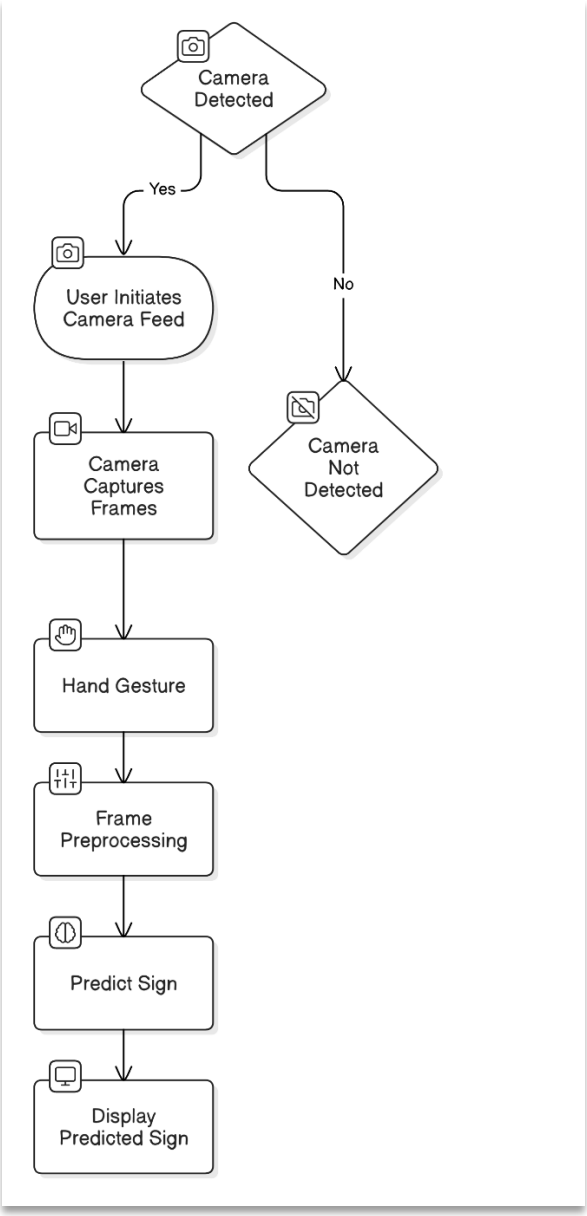

Figure 4. 8 Activity Diagram

4.9 Design Class Diagram

The Design Class Diagram (DCD) provides a visual representation of the backend structure of the system, as illustrated in Figure 4.9. It illustrates the classes, their methods, and relationships. For the "UrduGest" system, the DCD will depict key modules such as "Dataset Collection," "Preprocessing," "Model training" and "GUI." These classes will represent the core components of the system and their interactions, providing a blueprint for the system's implementation.

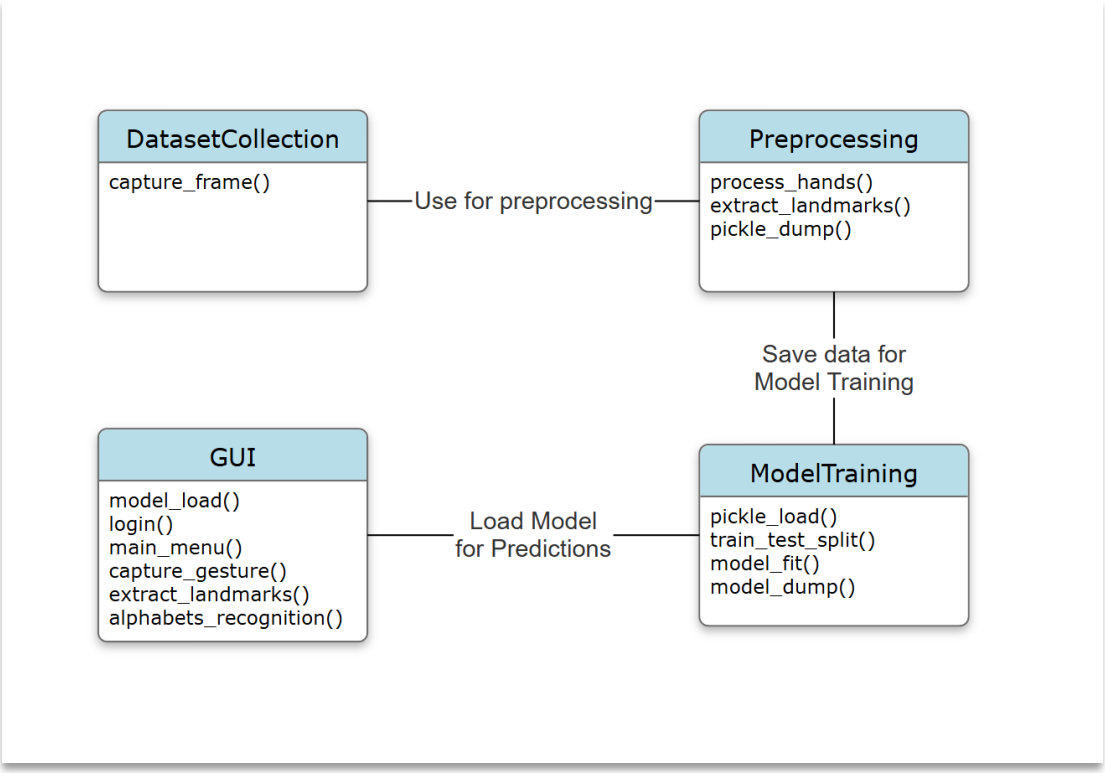

Figure 4. 9 Class Diagram

4.10 Architectural Diagram

The "UrduGest" system will adopt a layered architecture, as illustrated in Figure 4.10. This architecture will separate the system into distinct layers, such as the user interface layer, the application logic layer, and the data access layer. This layered approach enhances modularity, maintainability, and reusability of components. The architectural diagram will provide a visual representation of these layers and their interactions.

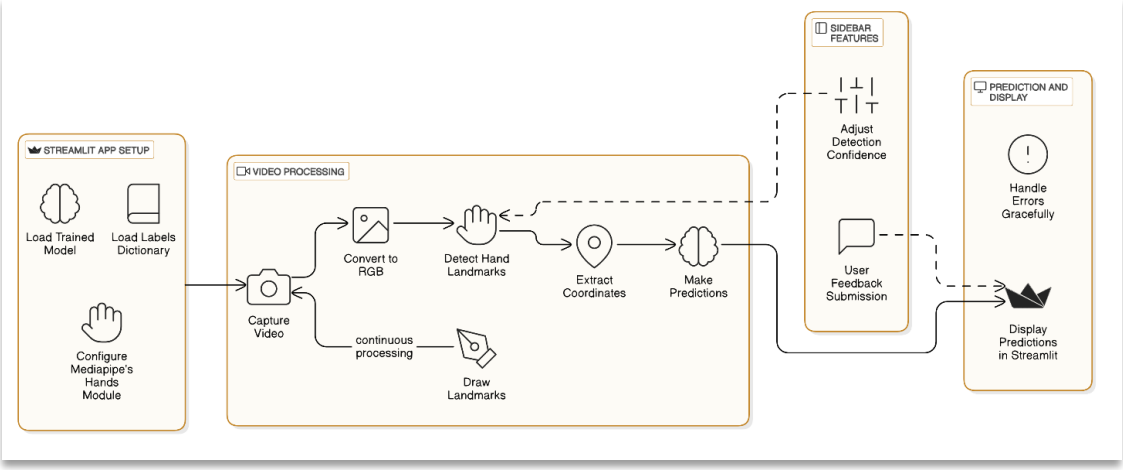

Figure 4. 10 Architectural Diagram

4.11 Summary

This chapter has presented the system design and analysis for the "UrduGest" project. It explored various design models, including use case diagrams, activity diagrams, data flow diagrams, sequence diagrams, and class diagrams, to visualize and understand the system's structure and behavior. These models provide a comprehensive understanding of the system's components, interactions, and data flow, guiding the development process and ensuring the successful implementation of the "UrduGest" system.

CHAPTER 5:

IMPLEMENTATION

5.1 Introduction

This chapter explains how we turned our Urdu Sign Language Recognition project from an idea into a working system. We will walk through the different parts of the project, including how each one contributes to recognizing and translating sign language into text and speech.

The chapter covers the various steps in the implementation process, such as capturing video, detecting hand landmarks, training the model to recognize different signs, and integrating text-to-speech and translation features. We also discuss the libraries and tools used, like MediaPipe for hand tracking, OpenCV for capturing video, and other services for text translation and speech generation.

Finally, we'll talk about how all these components work together to create a seamless experience, from detecting a sign to outputting the translated speech. The goal of this chapter is to show how each part of the system contributes to making the Urdu Sign Language Recognition System function effectively.

5.2 Modules of the Project

This section explains the different modules that make up the Urdu Sign Language Recognition system. Each module is designed to perform a specific task, working together to recognize, translate, and convert sign language into text and speech. The following sub-sections describe the algorithms, libraries, and frameworks used in the backend and frontend of the system.

5.2.1 Dataset Collection and Preprocessing

The dataset collection module involves capturing images of various sign language gestures using a webcam and OpenCV. This was done under varied lighting conditions and from different angles to ensure robustness. The data is collected in real time to cover all possible variations in sign language gestures. Once the raw data is captured, it is processed and stored for future use in training and validation. Each gesture is associated with a specific label representing the Urdu alphabet or action. This comprehensive dataset serves as the foundation for training machine learning models and ensures the system can recognize a wide range of gestures.

5.2.2 Hand Landmark Detection

Mediapipe [7], a library by Google, is employed in this module for real-time hand landmark detection. Mediapipe detects and tracks 21 [8] key points on each hand, which are essential for accurately recognizing hand gestures in sign language. These landmarks are used to analyze the gestures and map them to the corresponding Urdu characters. Mediapipe's efficiency allows for accurate tracking even in real-time video streams, making it ideal for live gesture recognition. The extracted landmarks are then stored and further processed for gesture classification.

5.2.3 Model Training and Classification

In this module, a machine learning model [9] is trained using the Random Forest Classifier. The training process uses a labeled dataset of hand landmarks, where each gesture is associated with a specific label. The dataset is split into training and testing sets to ensure the model generalizes well on unseen data. After training, the Random Forest model's performance is evaluated, and it achieves a high accuracy rate in recognizing gestures based on the hand landmarks. This trained model is saved and used for real-time gesture recognition in the application.

5.2.4 Real-Time Gesture Recognition

The real-time recognition module uses the trained Random Forest Classifier model to identify hand gestures in a live video feed. OpenCV is used to capture the video stream and process each frame. As the user performs a gesture, the landmarks are extracted, and the model classifies the gesture, displaying the corresponding Urdu text in real-time. The Urdu characters are shown using a clear, legible font (Jameel Noori Nastaleeq) to ensure readability. This module enables the user to interact with the system in real time, making it more intuitive and efficient for practical use.

5.2.5 GUI and User Authentication System

The GUI module is built using Streamlit, providing a user-friendly interface that allows seamless interaction with the system. The interface includes a login page, where users can either sign in with existing credentials or register for a new account. Passwords are securely hashed using the bcrypt library to ensure user data security. After logging in, users are directed to the main interface, where they can select different sign language categories (Alphabets, Numbers, Common Actions, and Formed Words/Sentences) to begin practicing

or recognizing gestures. This module enhances the usability of the system and makes it accessible to non-technical users.

5.2.6 Words and Sentences Formation Module

This module allows the system to recognize individual gestures and combine them to form words or sentences. As the user performs gestures corresponding to the Urdu alphabet, the system allows them to add the recognized characters to form a word. The user can also add spaces, clear the word, or save the final output. Once a sentence is formed, it can be displayed in the interface, and the system can use translation services to convert the Urdu text into English, making communication smoother and more interactive. This module plays a vital role in bridging communication between sign language users and non-sign language speakers.

5.2.7 Translation and Text-to-Speech Module

In this module, the recognized Urdu text is converted into speech and translated into English for broader accessibility. After recognizing and forming a word or sentence, the system uses a text-to-speech service to generate audio output of the Urdu text. The translation functionality is powered by the Hugging Face API, which translates the Urdu text into English. This module enhances the system's ability to assist non-sign language users and facilitates better communication between people who use sign language and those who don’t,

5.3 Hardware Module Details

The hardware used in this project is minimal but essential for real-time gesture recognition. The main hardware component is a webcam, which acts as the input device for capturing live hand gestures.

An external webcam was used to continuously record video frames. These frames served as the input for the entire recognition system. The OpenCV (cv2) library was used to access the camera and process each frame in real-time. The camera provided a steady stream of visual data, which was then passed to the MediaPipe library for extracting hand landmarks.

The webcam also played a crucial role in the testing and validation phase, where real-time recognition of Urdu alphabets, numbers, and common actions was performed. It allowed

the system to analyze live hand movements, recognize the gesture, and display the output instantly using Urdu text rendered with Jameel Noori Nastaleeq font.

No additional hardware such as microphones, external sensors, or controllers was required. The goal was to keep the system low-cost, simple, and accessible, making it easier to use in educational or assistive applications where specialized hardware might not be available.

5.4 Summary

This chapter explained the complete implementation of the Urdu Sign Language Recognition System. It covered how different modules were developed, including dataset collection, landmarks extraction, model training, real-time recognition, and the user interface. Each part was discussed in detail to show how the system works together to perform live gesture recognition and display the results in Urdu text. The hardware used was kept simple, with a webcam serving as the primary input device. This made the system more accessible and affordable for real-world use.

CHAPTER 6:

RESULT/TESTING, ANALYSIS AND VALIDATION

6.1 Introduction

This chapter presents the testing, results, and performance analysis of the Urdu Sign Language Recognition System. The goal was to evaluate how well each module performs, especially during real-time usage. We tested the accuracy of the trained models, the system's ability to detect and classify signs correctly, and the smooth functioning of the complete interface under different conditions. Each part of the system, from data collection to live prediction and user feedback, was tested using practical scenarios. The results were analyzed to check if the system meets the goals set at the beginning of the project. The following sections give a detailed breakdown of the testing process, results, and how reliable and accurate the system is in recognizing Urdu signs through a webcam.

6.2 Testing, Analysis and Validation

- Unit Testing: Individual components such as model loading, character appending, and file handling were tested independently to ensure correctness of internal logic.
- Integration Testing: After unit testing, modules were combined and tested to ensure proper communication between input processing, prediction, and GUI layers.
- Functional Testing: The system was tested from a user's perspective to verify that each feature works as intended, including login, selection, and gesture display.
- Real-Time Testing: Live gesture recognition was tested using a webcam under different lighting conditions to evaluate responsiveness, accuracy, and usability.

6.2.1 Test Case 1 – User Login Functionality

Table 6.2.1 test case validates the login feature by checking whether existing users can log into the system securely and access relevant functionalities.

Table 6.2. 1 Test Case 1 – User Login Functionality

Test Description	**User Login Functionality**
Test Result	✓ Pass
Requirement(s) to be Tested	User Authentication
Test Items and Features	Login Page, bcrypt-based password validation

Procedural Steps	1. Open GUI 2. Enter correct username/password 3. Click Login
Expected Results	User should be logged in and redirected to main page
Output Results	Login successful, access granted

6.2.2 Test Case 2 – Register New User

Table 6.2.2 presents the testing of the user registration module, ensuring that new users can successfully create an account and store their credentials in the system.

Table 6.2. 2 Test Case 2 – Register New User

Test Description	**Register New User**
Test Result	✓ Pass
Requirement(s) to be Tested	User Registration and Secure Storage
Test Items and Features	Register Form, bcrypt, File Handling
Procedural Steps	1. Open GUI 2. Enter new username/password 3. Click Register
Expected Results	New user credentials stored securely
Output Results	Registration successful, new entry created

6.2.3 Test Case 3 – Urdu Alphabets Prediction

In Table 6.2.3, the test focuses on verifying whether the system can accurately recognize and display individual Urdu alphabets through hand gesture inputs.

Table 6.2. 3 Test Case 3 – Urdu Alphabets Prediction

Test Description	**Urdu Alphabets Prediction**

Test Result	✓ Pass (with 85% percent accuracy)
Requirement(s) to be Tested	Gesture Prediction for Urdu Alphabets using Trained Model
Test Items and Features	Webcam, MediaPipe, Random Forest Alphabet Model
Procedural Steps	1. Open GUI 2. Select "Alphabets" 3. Show gesture to camera
Expected Results	Urdu alphabet label displayed above bounding box
Output Results	Correct alphabet detected and displayed

6.2.4 Test Case 4 – Common Actions/Numbers Prediction

In Table 6.2.4, the test case checks the recognition accuracy of commonly used signs and numeric gestures, evaluating the system's ability to identify gestures correctly.

Table 6.2. 4 Test Case 4 – Common Actions/Numbers Prediction

Test Description	**Common Actions/Numbers Prediction**
Test Result	✓ Pass (with 87% percent accuracy)
Requirement(s) to be Tested	Classification Accuracy with Action Model
Test Items and Features	Dropdown Menu, Common Actions Model
Procedural Steps	1. Select "Common Actions" from dropdown 2. Perform gesture
Expected Results	Action label should be shown above the bounding box
Output Results	Gesture detected and displayed correctly

6.2.5 Test Case 5 – Formed Words and Sentences

The Table 6.2.5 tests whether the system can successfully form meaningful words or sentences by combining individually recognized letters in sequence.

Table 6.2. 5 Test Case 5 – Formed Words and Sentences

Test Description	**Formed Words and Sentences**
Test Result	✓ Pass
Requirement(s) to be Tested	Word and sentences formation using sidebar buttons
Test Items and Features	Sidebar Buttons (Add, Space, Question mark, Save word, Clear word, Quit)
Procedural Steps	1. Select Formed Word/Sentence 2. Detect gesture 3. Use buttons to form word or sentence 4. Save word
Expected Results	Characters collected and final word formed
Output Results	Word/Sentence displayed correctly in output section

6.2.6 Test Case 6 – Text-to-Speech Output

In Table 6.2.6, the test ensures that the system is able to correctly convert the recognized text into natural-sounding Urdu speech for improved communication.

Table 6.2. 6 Test Case 6 – Text-to-Speech Output

Test Description	**Text-to-Speech Output**
Test Result	✓ Pass
Requirement(s) to be Tested	Audio Output Generation
Test Items and Features	LughaatNLP TTS
Procedural Steps	1. Save word 2. Output below formed word
Expected Results	Urdu speech generated from text

Output Results	Clear audio output played through system

6.2.7 Test Case 7 – Urdu to English Translation

The table 6.2.7 represent final test case that evaluates the accuracy and relevance of the English translation produced by the system from Urdu sign input, ensuring meaningful conversion.

Table 6.2. 7 Test Case 7 – Urdu to English Translation

Test Description	**Urdu to English Translation**
Test Result	✓ Pass
Requirement(s) to be Tested	API Integration for Translation
Test Items and Features	HuggingFace Translation API
Procedural Steps	1. Save word 2. Output below TTS output
Expected Results	English version of sentence displayed
Output Results	Accurate translation shown in GUI

6.3 Summary

In this section, we tested and validated the functionality of all key features in the project, ensuring they met the requirements. Using a mix of unit, integration, and functional testing approaches, we covered critical components such as user login, gesture recognition, and sentence building. Each feature, including the real-time gesture detection, translation, and text-to-speech functionalities, was verified through individual test cases. We employed structured testing methodologies to guarantee that each part of the system operated correctly, with detailed checks for the output results. The performance of the model, particularly in real-time gesture detection, was also validated through practical testing with different gestures. Additionally, we confirmed that the translation and TTS features produced accurate and reliable outputs.

CHAPTER 7:

CONCLUSION AND FUTURE WORK

7.1 Introduction

This chapter summarizes the work carried out during the development of the Urdu Sign Language Recognition system. It highlights the overall achievements, evaluates them against the initial goals, and reflects on the limitations encountered during the process. Additionally, it presents a roadmap for future enhancements to improve the performance, usability, and scalability of the system. The aim is to provide a clear picture of what has been accomplished and how the project can evolve further.

7.2 Conclusion and Future Work

7.2.1 Proposed Objectives vs Final Implementation

The main objective of this project was to design a system capable of recognizing Urdu Sign Language gestures in real-time, translate them into readable text, and provide additional features like sentence formation, text-to-speech, and translation. All core objectives were successfully met:

- Custom datasets for Urdu alphabets, numbers, and common actions were created.
- A machine learning model was trained with satisfactory accuracy using a Random Forest Classifier.
- Real-time recognition was implemented with efficient landmark detection.
- A user-friendly GUI was built for both login and main recognition interface.
- Additional features like sentence formation, Urdu-to-English translation, and Urdu speech output were added.

These results closely align with the project's initially defined goals.

7.2.2 Key Achievements

- Custom datasets were recorded for 37 Urdu alphabets, 10 numbers, and 7 common actions using OpenCV.
- MediaPipe was successfully used to extract hand landmarks for each image, with data saved using pickle.
- The Random Forest model achieved an accuracy of **85%** for alphabets, **87%** for numbers, and **99%** for common actions, showing effective classification performance.

- Integrated CV2 live webcam feed with bounding boxes and gesture prediction in Nastaleeq font.
- Used Pickle library to save landmarks and model data.
- Streamlit-based interface supported secure user authentication, adjustable confidence sliders, and feedback system.
- Included sentence creation, Urdu-to-English translation via Hugging Face, and text-to-speech via LughaatNLP.

7.2.3 Limitations

- The system is sensitive to variations in lighting and background, which can affect landmark detection and overall recognition accuracy.
- The current model supports only one-hand static gestures and does not account for dynamic or multi-hand signs.
- The model doesn't support facial expressions yet, which might be important in some other sign languages.

7.2.4 Future Work

- Implementation of additional sign languages such as JSL, CSL, or ESL to expand system usability across different regions.
- Add functionality for recognizing gestures involving both hands.
- Combine hand landmarks with facial expressions to recognize context-dependent or expressive signs.
- Convert the current UI into a mobile and desktop app to increase accessibility and user reach.

7.3 Summary

This chapter presented the conclusion and future direction of the project. The system was reviewed against the original objectives, which included developing a real-time recognition framework for Urdu sign language with features like text display, sentence formation, translation, and text-to-speech. Each objective was successfully met, and the system was implemented with good performance. Achievements included a stable GUI, an effective model with 85% accuracy, and smooth real-time predictions. However, limitations such as dependency on consistent lighting and limited gesture types were acknowledged. Future

work may include expanding the system to support other regional and international sign languages and enhancing recognition to handle dynamic or two-handed signs.

References

[1] A. Dhanjee, "YesPrograms," 2014. [Online]. Available: https://www.yesprograms.org/stories/sign-language-accessibility-for-the-deaf-in-pakistan#:~:text=According%20to%20the%20World%20Health%20Organization%2C%205%,providing%20accessibility%20to%20these%20ten%20million%20people..

[2] M. M. R. e. al., "A New Benchmark on American Sign Language Recognition using Convolutional Neural Network," 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9067974.

[3] K. e. al., "Enhancing Communication Accessibility: UrSL-CNN Approach to Urdu Sign Language Translation for Hearing-Impaired Individuals," 2024. [Online]. Available: https://www.sciencedirect.com/org/science/article/pii/S1526149224002406.

[4] H. e. al., "Recognition of Urdu sign language: a systematic review of the machine learning classification," 2022. [Online]. Available: https://peerj.com/articles/cs-883/.

[5] A. e. al., "Dataset of Pakistan Sign Language and Automatic Recognition of Hand Configuration of Urdu Alphabet through Machine Learning," 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235234092100305X?via%3Dihub.

[6] S. e. al., "UAlpha40: A comprehensive dataset of Urdu alphabet for Pakistan sign language," 2025. [Online]. Available: http://sciencedirect.com/science/article/pii/S2352340925000745.

[7] C. Lugaresi, "MediaPipe: A Framework for Building Perception Pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172.

[8] Google, "Mediapipe Hand landmarks detection guide," [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker.

[9] M. J. e. al., "Sign Language Recognition using Machine Learning," 2022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9914155?casa_token=2-lgBI4ZaX0AAAAA:EzWBl8gSX6ccr29aJaz75fOPbqXqPtuBjg-FkzmwfTOm7wSh4sA7vnR15gXIWZ_8dKvz0SiAfVrm.

[10] C. L. e. al., "MediaPipe: A Framework for Building Perception Pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172.