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