

**TEST PLAN FOR**

**Sign Language Recognition System**

*ChangeLog*

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| 001 | 30.10.2023 | Vishakha Rana | Initial Draft |
| 002 | 21.11.2023 | Vishakha Rana | Final Draft |
|  |  |  |  |

1. [INTRODUCTION 2](#_TOC_250010)
   1. [SCOPE 2](#_TOC_250009)
      1. [In Scope 2](#_TOC_250008)
      2. [Out of Scope 2](#_TOC_250007)
   2. [QUALITY OBJECTIVE 3](#_TOC_250006)
   3. [ROLES AND RESPONSIBILITIES 3](#_TOC_250005)
2. [TEST METHODOLOGY 3](#_TOC_250004)
   1. [OVERVIEW 3](#_TOC_250003)
   2. [TEST LEVELS 4](#_TOC_250002)
   3. [TEST COMPLETENESS 4](#_TOC_250001)
3. [TEST DELIVERABLES 4](#_TOC_250000)
4. RESOURCE & ENVIRONMENT NEEDS 5
   1. TESTING TOOLS 5
   2. TEST ENVIRONMENT 5
5. TERMS/ACRONYMS 6

# Introduction

In the realm of assistive technologies, Sign Language Recognition Systems play a pivotal role in bridging communication gaps for those with hearing impairments. This report examines a cutting-edge system that leverages computer vision and machine learning. Using a camera-based approach, it captures and interprets sign language gestures, translating them into spoken language.

This report focuses on evaluating the Sign Language Recognition System's functionality, accuracy, and overall performance. Through advanced algorithms, the system recognizes a diverse range of sign language gestures, associating each with its corresponding letter and converting it into audible speech. This technological advancement has significant potential to enhance communication for the deaf and hard-of-hearing community, fostering inclusivity and accessibility.

## Scope

### In Scope

Scope defines the features, functional or non-functional requirements of the software that **will be** tested. Features of the Project:

Scope of the Sign Language

Gesture Recognition: The system captures and interprets sign language gestures through a camera-based approach.

Letter Binding: Recognition algorithms associate each recognized gesture with its corresponding letter, forming the basis for subsequent speech conversion.

Speech Conversion: Translates recognized sign language gestures into audible speech, enabling effective communication.

Real-time Processing: Ensures timely and responsive recognition and conversion, facilitating dynamic and interactive communication.

User Interface: A user-friendly interface for ease of interaction and integration into various devices.

2. Functional Requirements:

Gesture Recognition Module:

- Must accurately detect and classify a diverse range of sign language gestures.

- Should operate in real-time to provide instantaneous feedback.

- Must be adaptable to different lighting conditions for robust performance.

Letter Binding Module:

- Must associate each recognized gesture with the correct corresponding letter.

- Should account for variations in signing styles and regional differences.

Speech Conversion Module:

- Must convert recognized gestures into clear and coherent audible speech.

- Should support multiple languages for enhanced accessibility.

Real-time Processing:

- Must process gestures and convert them into speech with minimal latency.

- Should handle concurrent user interactions without compromising performance.

User Interface:

- Must provide an intuitive and user-friendly interface for seamless interaction.

- Should display feedback on recognized gestures and converted speech.

3. Non-functional Requirements:

Performance:

- The system should have high accuracy in gesture recognition and speech conversion.

- The response time for gesture recognition and speech conversion should be within acceptable limits.

Reliability:

- The system should operate reliably under varying environmental conditions.

- It should handle a reasonable level of user input variations and gestures.

Scalability:

- The system should scale efficiently to accommodate potential future updates and increased user load.

Security:

- User data and communication should be secure and encrypted to protect user privacy.

-Compatibility:

- The system should be compatible with a variety of camera devices and operating systems.

- It should integrate seamlessly with other assistive technologies and communication devices.

Usability:

- The system should provide clear feedback to users on recognized gestures and converted speech.

### Out of Scope

Out Of Scope defines the features, functional or non-functional requirements of the software that **will NOT be**

tested:

Complex Sign Language Gestures:

-Extremely intricate or highly complex sign language gestures, which fall beyond the defined set of recognized gestures, will not be tested in this phase.

Sign Language Dialects:

-Recognition and interpretation of regional or highly specific sign language dialects that deviate significantly from the standard set will not be a focus of this testing phase

Extreme Environmental Conditions:

- The system's performance under extreme environmental conditions, such as exceptionally low light or high levels of noise, will not be the primary focus of this testing phase.

## Quality Objective

Here makes a mention of the overall objective that you plan to achieve without your testing.

Explore Advanced Gestures: Conduct further research to explore the recognition of advanced sign language gestures, including complex and region-specific signs, to enhance the system's inclusivity.

Dialect Recognition: Investigate the incorporation of dialect recognition to expand the system's capability to understand and interpret various sign language variations.

Accessibility for Severe Disabilities: Investigate and implement features to enhance system accessibility for users with severe disabilities, ensuring a more inclusive user experience.

Some objectives of your testing project can be-

* Ensure the Application Under Test conforms to functional and non-functional requirements.
* Ensure the AUT meets the quality specifications.
* Bugs/issues are identified and fixed before go live.

## Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

* QA Analyst: Sanskriti Bajpai
* Test Manager: Prof. Shreela Pareek
* Configuration Manager: Prof. Neha Shukla
* Developers: Sanskriti Bajpai, Vishaka Rana, Vidhi
* Installation Team: Prof. Shreela Pareek, Prof. Neha Shukla, Sanskriti Bajpai, Vishaka Rana, Vidhi

# Test Methodology

## Overview

We are using an iterative testing approach to make sure our project works well. This means we test it in small steps, starting with checking if each part works on its own. Then, we see how different parts work together. We keep testing as we make changes and add new things. This way, we make sure our project is always working well, even after modification.

## Test Levels

**Test Levels define the Types of Testing to be executed on the Application Under Test (AUT**).

We aim to test our project at the following levels:

1. Unit Testing: This is the lowest level of testing and focuses on individual components or functions within the software. Developers often perform unit tests to verify that specific parts of the code work correctly.
2. Integration Testing: This level of testing checks how different components or modules of the software work together. It ensures that integrated parts of the software function as intended.
3. System Testing: At this level, the entire system is tested. It verifies that the software meets its specified requirements and functions properly in its intended environment.

## Test Completeness

Here you define the criteria that will deem your testing complete. For instance, a few criteria to check Test Completeness would be

* 100% test coverage
* All Manual & Automated Test cases executed.
* All open bugs are fixed or will be fixed in the next release.

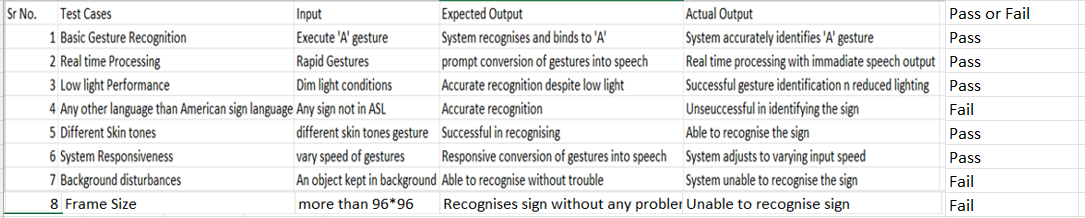
# Test Deliverables

Here are the deliverables.

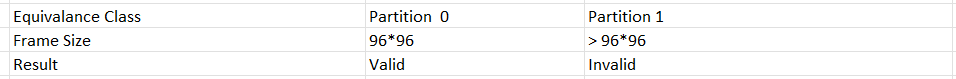
* Test Plan
* Test Cases
* Bug Reports

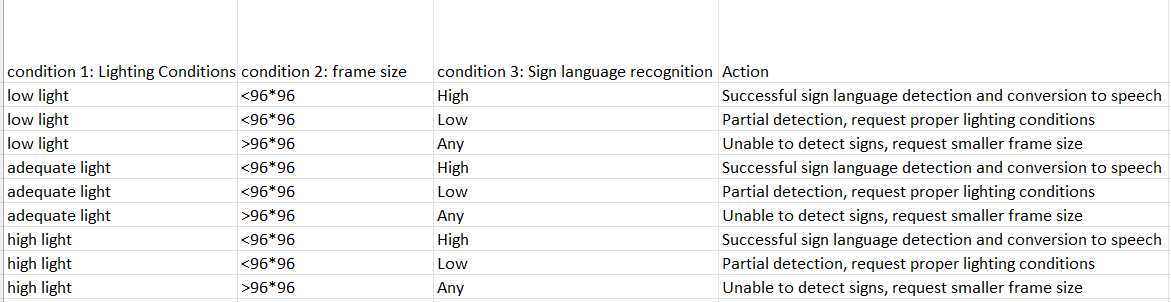
 Test Strategy

# Test Cases



**EQUIVALANCE CLASS PARTITIONING BASED ON FRAME SIZE -**



****

## Testing Tools

* Manual Testing:

## Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application. Following **software’s** are required in addition to client-specific software.

* Windows 10 and above preferred
* VSCode 2022 or above preferred
* Chrome, Mozilla or Edge Preferred over non-chromium-based browsers.
* The camera should be working properly.

.

# Terms/Acronyms

Make a mention of any terms or acronyms used in the project

|  |  |
| --- | --- |
| **TERM/ACRONYM** | **DEFINITION** |
| API | Application Program Interface |
| AUT | Application Under Test |