AUTOMATED SIGN LANGUAGE RECOGINITION SYSTEM

By

PC Rida Samee

NC Zainub Mumtaz

NC Hira Shahzad

Submitted to the Faculty of Computer Software Engineering
National University of Sciences and Technology, Islamabad in partial fulfillment
for the requirements of a B.E. Degree in Computer Software Engineering
JUNE 2013

ABSTRACT

The aim of Automated Sign Language Recognition System is to facilitate communication between normal people and hearing impaired people. C # (C-sharp) is used as Implementation Language as it requires less computational power and less memory. C # is used under .net Framework 4 as it supports System Threading and Parallel Implementation of Video.

The solution is cost effective as it does not require a Data Glove as Input Device rather it uses a Camera & Image Processing Algorithms for Tracking and Localizing Hand Postures. Image Binarization and Contour Extraction is applied to extract hand from background.

The system is trained using Supervised Learning Technique in order to recognize gestures with accuracy using Artificial Neural Networks. Aforge Library is used as it supports Artificial Intelligence and Learning Algorithms. The system is validated using Validation Data Set.

The webcam takes input gesture from live streaming video. This gesture is fed to the trained artificial neural network which compares the feature of input gesture with the trained samples and generates an output showing the alphabet recognized and the system generates speech which indicates the output gesture vocally.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent.

To our parents, without whose constant support and unstinting cooperation and assistance a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents, family and well-wishers for their admirable support. We would like to thank our supervisors Col. Dr. Fahim Arif and Dr. Imran Siddiqi, for their help and motivation throughout the course of our project. Without their help we would have not been able to accomplish anything. Furthermore, we would like to thank Misbah Munir for her kind support. They provided us with the opportunity to polish our technical skills and guided us into this area of learning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

Table Number **Page Number**

CHAPTER 1

INTRODUCTION

1. Introduction

1.1 Purpose

This project is intended to recognize Pakistan sign language which is a visual language that incorporates gestures, orientation and movement of the hands. The aim of the project is to facilitate communication between normal people and hearing impaired people. Since the latter use sign language, the objective is to bridge this gap by providing a system which will help normal entities to understand and communicate with impaired entities.

1.2 Project Background

Following successful projects are done for recognizing sign language:-

1. Boltay Hath

2. Glove Talk & Glove Talk 2

3. CANDY (Communication Assistance to Negate Disabilities in Youth)

4. Talking Glove

5. GRASP (Glove Based Recognition of Auslan using Simple Processing)

6. GIVEN (Gestures Driven Interactions in Virtual Environment)

7. NTT/ATR (Japanese Manual Alphabet Recognition System)

All of these above mentioned projects used Data Glove for Hand Tracking and Localization.

However, Data Glove provides an expensive solution to this problem.

Our solution will incorporate Image Processing techniques for hand localization & tracking.

(a) (b) (c)

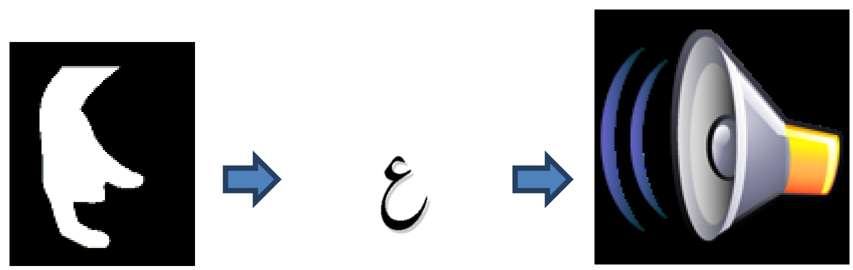


Figure 1-1: Automated Sign Language Recognition System.

(a) User performing Hand Gesture

(b) Recognized Gesture

(c) Text to Speech Conversion

1.3 Project Scope

Purpose of the system is to develop an Automated Sign Language Recognition System capable of recognizing the hand gestures in the input image. The recognized gesture will be displayed as text. The work will mainly focus on Pakistan Sign Language for Urdu. The text will be converted to speech. Only a limited range of gestures will be handled by the system. 10 gestures will be incorporated.

1.4 Objectives

1.4.1 Academic Objectives

The academic objectives include understanding and implementation of Image Processing algorithms and techniques along with the study and implementation of Artificial Neural Networks (ANN). Application Programming is done in C# under .net framework 4.

1.4.2 Application/Technical Objectives

Technical objective is to bridge the communication gap between hearing impaired and normal individuals by automating the recognition of gestures.

CHAPTER 2

LITERATURE REVIEW

2. Literature Review

2.1 Introduction

In Pakistan, the existence and prevalence of disability, the need to take preventive measures and to make curative efforts are well recognized at both the governmental and nongovernmental levels. Efforts have been made to carry out nationwide programs for the prevention of disabilities and for the rehabilitation of people with disabilities.

References:

According to 1998 census:

Table	Population (Disability-wise)				
S.No.	Disability	Male	Female	Total	As %age of total disabled
a.	Blind	146029	119369	265398	8.06
b.	**Deaf and Mute**	**138235**	**105448**	**243683**	**7.40**
c.	Crippled	382262	243523	625785	19.00
d.	Insane	119645	91209	210854	6.40
e.	Mentally Retarded	134887	115297	250184	7.60
f.	Having more than one Disability	140285	130166	270451	8.21
g.	Others	857362	568438	1426800	43.33

Figure 2-1: Disability Record 1998 census

2.2 Problem Domain

This project falls under the domain of Image Processing and Artificial Intelligence. This project enhances human computer interaction. Already implemented systems use data glove which leads to a very expensive solution. To overcome this problem we incorporated image processing algorithms to provide a cost effective solution. In terms of Image Processing fast computation of image and video frames is challenging. Next hurdle was to implement Image Processing algorithm in a parallel fashion to ensure speedy work. The domain is itself newly emerging since no gesture recognition has been done using Image Processing.

2.3 Related Work

Previously a project was done named Pakistan Sign Language Recognition using Statistical Template Matching (Boltay Hath) by a group of students from Sir Sayed University, Karachi, Pakistan.

The basic components of the Boltay Haath system are given below:

Modules for Gesture Input – Get state of hand (position of fingers, orientation of hand) from glove and convey to the main software.

Gesture Preprocessing Module – Convert raw input into a process-able format for use in pattern matching. In this case, scaled integer values ranging from 0 to 255.

Gesture Recognition Engine – Examines the input gestures for match with a known gesture in the gesture database.

Gesture Database - Contains the necessary information required for pattern matching as well as a gesture-to-text dictionary.

Speech Synthesis Module – Converts word / letters obtained after gesture analysis into corresponding sound.

Following successful projects are done for recognizing sign language:-

1. Glove Talk & Glove Talk 2
2. CANDY (Communication Assistance to Negate Disabilities in Youth)
3. Talking Glove
4. GRASP (Glove Based Recognition of Auslan using Simple Processing)
5. GIVEN (Gestures Driven Interactions in Virtual Environment)
6. NTT/ATR (Japanese Manual Alphabet Recognition System)

2.4 Shortcomings/issues

All of these above mentioned projects used Data Glove for Hand Tracking and Localization. However Data Glove provides an expensive solution to this problem.

This solution will incorporate Image Processing techniques for Hand Tracking and Localization.

2.5 Proposed Project

Our solution is projected to recognize sign language that incorporates gestures**,** orientation and movement of the hands**.** Hand movements will be provided to the system**.** Those movements will then be interpreted by the software and based on those gestures the system will translate it into text (Urdu language). Based on performance, optimality and time constraints the project might be extended up to 10-12 gestures.

2.6 Technological requirements

.net Framework 4

The software system can run on windows platform via Visual Studio 2010 using .net framework 4. The library used is Aforge.

Camera Based Hand Tracking and Localization

The solution is cost effective as it does not require a Data Glove as Input Device rather it uses a Camera & Image Processing Algorithms for Tracking and Localizing Hand Postures.

Artificial Intelligence Supervised Learning Technique

The system will be trained using Supervised Learning Technique in order to recognize gestures with accuracy.

Validation of System

The system will be validated using Validation Data Set and will be refined further by parameter tuning.

Standard Recall Measure

The overall tuned system will be evaluated using Standard Recall Metric.

Artificial Neural Network

System will be trained using neural network. 250 samples for 10 gestures will be provided for training.

Parallel implementation of video manipulation

In order to meet system real time constraints video manipulation was done in parallel fashion.

2.7 Software Requirements

The software(s) required for the implementation of our project includes:

Visual Studio 2010 under .net framework 4

Aforge library

2.8 Hardware Requirements

The Hardware required for the implementation of our project includes:

Personal computer

Webcam

Speaker

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

3. Introduction

3.1 Purpose

The purpose of this section is to describe the software requirements for developing the Automated Sign Language Recognition System. Working & features that are present in the system along with the constraints over them are explained.

3.2 Document Conventions

The format of this document is simple. Bold face indicates headings.

3.3 Intended Audience and Reading Suggestions

This document is intended to be read by: -

Supervisor

Team members

Valuable evaluators/faculty members of **Military College of Signals.**

3.4 Product Scope

This project is intended to recognize Pakistan sign language which is a visual language that incorporates gestures, orientation and movement of the hands. The aim of the project is to facilitate communication between normal people and hearing impaired people. Since the latter use sign language, the objective is to bridge this gap by providing a system which will help normal entities to understand and communicate with impaired entities.

3.5 References

NA

3.6 Overall Description

3.6.1 Product Perspective

The Automated Sign Language Recognition System is new, self-contained product.

Hand movements will be provided to the system. Those movements will then be interpreted by the software and based on those gestures the system will display the output in the form of text and speech.

3.6.2 Product Functions

3.6.2.1 Gesture Recognition

Based on the input the system will recognize the gesture.

3.6.2.2 Textual Output

The recognized gesture will be translated to the textual output.

3.6.2.3 Speech Output

The textual output will be converted to speech.

3.7 User Classes and Characteristics

The product will be used by potentially one group of actors (users):

User:

The user will provide hand gestures to the system with the help of the camera. Those hand gestures will be translated into text & speech as the output.

3.5.1 Operating Environment

The software can be run on windows platform. It will be developed using Microsoft Visual Studio 2010 in C# under .net framework 4.

3.5.2 Design and Implementation Constraints

.net Framework

C # will be used under .net Framework 4 as it supports data base handling, web development, numeric algorithm application, machine learning etc.

Camera Based Hand Tracking

The solution is cost effective as it does not require a Data Glove as input device rather it uses a camera & Image Processing Algorithms for Tracking and Localizing Hand Postures.

Parallel implementation of Video Manipulation

Steps of the algorithm have been chosen to be implemented in a parallel form to meet the systems real time constraints.

3.5.3 User Documentation

There is no user documentation.

3.5.4 Assumptions and Dependencies

The module for Hand Tracking & Localization is taken from open source.

3.6 External Interface Requirements

3.6.1 User Interfaces

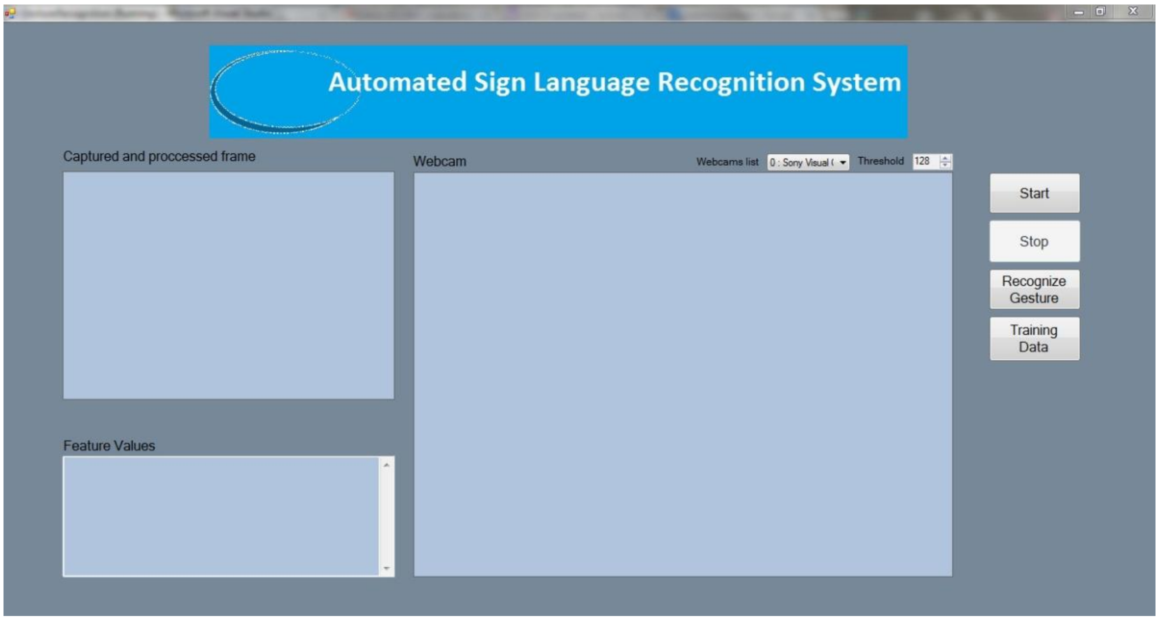


Figure 3-1: Interface of Automated Sign Language Recognition System

An overview of the graphical interface can be seen in figure and contains the following sections:

Webcam list: Displays the video capture devices detected on the system by AForge .net.

Start and stop buttons: Are used for starting and stopping the acquisition process.

Threshold: The control is used for setting the threshold for the binarization process permitting the user to accommodate for different conditions.

Webcam: The frames captured from the webcam, binarized and filtered are shown in that area. The tracked hand is shown in this area.

Captured and processed frame: The area is used for displaying the current captured frame and the contour is highlighted. The frames are captured at a rate of one frame per second. The denoised image is also shown here.

Recognize gesture: The button is used to recognize the current gesture.

Feature Values: The text box is used to display feature values of the current gesture.

Training Data: This button is used to generate training data for new gestures.

3.6.2 Hardware Interfaces

The software will interact with the Integrated Webcam (hardware) of the machine for receiving Input Video. The software can operate on any machine that has a webcam integrated with it.

3.6.3 Software Interfaces

The software system can run on windows platform via Visual Studio 2010 using .net framework 4. The library is Aforge.

3.6.4 Communication Interfaces

NA.

3.7 System Features

The software shall provide the following features:

3.7.1 Reception of Input Video

3.7.1.1 Description and Priority

The working of the system highly depends upon this feature. The system will receive input video via an integrated Webcam.

Priority: high

Benefit: 8

Penalty: 7

Cost: 7

Risk: 4

3.7.1.2 Stimulus/Response Sequences

User Action -> User will select the camera from the list of cameras & click “start” button.

System Response -> Primary Scenario: System will receive the video.

Secondary Scenario: In case system is unable to receive video, error message will be displayed along with help.

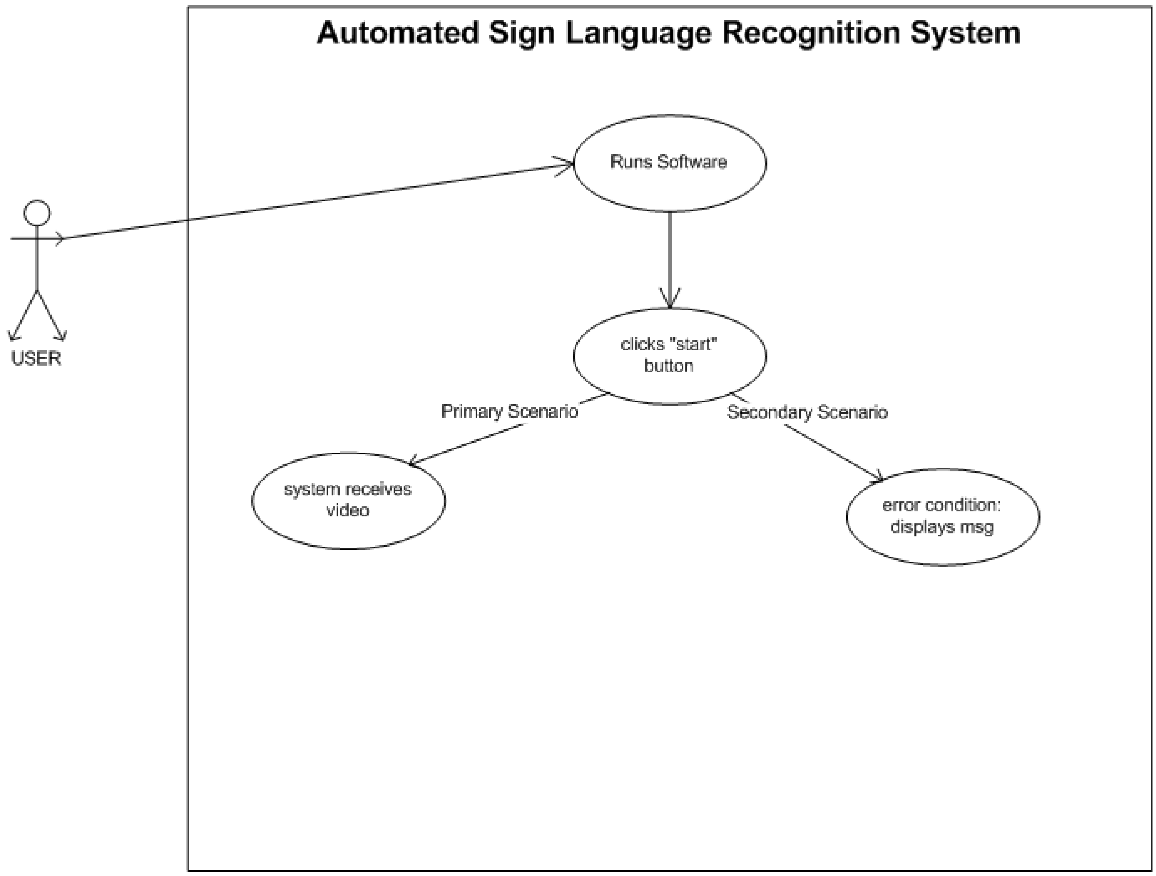
3.7.1.3Functional Requirements

REQ-1: To carry out the services provided by this feature the webcam in working condition shall be integrated with the software.

REQ-2: Camera shall receive the video.

REQ-3: On error conditions or invalid inputs the software shall display error message and shall help the user to use the software efficiently by providing help.

3.7.1.4 Use case diagram



Use Case 3-1: Reception of Input Video

3.7.2 Hand Tracking & Localization

3.7.2.1 Description and Priority

The working of the system highly depends upon this feature. The system will receive input video via an integrated Webcam. Parallel pixel manipulation will be done in order to extract hand from the background.

Priority: high

Benefit: 8

Penalty: 7

Cost: 7

Risk: 7

3.7.2.2 Stimulus/Response Sequences

Precondition: The system is receiving input video successfully.

User Action -> User will perform hand gestures in front of the camera.

System Response ->

Primary Scenario: System will receive the video and will extract the hand from the background.

Secondary Scenario: Incase system is unable to track hand, message will be displayed.

3.7.2.3 Functional Requirements

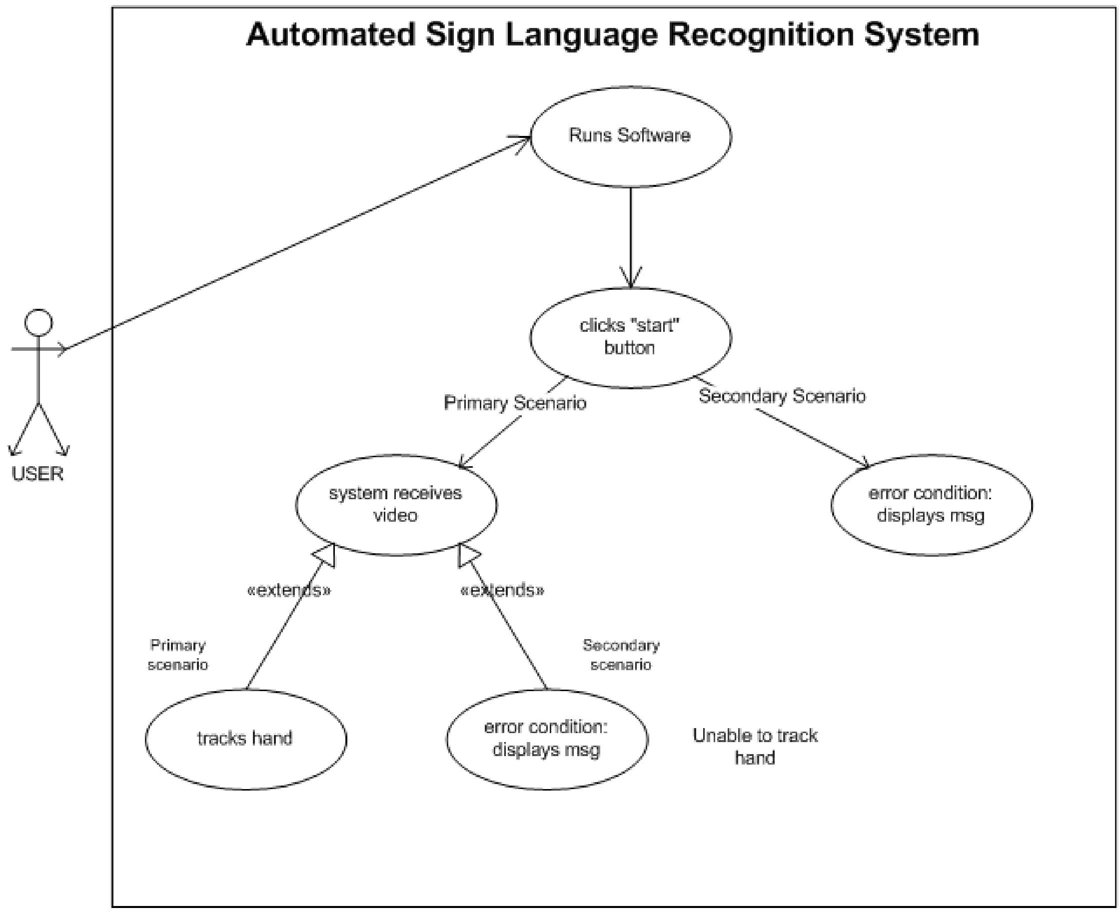
REQ-1: To carry out the services provided by this feature the webcam in working condition shall be integrated with the software.

REQ-2: Camera shall receive the video.

REQ-3: The hand shall be extracted from the background.

REQ-4: On error conditions or invalid inputs the software shall display error message and shall help the user to use the software efficiently by providing help.

3.7.2.4 Use Case Diagram



Use Case 3-2: Hand Tracking & Localization

3.7.3 Gesture Recognition

3.7.3.1 Description and Priority

The working of the system highly depends upon this feature. The system will perform algorithmic manipulations on the hand orientation extracted from the video already received via an integrated Webcam.

Priority: high

Benefit: 9

Penalty: 8

Cost: 8

Risk: 8

3.7.3.2 Stimulus/Response Sequences

Precondition: The hand is extracted from the background.

User Action -> User will perform hand gestures in front of the camera.

System Response -> System will receive the video and parallel pixel manipulation will be done in order to extract the hand orientation.

Primary Scenario: Based on hand orientation the gesture will be recognized.

Secondary Scenario: Incase system is unable to recognize gesture, message will be displayed.

3.7.3.3 Functional Requirements

REQ-1: To carry out the services provided by this feature the webcam in working condition shall be integrated with the software.

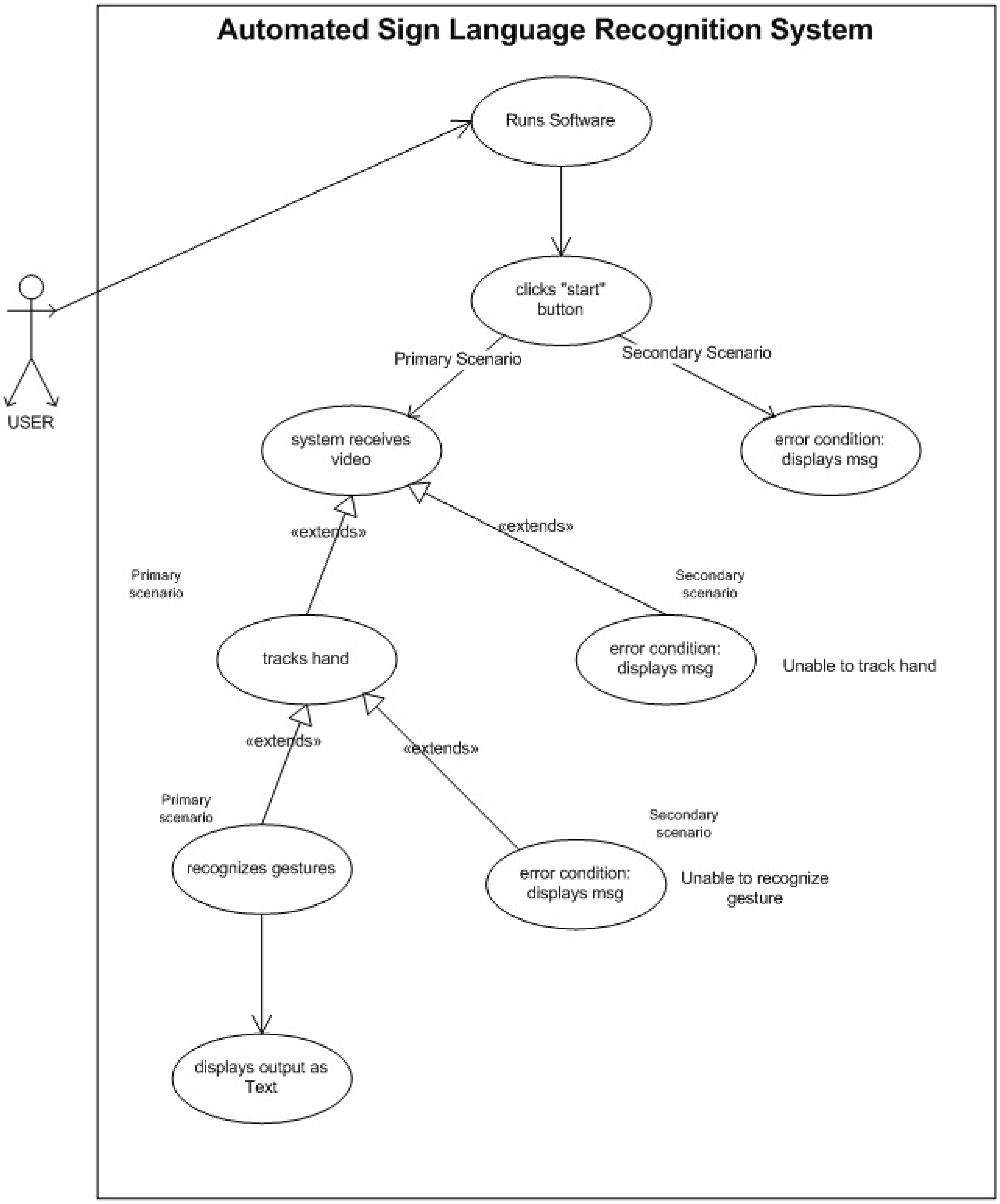
REQ-2: Camera shall receive the video.

REQ-3: The hand shall be extracted from the background.

REQ-4: Valid hand gesture will be recognized.

REQ-5: On error conditions or invalid inputs the software will display error message and will help the user to use the software efficiently by providing help.

3.7.3.4 Use Case Diagram



Use Case 3-3: Gesture Recognition

3.8 Other Nonfunctional Requirements

3.8.1 Performance Requirements

The software shall generate output based on the real time video. The streaming and video manipulation shall not degrade the system performance. The system shall generate output within a reasonable time delay. The system performance can be improved via parallel manipulation on video/image pixels.

3.8.2 Safety Requirements

NA

3.8.3 Security Requirements

NA

3.8.4 Software Quality Attributes

Maintainability: In order meet new requirements of increasing the number of gestures the system can be easily maintained.

Usability: The system shall be user friendly and easy to use.

Correctness of Algorithms: Algorithms shall be correct with respect to specification. For each input, correct output shall be produced.

Reliability: The system shall perform and maintain its functions in routine and in unexpected circumstances.

3.8.5 Business Rules

NA

3.9 Other Requirements

Standard of Sign language: The software shall incorporate Pakistan Sign Language which is adopted all over Pakistan.

Reuse objectives: The software shall have the ability to be reused & extended further by increasing the number of gestures.

CHAPTER 4

SYSTEM DESIGN SPECIFICATION

4. Architectural Design

4.1 System Block Diagram

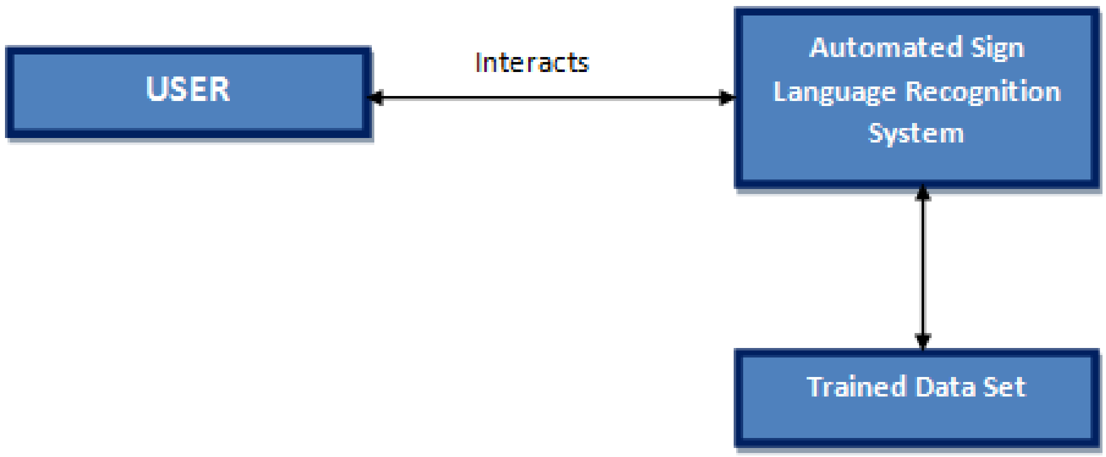


Figure 4-1: System Block Diagram

Before going into the detail of architectural design we here present logical division of project into subparts and different useful views.

4.2 Software Components

The project contains following software components:

Operating Systems

Windows XP/ Windows 7/Vista

Software Packages

Visual Studio 2010 SDK (Software Development Kit)

C# under .net framework 4

4.3 Hardware Components

Personal Computer(s)

Webcam

Speaker

4.4 High Level Design Diagram (Modules Identification)

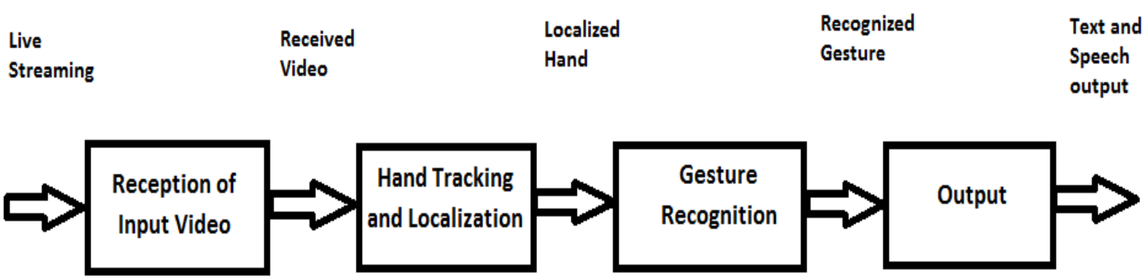


Figure 4-2: High Level Design

4.5 Interaction among Modules (Abstract View)

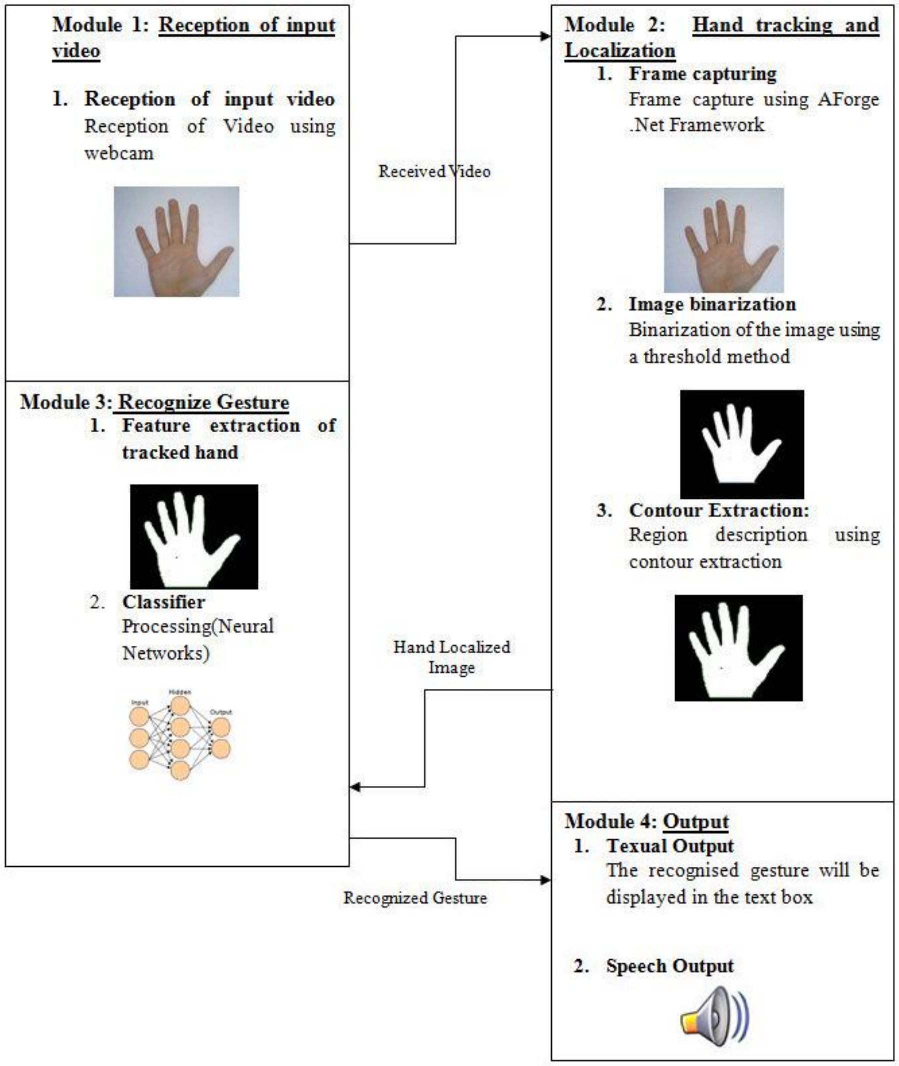


Figure 4-3: Interaction among Modules- An Abstract View

4.6 Architectural Style

The architecture of Automated Sign Language Recognition System can be modeled using Pipeline. This pattern is used because data flows through a sequence of tasks or stages. It represents a "pipelined" form of concurrency. In this architecture, there are stages and pipes connecting the stages. Stages are the modules of the system that are joined by the pipes. This shows that operations are performed sequentially. Output of one stage is the input to the next stage. There are four stages or modules in this system; Reception of Input Video, Hand Tracking and Localization, Recognize Gesture, Output. In first stage, Reception of input video, input video is received using a camera. This received video will be input for the next stage, which is Hand Tracking and Localization, Hand will be tracked and localized and the output goes to the next stage through pipeline. The next stage is Recognize Gesture. In this stage, classifier will be used to recognize the performed gesture. Output will be correctly identified gesture. This output will go to Output stage which will display the recognized gesture in the form of text and speech.

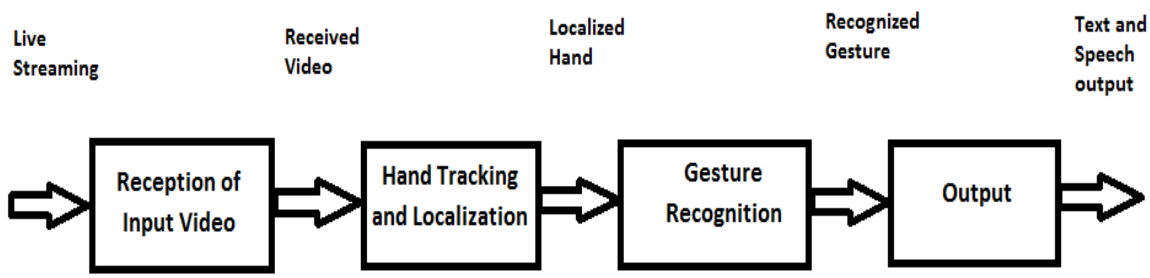


Figure 4-4: Pipeline – Architectural Style

4.7 Detailed Design

4.7.1 Database Diagram

4.7.1.2 Entity Relationship Diagram

Automated Sign Language Recognition System requires trained data set which contains collection of images on which training is done so that each gesture can be recognized. This training is done on a limited number of gestures so recognition can also be done only on these gestures. Extracted hand postures from the captured image are recognized according to this data set.

ERD of the system is given below

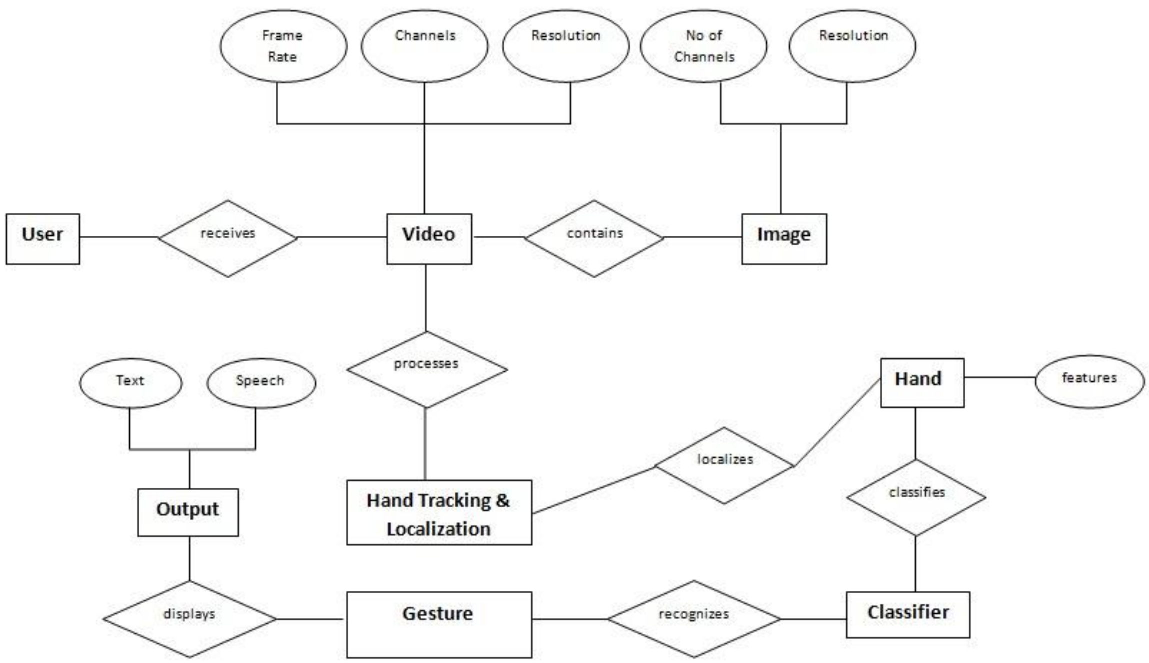


Figure 4-5: Entity Relationship Diagram

Entities & Attributes

<u>User</u>

<u>Video</u>

Resolution (640x480 pixels)

Frame Rate

Channels

Image

Resolution

No of channels

Hand

Features

Hand Tracking & Localization

Classifier

Gesture

Output

Text

Speech

4.8 UML Diagrams

4.8.1 Usecase Diagrams

4.8.1.1 Usecase Specifications

4.8.1.1.1 Use Case # 1: Reception of Input Video

a) Description and Priority

The working of the system highly depends upon this feature. The system will receive input video via an integrated Webcam.

Priority: high

Benefit: 8

Penalty: 7

Cost: 7

Risk: 4

b) Stimulus/Response Sequences

User Action -> User will select the camera from the list of cameras & click “start” button.

System Response ->

Primary Scenario: System will receive the video.

Secondary Scenario: In case system is unable to receive video, error message will be displayed along with help.

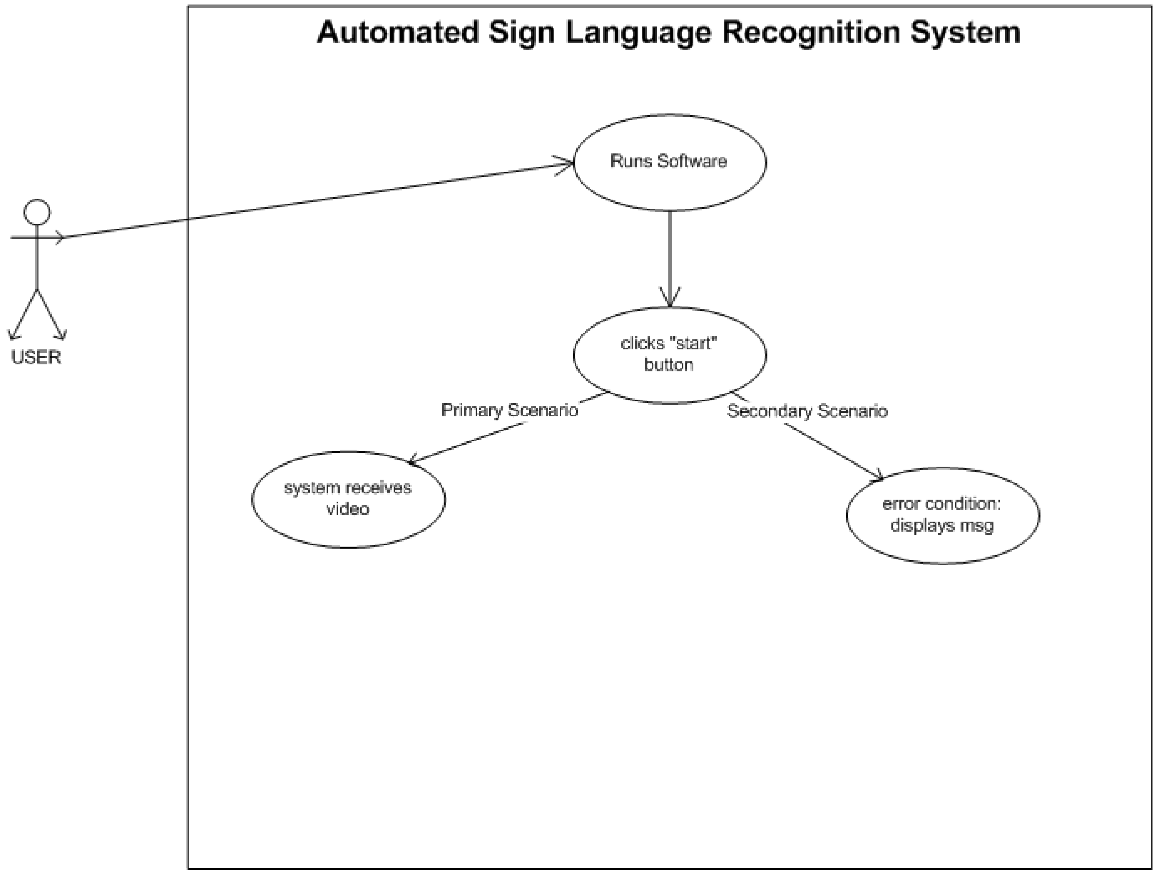
c) Functional Requirements

REQ-1: To carry out the services provided by this feature the webcam in working condition shall be integrated with the software.

REQ-2: Camera shall receive the video.

REQ-3: On error conditions or invalid inputs the software shall display error message and shall help the user to use the software efficiently by providing help.

d) Use Case Diagram



Use Case 4-1: Reception of input video

4.8.1.1.2 Use Case # 2: Hand Tracking & Localization

a) Description and Priority

The working of the system highly depends upon this feature. The system will receive input video via an integrated Webcam. Parallel pixel manipulation will be done in order to extract hand from the background.

Priority: high

Benefit: 8

Penalty: 7

Cost: 7

Risk: 7

b) Stimulus/Response Sequences

Precondition: The system is receiving input video successfully.

User Action -> User will perform hand gestures in front of the camera.

System Response ->

Primary Scenario: System will receive the video and will extract the hand from the background.

Secondary Scenario: Incase system is unable to track hand, message will be displayed.

c) Functional Requirements

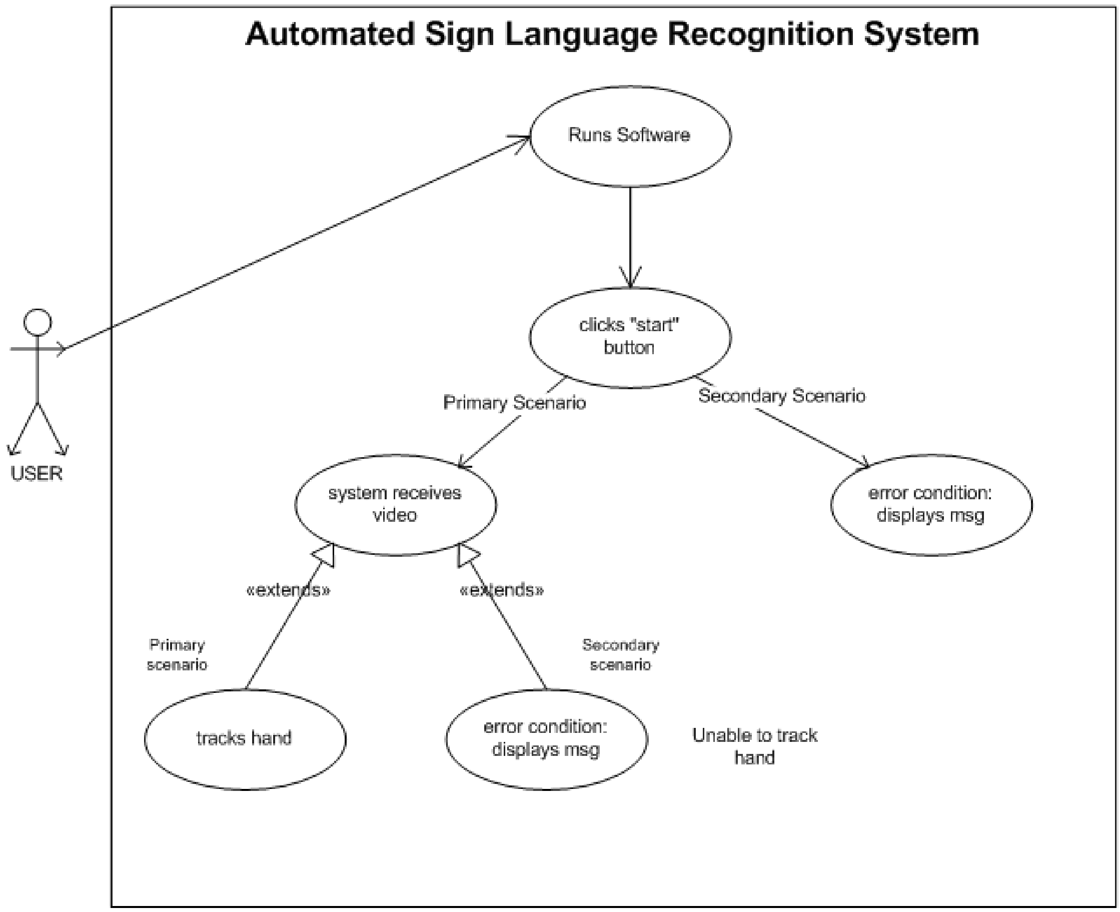
REQ-1: To carry out the services provided by this feature the webcam in working condition shall be integrated with the software.

REQ-2: Camera shall receive the video.

REQ-3: The hand shall be extracted from the background.

REQ-4: On error conditions or invalid inputs the software shall display error message and shall help the user to use the software efficiently by providing help.

d) Use Case Diagram



Use Case 4-2: Hand Tracking & Localization

4.8.1.1.3 Use case# 3: Gesture Recognition

a) Description and Priority

The working of the system highly depends upon this feature. The system will perform algorithmic manipulations on the hand orientation extracted from the video already received via an integrated Webcam.

Priority: high

Benefit: 9

Penalty: 8

Cost: 8

Risk: 8

b) Stimulus/Response Sequences

Precondition: The hand is extracted from the background.

User Action -> User will perform hand gestures in front of the camera.

System Response -> System will receive the video and parallel pixel manipulation will be done in order to extract the hand orientation.

Primary Scenario: Based on hand orientation the gesture will be recognized.

Secondary Scenario: Incase system is unable to recognize gesture, message will be displayed.

c) Functional Requirements

REQ-1: To carry out the services provided by this feature the webcam in working condition shall be integrated with the software.

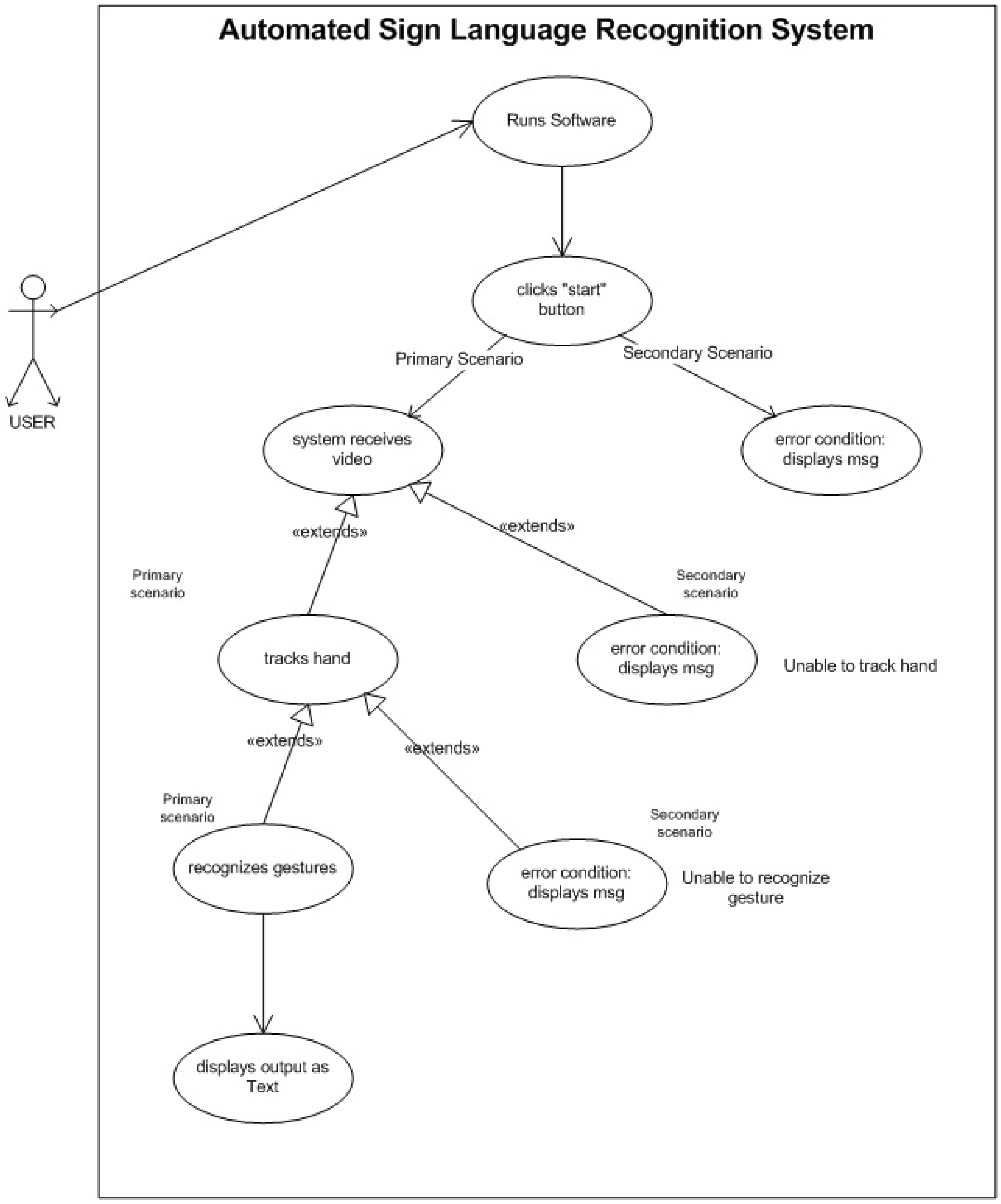
REQ-2: Camera shall receive the video.

REQ-3: The hand shall be extracted from the background.

REQ-4: Valid hand gesture will be recognized.

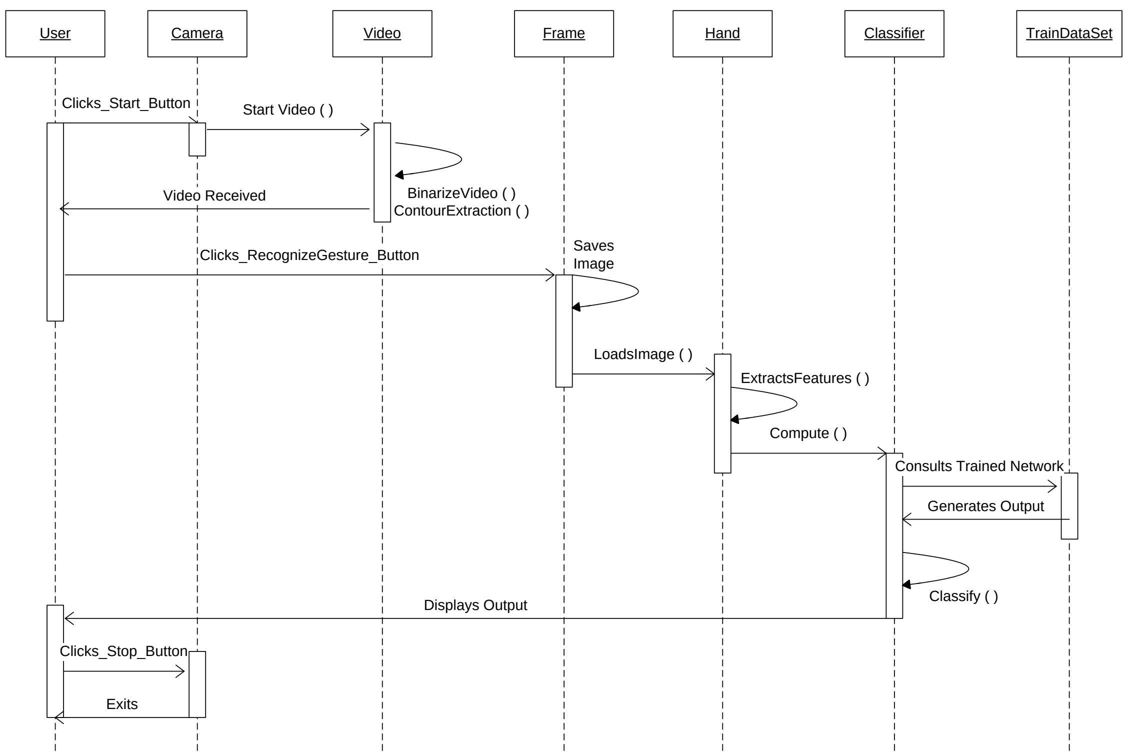
REQ-5: On error conditions or invalid inputs the software will display error message and will help the user to use the software efficiently by providing help.

d) Use Case Diagram

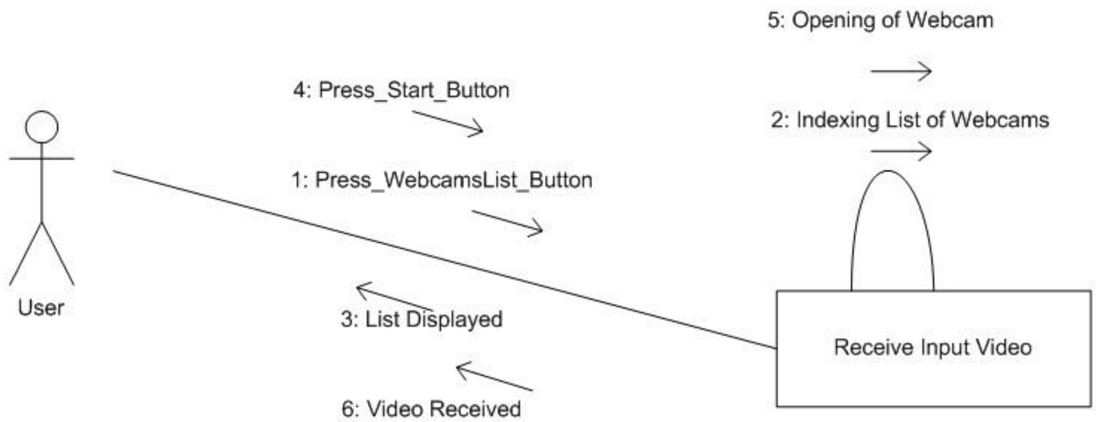


Use Case 4-3: Gesture Recognition

4.8.2 Sequence Diagram of Automated Sign Language Recognition System

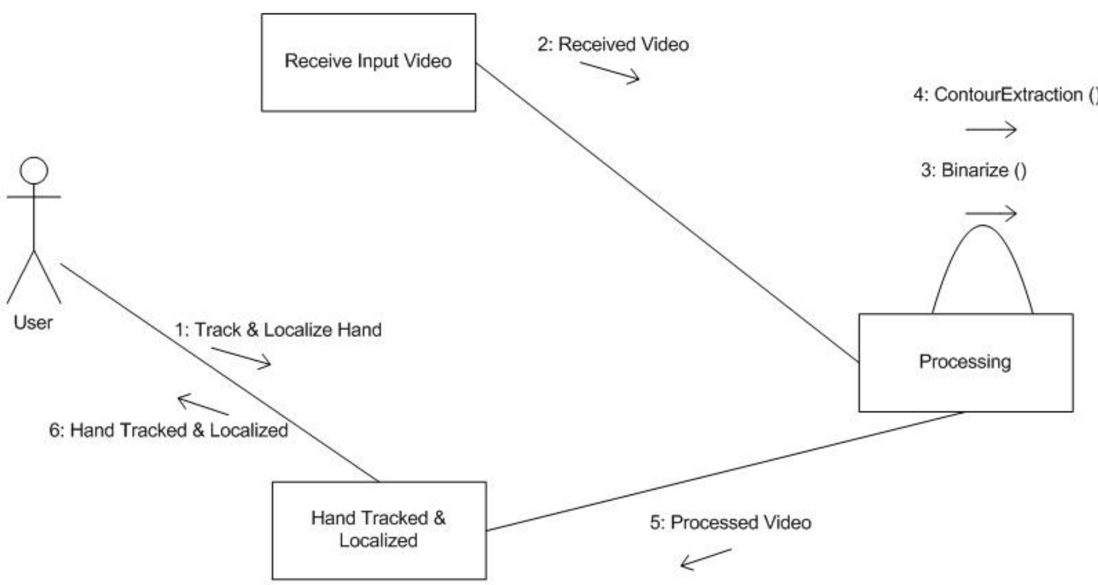


Sequence Diagram 4-1: Automated Sign Language Recognition System



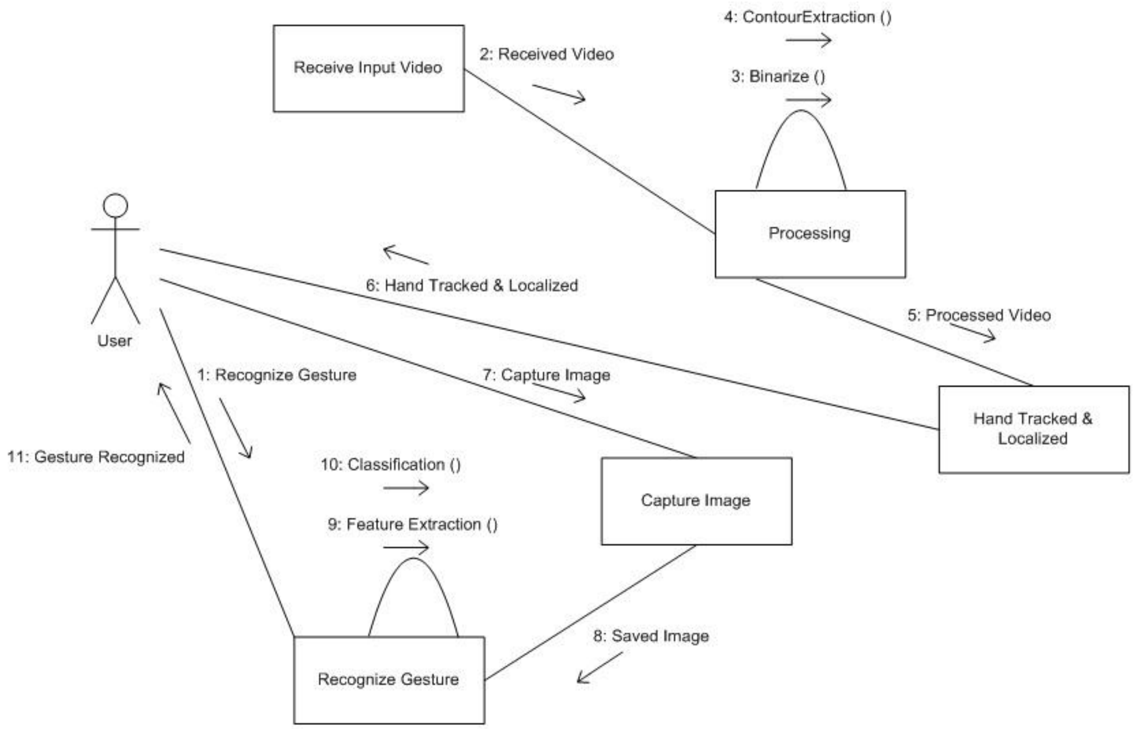
Collaboration Diagram 4-1: Input Video

a) Use case: Hand Tracking & Localization



Collaboration Diagram 4-2: Hand Tracking & Localization

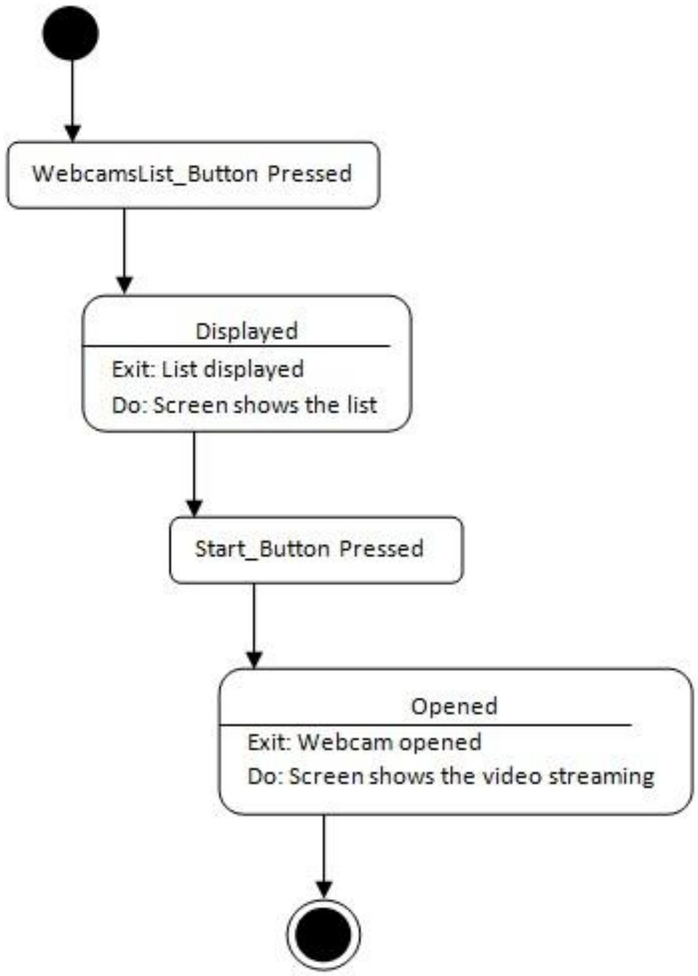
b) Use case: Gesture Recognition



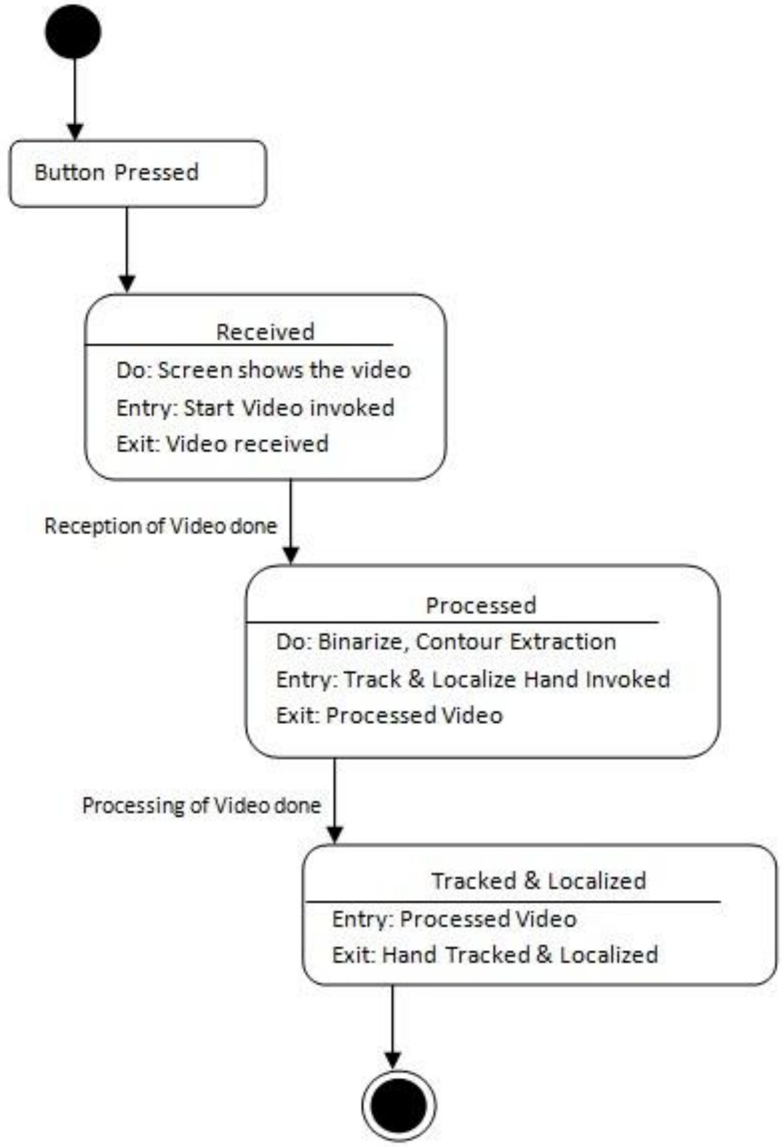
Collaboration Diagram 4-3: Gesture Recognition

4.9 Logical View

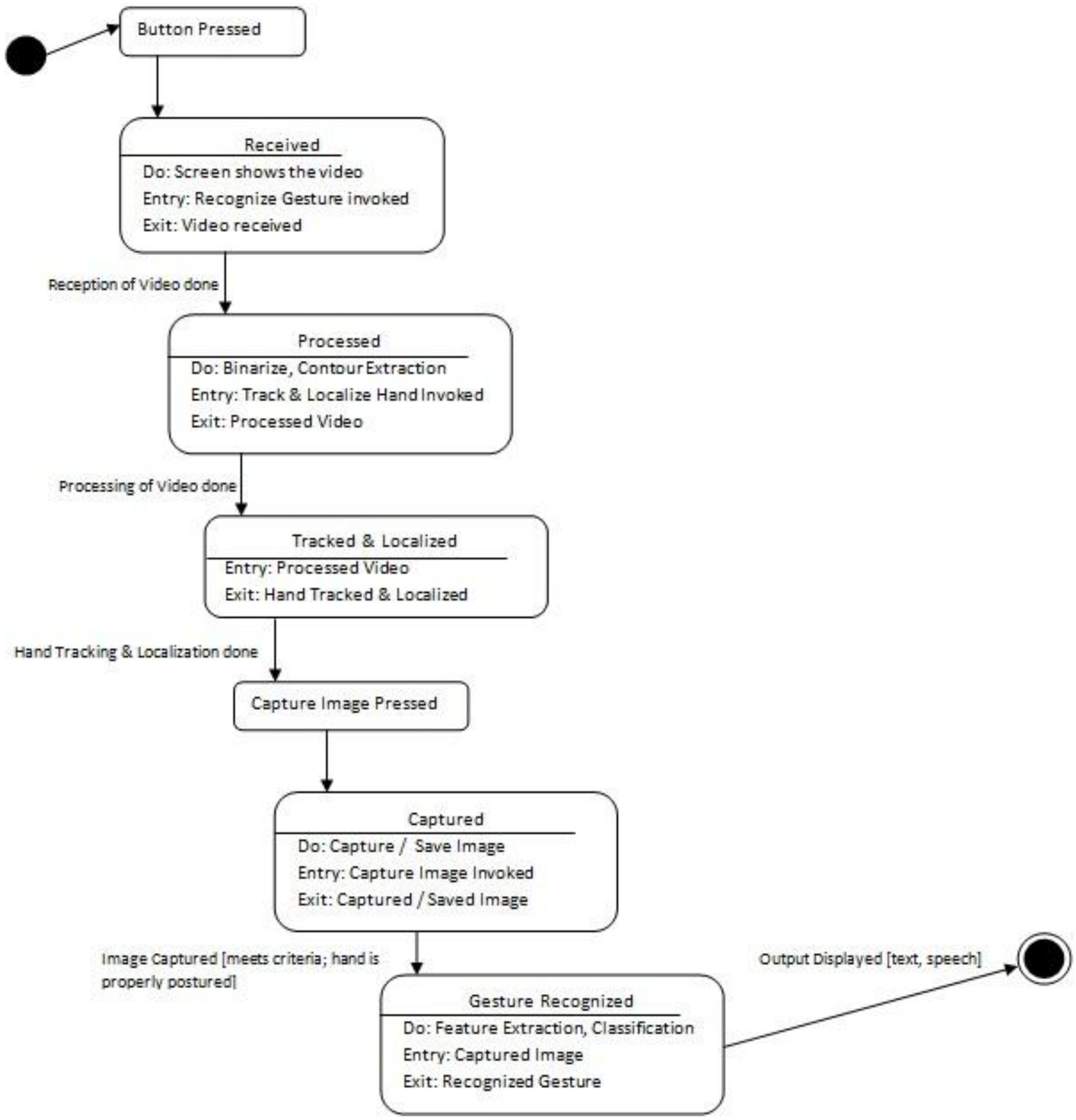
4.9.1 State Transition Diagrams



State Transition Diagram 4-1: Reception of Input Video



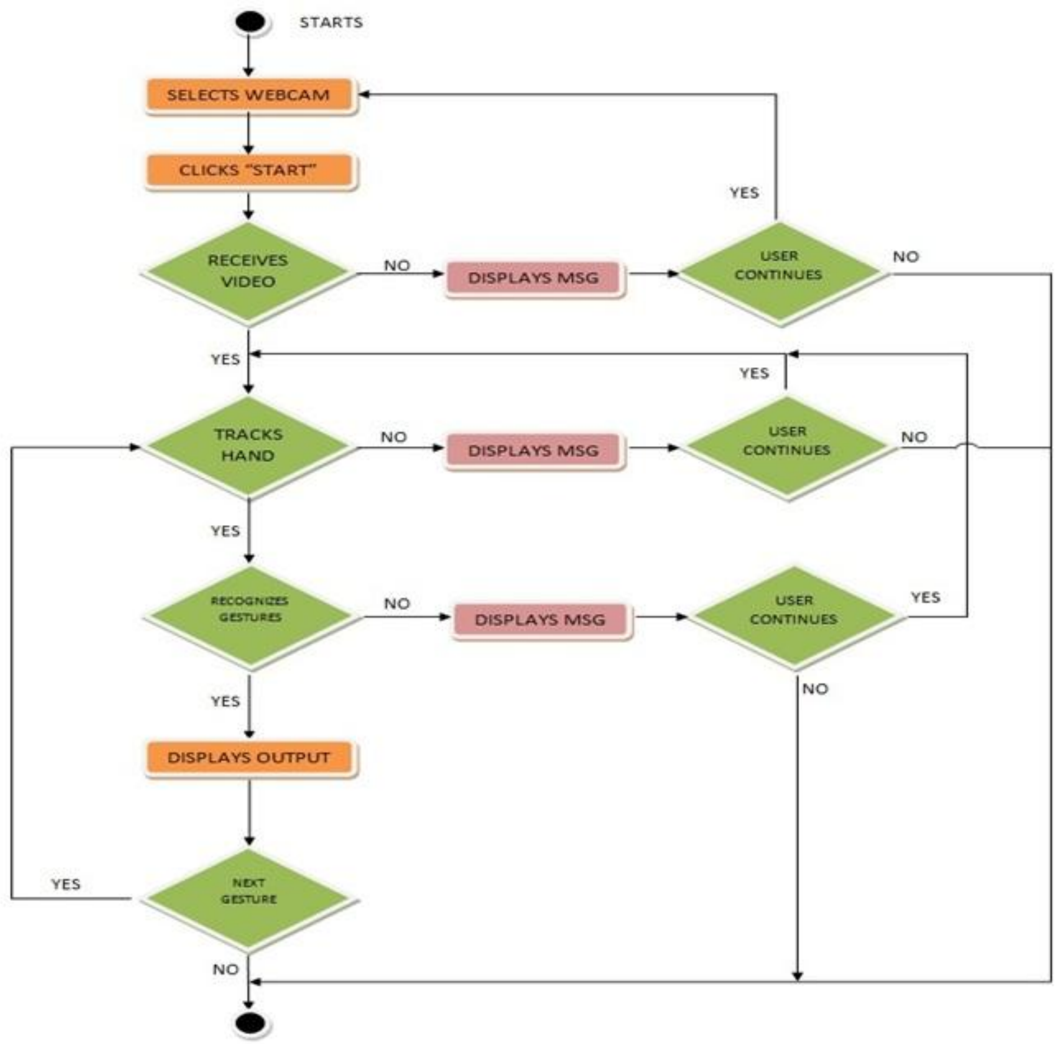
State Transition Diagram 4-2: Hand Tracking & Localization



State Transition Diagram 4-3: Gesture Recognition

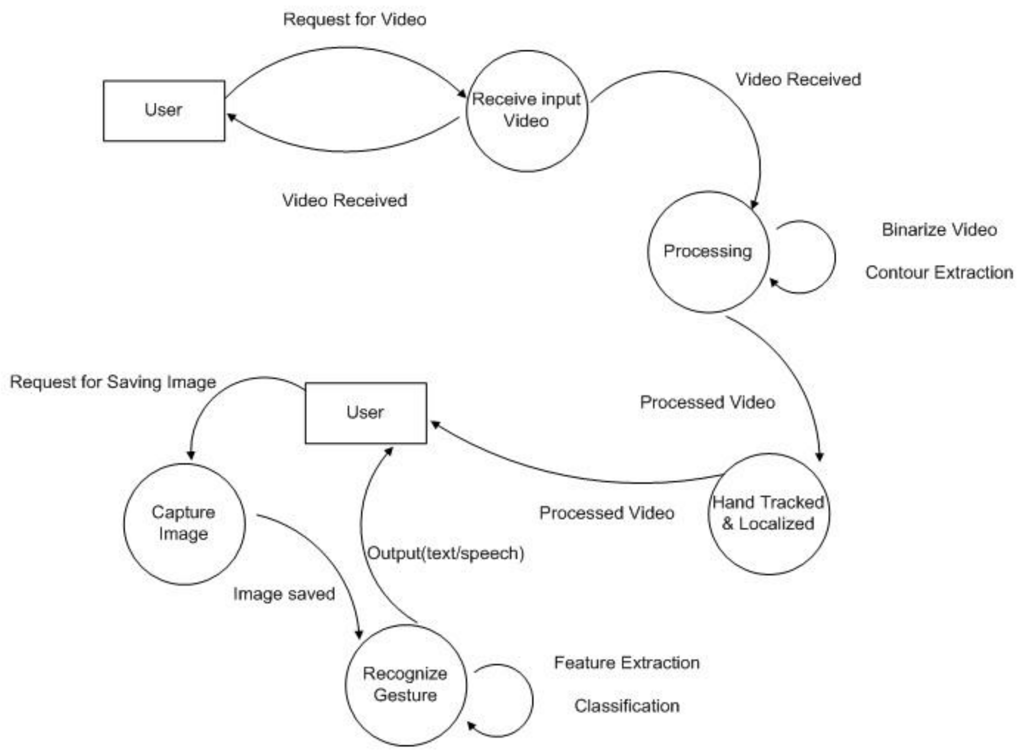
4.10 Dynamic View

4.10.1 Activity diagram



Activity Diagram 4-1

4.10.2 Data Flow Diagram



Data Flow Diagram 4-1

4.11 Implementation View

4.11.1 System Class Diagram

Gesture recognition system comprises of following classes:

User <<interface>>

Camera

Video

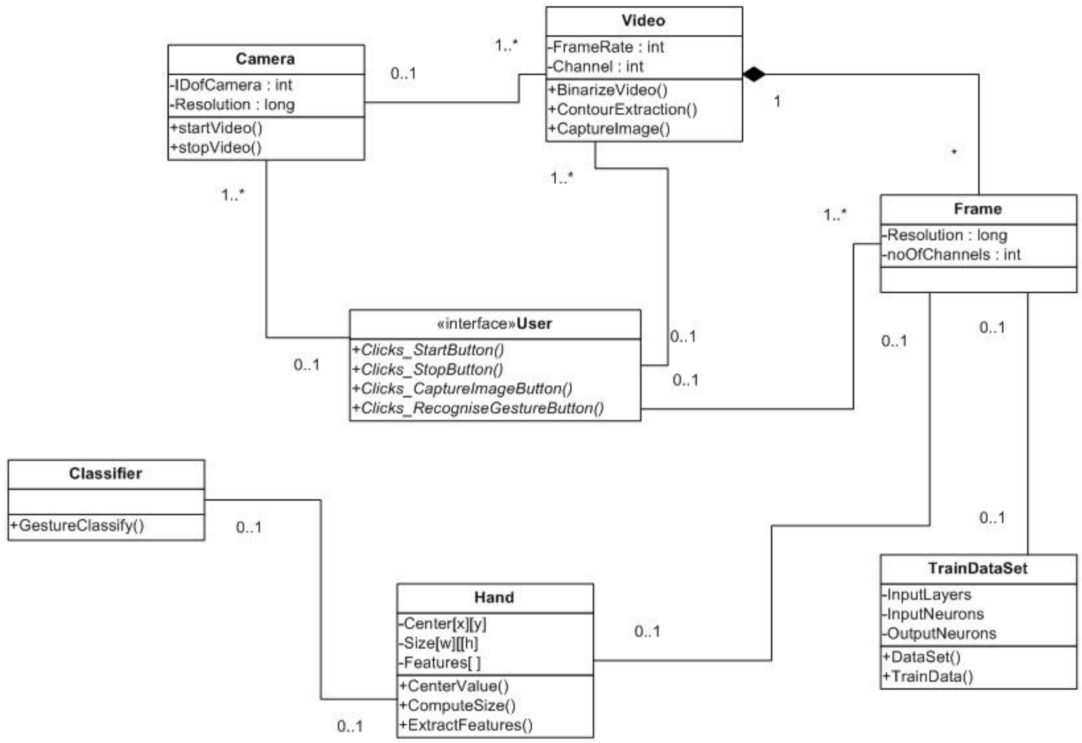
Frame

Hand

Classifier

TrainDataSet

Class Diagram is given below



Class Diagram 4-1

4.11.1.1System Classes Description

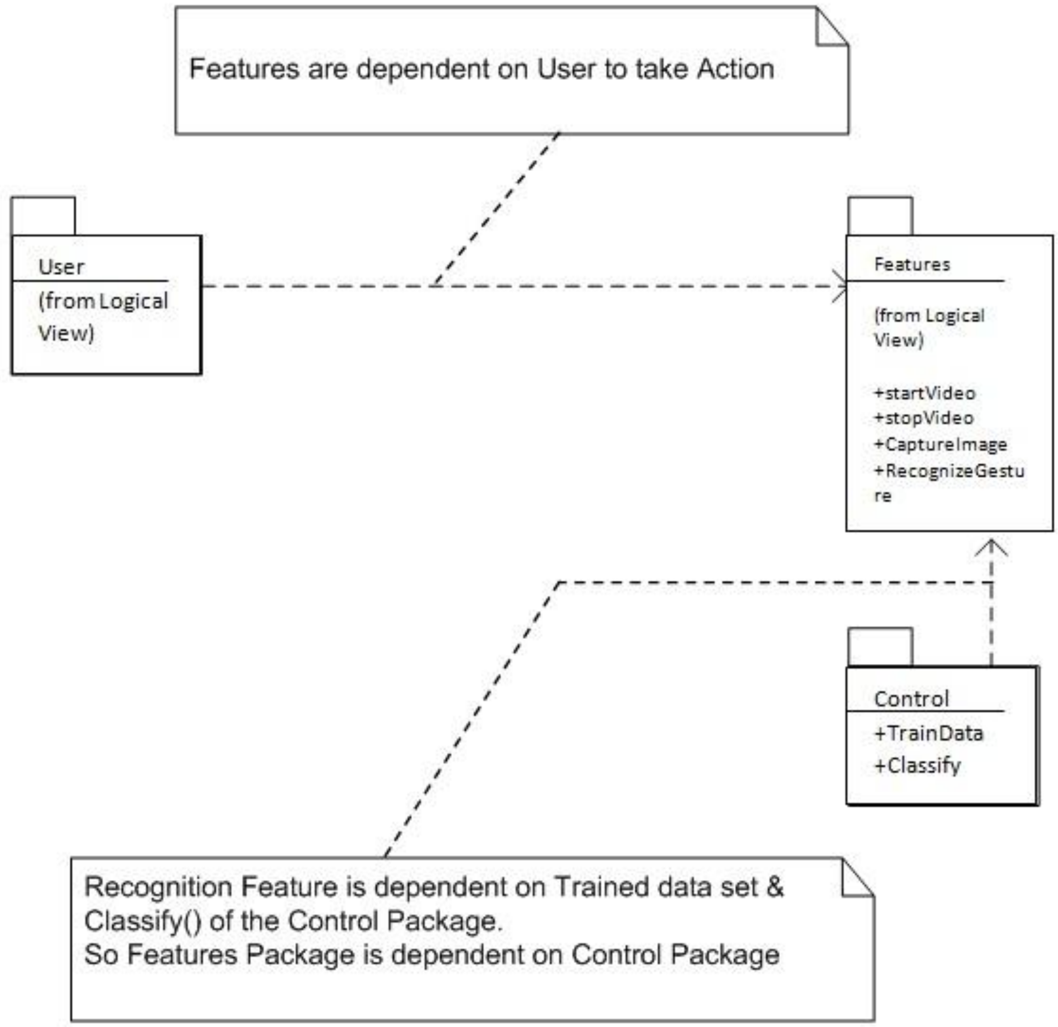
Name	Description
User **<<interface>>**	This class contains the functions which are directly related to the UI.Clicks_StartButton() is called when an external user presses the button

	Start().This function then invokes the functions in the Camera class which processes the request. Clicks_StopButton() is called when an external user presses the button Stop().This function then invokes the functions in the Camera class which processes the request. Clicks_CaptureImageButton() is called when an external user presses the button Capture Image().This function then invokes the functions in the Video class which processes the request. Clicks_RecognizeGestureButton() is called when an external user presses the button Recognize Gesture().This function then invokes the functions in the Frame class which processes the request.
Camera	This class contains startVideo( ) and stopVideo( ).This class is solely responsible for the receiving input video.
Video	The main role of this class is to process the Video. Processing is done by Binarize( ),ContourExtraction() ,CaptureImage( ). This processing is done so that the video meets the criteria for further processing. It is composed of Frame class as on video is composed of many frames.

Frame	This class uses the processed image from the Video class for processing. It passes the Image to Hand class for furthur processing.
Hand	This class determines the center of hand, size of hand and features of hand via CenterValue(), ComputeSize(), ExtractFeatures() respectively.
Classifier	This class classifies the hand gesture via GestureClassify().
TrainDataSet	This class contains the training dataset and processing on it so that the gestures can be recognized according to the trained dataset.

Table 4-1: Class Description

4.11.2 Package Diagram



Package Diagram 4-1

4.12 Design Patterns

The Design Pattern of Automated Sign Language Recognition System can be modeled using Pipeline. This pattern is used because data flows through a sequence of tasks or stages. It represents a "pipelined" form of concurrency. In this architecture, there are stages and pipes connecting the stages. Stages are the modules of the system that are joined by the pipes. This shows that operations are performed sequentially. Output of one stage is the input to the next stage. There are four stages or modules in this system; Reception of Input Video, Hand Tracking and Localization, Recognize Gesture, Output. In first stage, Reception of input video, input video is received using a camera. This received video will be input for the next stage, which is Hand Tracking and Localization, Hand will be tracked and localized and the output goes to the next stage through pipeline. The next stage is Recognize Gesture. In this stage, classifier will be used to recognize the performed gesture. Output will be correctly identified gesture. This output will go to Output stage which will display the Recognized gesture in the form of text and speech.

CHAPTER 5

SYSTEM IMPLEMENTATION

5. System Implementation

5.1 Gesture Recognition Process: An Overview

5.1.1 Methods of Automated Sign Language Recognition System

The main principle in automated sign language recognition system is to teach the system particular gestures. Each different hand orientation corresponds to a different alphabet of sign language. The learning of the machine is performed by showing the machine examples of alphabets of all the different classes. Based on these examples the machine builds a neural network. Then, during recognition phase, the real time hand orientations are compared to the previously obtained descriptions, and assigned the class that gives the best match. Our system also provides the capability of training the system for new gestures.

5.1.2 Components of Automated Sign Language Recognition System

The system consists of several components. In figure 5-1 a common setup is illustrated. The first step in the process is the reception of input video. After the reception of the video, hand is tracked and localized. The Video is binarized and the contour is extracted. After these processes, the features of the hand are computed.

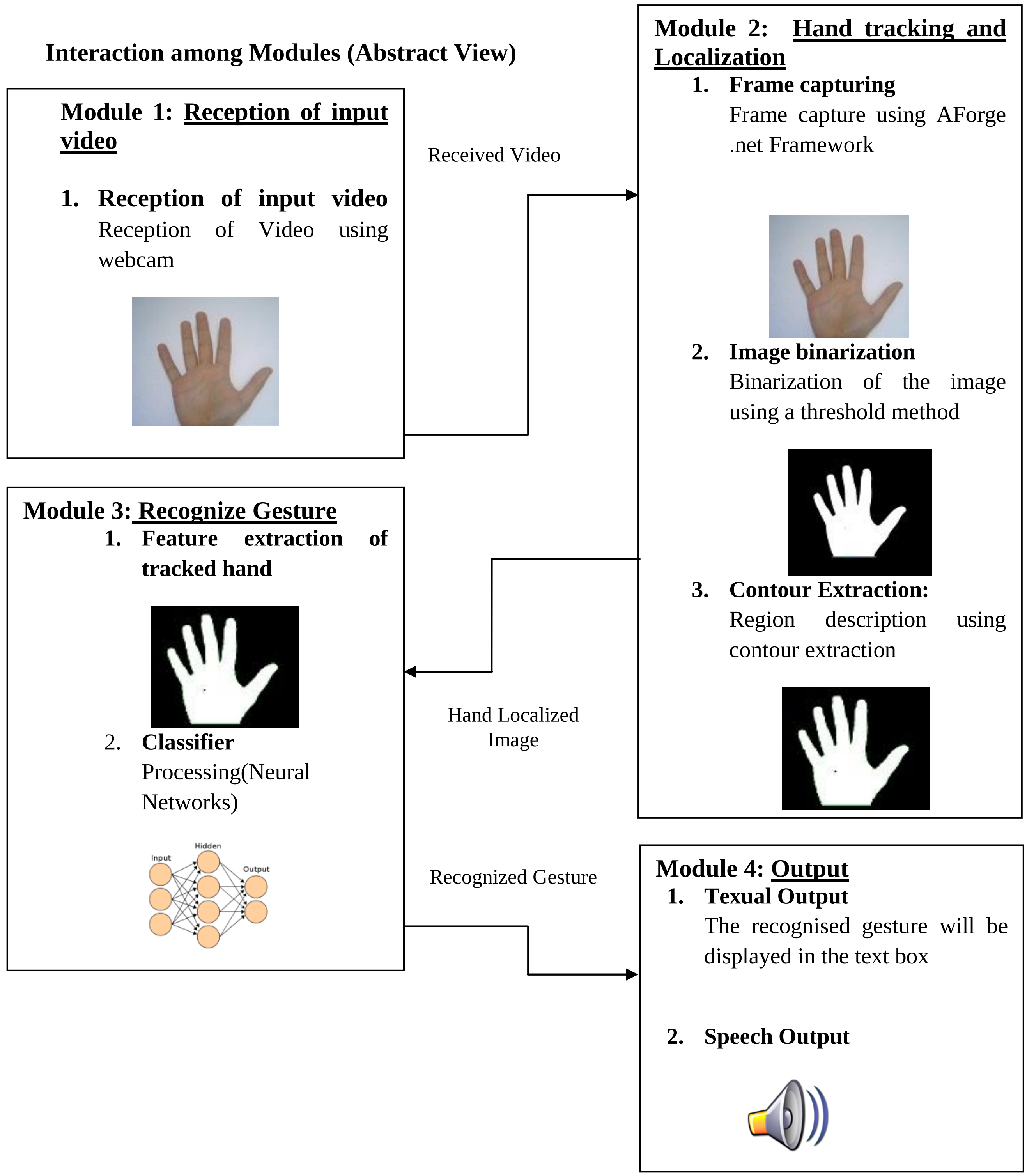


Figure 5-1: A common setup of Automated Sign language Recognition Process

The identity of each gesture is found by comparing the extracted features with the description of the symbol classes obtained through a previous learning phase. Finally the gesture is recognized and the output is displayed in the form of corresponding text and sound is generated.

5.1.3 Our Methodology

The methodology used for Automated Sign Language Recognition System is illustrated in Figure 5-3. It basically consists of a combination of various image processing, feature extraction and machine learning based steps. Each step of the methodology is explained in detail subsequently in this chapter.

5.2 Procedure

5.2.1 Reception of Input Video

Webcam will be used for streaming video. As the quality of the image and subsequently resolution is widely dependent on the camera. The system is trained of image size of 640x480 pixels, so recognition video shall also be of the same resolution.

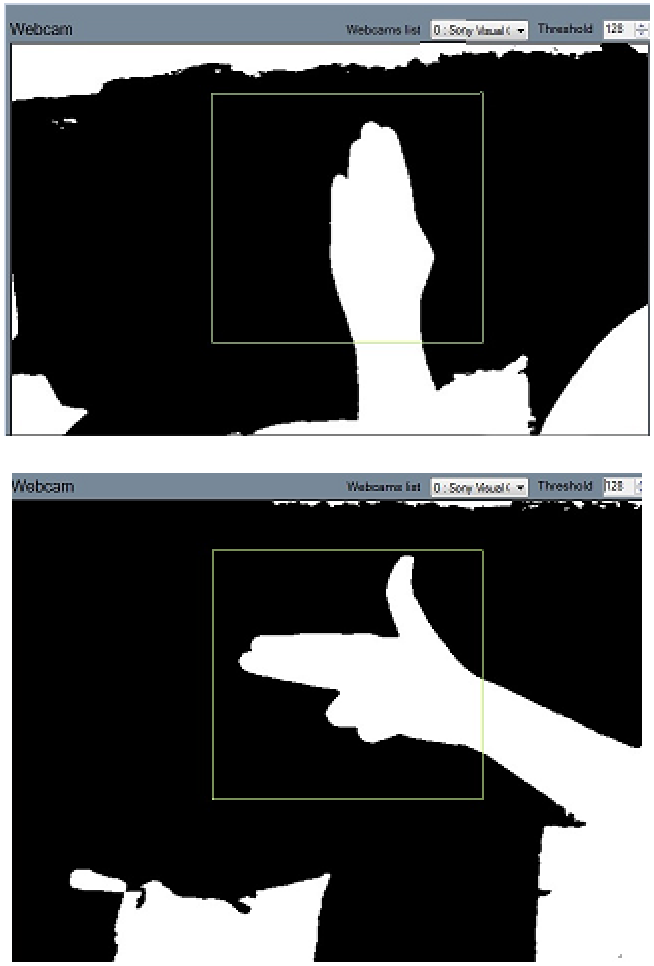


Figure 5-2: Various instances of receiving input video

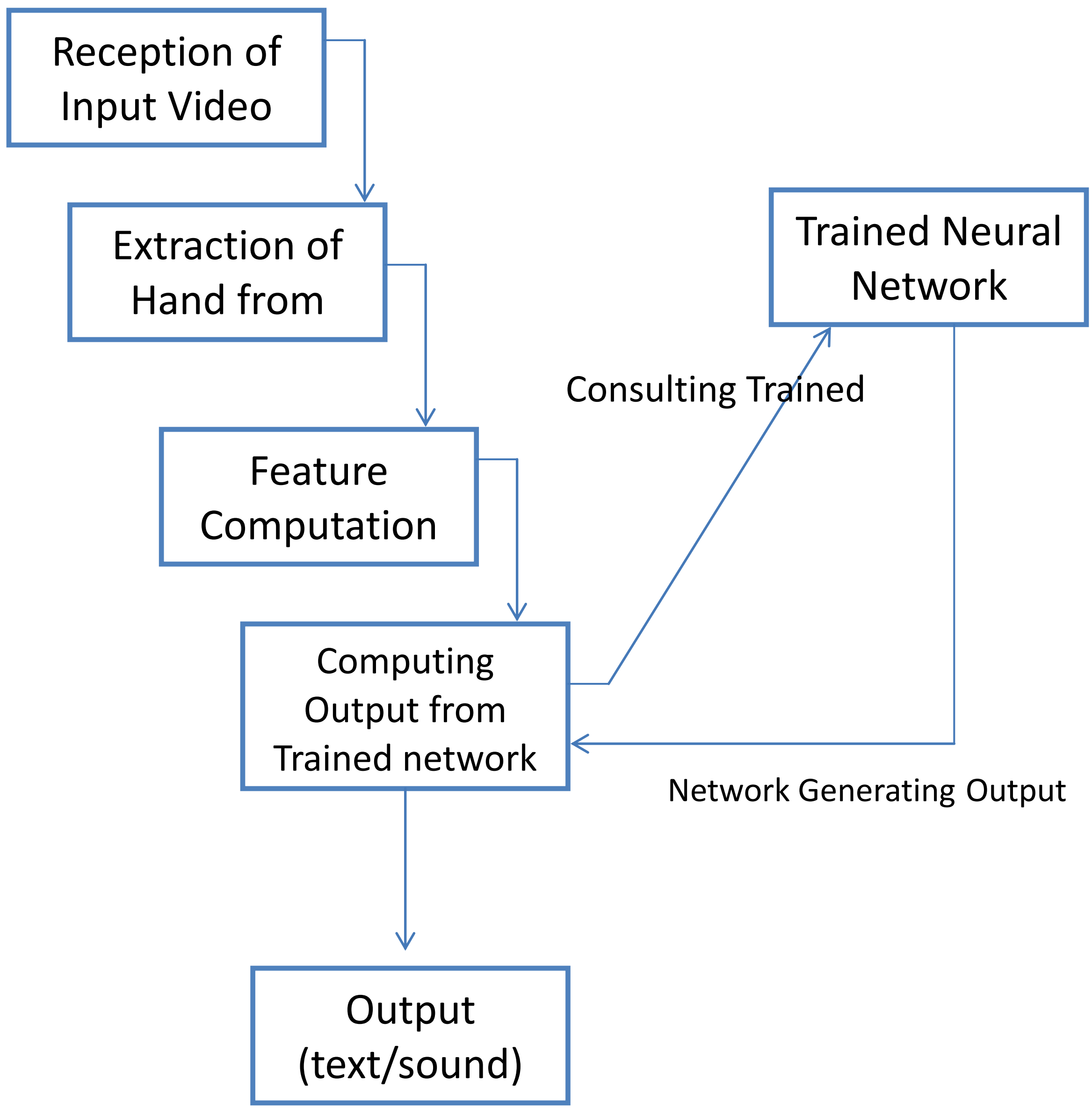


Figure 5-3: Complete flow of Automated Sign Language Recognition System

5.2.2 Hand Tracking and Localization

After the reception of video, hand is tracked and localized using parallel implementation. The software checks for the available system threads. The system generates the copies of the frame and assigns each copy to the thread. The system applies binary filter to each pixel of the frame using binarization method. Adaptive thresholding is necessary as the image may be affected from uneven lightning conditions. In this case, different regions of the image may exhibit different shadings, for which a global threshold may not feasible to binarize the image. Hence, there is a need of different thresholds for different regions of image. The user can set the threshold value at real time, however default value is 128. The contour is extracted that results in the localization and tracking of hand.

5.2.3 Recognize Gesture

After hand tracking and localization, the user presses the Recognize Gesture Button; this maps the hand portion out of the image to a black image. Thus noise is removed as the new image only contains hand with black background. The features of hand are computed using system threads to ensure speedy work.

Following features are computed in parallel using system threads.

Feature	Description
xMin	The starting point of hand along x-axis
xMax	The ending point of hand along x-axis
yMin	The starting point of hand along y-axis

yMax	The ending point of hand along y-axis
xCOG	Center of gravity along x-axis
yCOG	Center of gravity along y-axis
xCenter	Center of hand along x-axis
yCenter	Center of hand along y-axis
Px	(xMax - xCOG) / (xCOG - xMin)
Py	(yMax - yCOG) / (yCOG - yMin)
xRange	The width of hand
yRange	The height of hand
Border Count	The boundary of hand
Border Length	The length of hand
Area	Area of hand in terms of counting number of white pixels

Table 5-1: Hand Features

Following is the list of the features not computed in a parallel fashion.

Feature	Description
Object Compactness	Compactness of hand
Object Aspect ratio	The ratio of width to height
Object Rectangular Area	The product of xRange and yRange
Object Extent	The ratio of area to rectangular area
Object Equivalent Diameter	Math.Sqrt((4.0 * Area * 7.0) / 22.0)

Table 5-2: Feature List

5.3 Noise Removal using Image mapping

The user performs hand gestures in front of the camera. The hand is tracked but this also incorporates noise due to the arm of the user, his body, the background and the lighting conditions. This noise is removed using Image Mapping Technique in which the hand is automatically cropped out of the noisy image and is mapped on a black image.

5.4 Training the Automated Sign Language Recognition System

5.4.1 An Overview

This stage of Automated Sign Language Recognition System is the most critical and challenging stage as far as the accuracy is concerned. The classification stage includes a classifier to correctly classify the input based on training the intended output. Hence the role of classifier is extremely focal in any recognition paradigm. There are various classifiers existing like bi-level or multi-level classifiers. For Automated Sign Language Recognition System we have used Multi-Layer neural networks to train the system. An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of the ANN paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in union to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of

ANNs as well. The most common neural network model is known as a supervised network because it requires a desired output in order to learn. The goal of this network type is to create a model that maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. A graphical representation of a Multi-Layer Perceptron (MLP) is shown in Figure 5-4

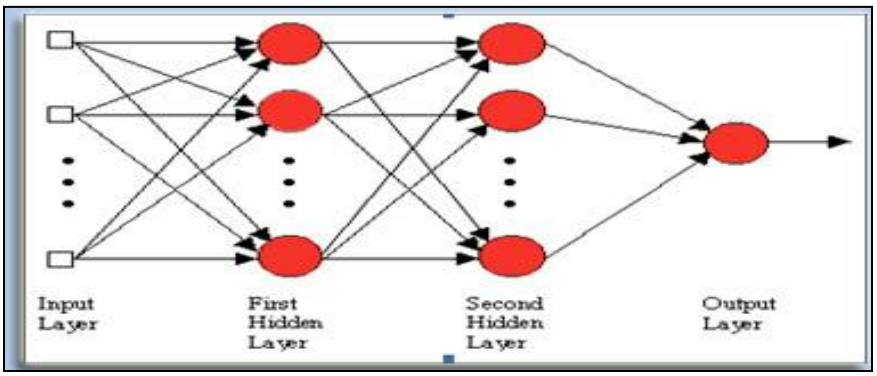


Figure 5-4: A graphical representation of Multi-layer neural network.

Figure 5-5 demonstrates a neural network used within Automated Sign Language Recognition System. For each input gesture, the system consults the trained neural network. Based on the features computed the neural network generates output and returns the closest gesture as output.

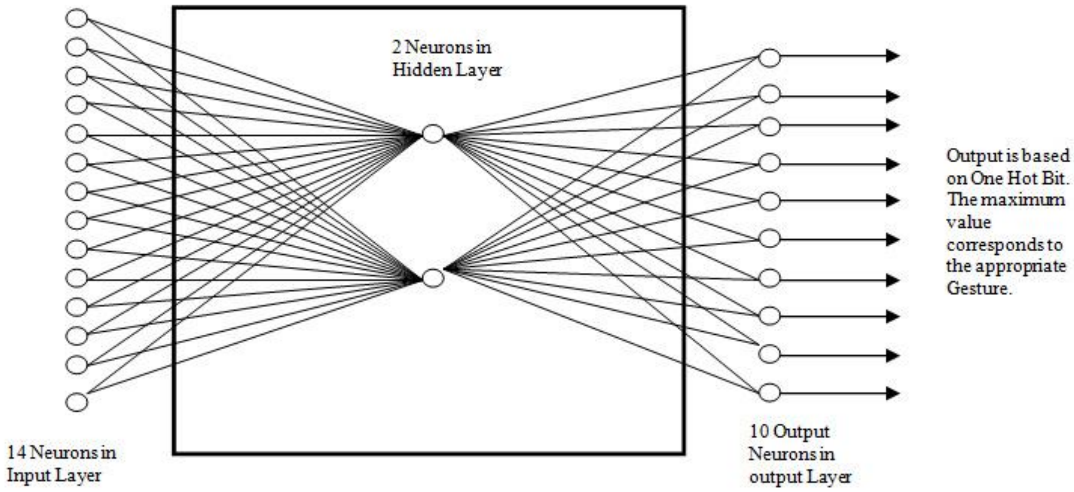


Figure 5-5: Structure of neural network

5.4.2 Our Training Procedure

A back-propagation neural network with 1 hidden layer was used to create a neural network using Bipolar Sigmoid Activation Function. The Output is computed on the basis of One Hot Bit Concept. There are 10 output neurons; the one with maximum value corresponds to the corresponding gesture.

5.4.2.1 Generation of Training Images

5.4.2.1.1 Training Data specifications

The first step was the generation of training images. The training data extracted from the training images. The training data used had following specifications:

The resolution of images was 640x480 pixels.

250 images were used for training data.

Different variations of each gesture were incorporated.

We used 25 such samples for each sign language alphabet.

Features were computed for each gesture image.

The feature values were normalized and excel sheet was generated.

The excel sheet was used as an input for the neural network.

Figure 5-6: Some instances of images used for training. Size of each image is 640x480 pixels

5.4.2.1.2 Neural Network Design

The neural network had three layers: an input layer consisting of 14 nodes (number of features), a hidden layer consisting of 2 nodes, and an output layer with 10 nodes (one for each gesture). The network uses back-propagation in addition to bias weights and momentum.

5.4.2.2 Training

The normalized excel sheet contained the input data for our neural network. The system reads excel sheet and created a data structure for neural network input along with the specified output label for each gesture. The neural network was trained for 1000 epochs the overall training time was approximately 12 sec.

5.5 Recognition, Text Output

The trained file produced in previous stage of training is used in Automated Sign Language Recognition project. The complete system works as follows.

Image is captured using webcam of the device.

Hand Tracking and Localization is done.

Features computed are matched with training file.

Likely alphabet corresponding to the input hand gesture is output in text and sound format.

5.6 Applications

Text to Speech

The text to speech application converts the recognized text to speech. This is a useful application for visually impaired people.

CHAPTER 6

TESTING AND EVALUATION

6. Testing

Testing any software project/product/program is essential to check and ensure the provision of intended functionality and quality of software product. We have tested our software product on two levels:

Interface Testing

Functional Testing

Gesture Recognition Testing

Application Testing

6.1 Interface Testing

Test Case: 1

Test Case ID	01
Test Case name	Start application test
Input(s)	Run the exe file
Output	Application starts
Sequence of Action(s)	Click the exe file. Run the application.
Result	Success

Table 6-1: Test Case 1

Test Case: 2

Test Case ID	02
Test Case name	Select webcam testing
Input(s)	Choose the webcam from the drop down list
Output	Webcam selected
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click the drop down list, select the webcam.
Result	Success

Table 6-2: Test Case 2

Test case: 3

Test Case ID	03
Test Case name	Start button testing
Input(s)	Press Start button , Perform gesture
Output	Live streaming
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click the start button
Result	Success

Table 6-3: Test Case 3

Test Case: 4

Test Case ID	04
Test Case name	Training Data button testing
Input(s)	Click the training data button
Output	Excel file containing features of gesture perform
Sequence of Action(s)	Run the application, Select webcam , set the threshold , click start button, click stop button, click recognize gesture button, Click training data button.
Result	Success

Table 6-4: Test Case 4

Test Case: 5

Test Case ID	05
Test Case name	Recognize gesture button
Input(s)	Click Recognize gesture button.
Output	Captured frame
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button.

Result	Success

Table 6-5: Test Case 5

Test Case: 6

Test Case ID	06
Test Case name	Stop button testing
Input(s)	Click the stop button
Output	Live streaming stops.
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click the start button then Click the stop button
Result	Success

Table 6-6: Test Case 6

Test Case: 7

Test Case ID	07
Test Case name	Setting Threshold testing

Input(s)	Adjust the threshold value
Output	Threshold adjusted
Sequence of Action(s)	Run the application, Select webcam , set the threshold
Result	Success

Table 6-7: Test Case 7

6.2 Functional Testing

We first present the representative test cases of gesture recognition testing.

6.2.1 Gesture Recognition Testing

Test Case: 1

Test Case ID	01
Test Case name	Normal flow testing
Input(s)	Perform gesture
Output	Text and speech output
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button.

Result	Success

Table 6-8: Test Case (Recognition 1)

Test Case: 2

Test Case ID	02
Test Case name	Lightning condition based testing
Input(s)	Perform gesture
Output	Text and speech output
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button.
Result	Success

Table 6-9: Test Case (Recognition 2)

Test Case: 3

Test Case ID	03
Test Case name	Black background testing
Input(s)	Perform gesture in front of black background
Output	Text and speech output

Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button
Result	Success

Table 6-10: Test Case (Recognition 3)

Test Case: 4

Test Case ID	04
Test Case name	Distance based testing
Input(s)	Perform gesture with 1-2 feet distance variation.
Output	text and speech output
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button
Result	Success

Table 6-11: Test Case (Recognition 4)

Test Case: 5

Test Case ID	05

Test Case name	Angle based testing
Input(s)	A gesture with variation in angle
Output	Text and speech output
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button
Result	Success

Table 6-12: Test Case (Recognition 5)

6.2.2 Application Testing

Now in the second phase of functional testing we would test each of our application.

Test Case: 6

Test Case ID	06
Test Case name	Text to Speech testing
Input(s)	Input gesture text
Output	Alphabet pronunciation in Urdu
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button
Result	Success

Table 6-13: Test Case (Recognition 6)

Test Case: 7

Test Case ID	07
Test Case name	Gesture to text testing
Input(s)	A gesture performed
Output	Text of gesture shown in output box
Sequence of Action(s)	Run the application, Select webcam , set the threshold , Click Start button, Perform gesture, Click Stop button, Click recognize gesture button
Result	Success

Table 6-14: Test Case (Recognition 7)

6.3 Results

We here present our sign language recognition results on the basis of different parameters used for testing and evaluation.

Validating Automated Sign Language Recognition System using Validation Data

Total samples: 45

Correctly Classified: 43

Incorrect Classification: 2

Accuracy: 95.555 %

Real Time Testing

Total samples: 97

Correctly Classified: 84

Incorrect Classification: 13

Accuracy: 89.56%

CHAPTER 7

CONCLUSION AND FUTURE WORK

7. Conclusion & Future Work

Hearing impaired equipment is required by every nation and every society as disable people are a part of our society and they can simply be not forgotten. All the solutions are very expensive which normally cannot be afforded at a large scale. Limitations of using data glove or any other medium have been eliminated by this solution. Hence a very cost effective solution has been presented which is simple enough to used yet providing all the functionality.

The project recognizes ten Urdu alphabets. It can be further extended to cover all the Urdu alphabets. Further, it can be extended more to recognize Urdu words and then sentences.

The scope of the project can be enhanced and it can be extended to cover the Urdu Language.

APPENDIX A-1

USER MANUAL

TABLE OF CONTENTS

8. APPENDIX A-1

8.1 USER MANUAL

1. Reading Instructions

This manual is a guide to the system "Automated Sign Language Recognition System".

It contains essential instructions for running and using the system.

The system provides a user friendly interface which allows you to directly interact and use the system.

This manual should be read in the order given.

2. Installation

Run the exe file.

3. How to use the system

Operation of Automated Sign Language Recognition System comprises of the following steps:

1. Run the system.

2. Select the webcam from the drop down list.

3. Adjust the threshold.

4. Click the Start button to start the live streaming for performing gesture.

5. Perform the gesture at a distance of about 1-2 feet.

6. When you are done with forming the right gesture, click the Stop button.

7. Click the Recognize Gesture button.

8. Text output will be shown in output box followed by speech.