

# Hand Gesture Recognition



ALMAS ARSHID  
01-134131-008

## Bachelor of Science in Computer Science

Supervisor: Ma'am Mahwish Pervaiz

Department of Computer Science  
Bahria University, Islamabad

May 2017



# Certificate

I accept the work contained in the report titled “Hand Gesture Recognition System”, written by Mr. Almas Arshid as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science..

Approved by . . . :

Supervisor: Ma'am Mahwish Pervaiz (Assistant Professor)

---

Internal Examiner: ()

---

External Examiner: ()

---

Project Coordinator: Dr. Sumaira Kausar (Sr. Assistant Professor)

---

Head of the Department: Dr. Faisal Bashir (Associate Professor)

---

Dec 27<sup>th</sup>, 2017

# Abstract

This work discusses an on-going project for hands gesture recognition in computer vision systems. The proposed approach is based on the shape analysis. More specifically, the K nearest neighbour algorithm is used statistical estimation and pattern recognition of hand gestures based on contours. This project also facilitate to the deaf people they can be easily communicate with other peoples.

*“We think someone else, someone smarter than us, someone more capable, someone with more resources will solve that problem. But there isn’t anyone else.”*

Regina Dugan

# Contents

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 About the System . . . . .	1
1.2 Problem Description . . . . .	1
1.3 Project Scope . . . . .	2
<b>2 Literature Review</b>	<b>3</b>
2.1 Related Work . . . . .	3
<b>3 Requirement Specifications</b>	<b>5</b>
3.1 System Overview and Scope . . . . .	5
3.2 Existing System . . . . .	5
3.3 Proposed System . . . . .	5
3.4 Requirements . . . . .	6
3.4.1 Functional Requirements . . . . .	6
3.4.2 Non-Functional Requirements . . . . .	6
3.5 Project Extends . . . . .	7
3.6 Use Cases . . . . .	7
<b>4 Design</b>	<b>14</b>
4.1 Objective . . . . .	14
4.2 System Architecture . . . . .	14
4.3 Package Diagram . . . . .	15
4.4 Data Flow Diagram . . . . .	15
4.5 Deployment Diagram . . . . .	17
4.6 Sequence Diagram . . . . .	18
<b>5 System Implementation</b>	<b>19</b>
5.1 Objective . . . . .	19
5.2 Tools and Technologies Used . . . . .	19
5.2.1 Environment . . . . .	19
5.2.2 Language . . . . .	19
5.2.3 Libraries . . . . .	19
5.2.4 Metro Framework . . . . .	20
5.2.5 Methodology . . . . .	20
5.3 Block Diagram . . . . .	20

<b>6</b>	<b>System Testing and Evaluation</b>	<b>22</b>
6.1	Objective . . . . .	22
6.2	Usability Testing . . . . .	22
6.2.1	Usability . . . . .	22
6.2.2	Learn-ability . . . . .	22
6.2.3	Safety . . . . .	22
6.2.4	Consistency . . . . .	23
6.2.5	GUI Testing . . . . .	23
6.3	Performance Testing . . . . .	26
6.4	Compatibility Testing . . . . .	26
6.4.1	Device . . . . .	27
6.4.2	Operating System . . . . .	27
6.5	Installation Testing . . . . .	27
6.6	Test Cases . . . . .	27
6.6.1	Test Cases 1 . . . . .	27
6.6.2	Test Cases 2 . . . . .	28
6.6.3	Test Cases 3 . . . . .	28
6.6.4	Test Cases 4 . . . . .	29
6.6.5	Test Cases 5 . . . . .	29
<b>7</b>	<b>Conclusions And Future Work</b>	<b>30</b>
	<b>conclusion</b>	<b>30</b>

# List of Figures

3.1	Home page of HGR . . . . .	6
3.2	Main Use Case Diagram . . . . .	7
3.3	Trainer Use Case . . . . .	8
3.4	Start Video . . . . .	8
3.5	view Result . . . . .	9
3.6	Capture Gesture . . . . .	9
3.7	Capture Video . . . . .	10
3.8	Exit . . . . .	10
3.9	Start . . . . .	11
3.10	Start . . . . .	11
3.11	Capture Gestures . . . . .	12
3.12	Match Faces . . . . .	12
3.13	Exit . . . . .	13
4.1	System Architecture . . . . .	14
4.2	Package Diagram . . . . .	15
4.3	Data Flow Diagram . . . . .	16
4.4	Package Diagram . . . . .	17
4.5	Deployment Diagram . . . . .	17
4.6	User Sequence Diagram . . . . .	18
4.7	Trainer Sequence Diagram . . . . .	18
5.1	system Architecture . . . . .	20
6.1	home page . . . . .	23
6.2	home page . . . . .	24
6.3	home page . . . . .	24
6.4	home page . . . . .	25
6.5	home page . . . . .	25
6.6	home page . . . . .	26
6.7	test case . . . . .	27
6.8	test case . . . . .	28
6.9	test case 3 . . . . .	28
6.10	test case 4 . . . . .	29
6.11	test case 5 . . . . .	29



# List of Tables

# Acronyms and Abbreviations

HGR	Hand Gesture Recognition
GUI	Graphical User Interface
IDE	Integrated Development Environment
OS	Operating System
TC	Test Case
UC	Use Case

# Chapter 1

## Introduction

The objective of this chapter is to describe the background objectives scope and modules of our project. The chapter provides an introduction to the techniques and methodologies used in the development of the system.

### 1.1 About the System

Hand gesture recognition is important research field because it is the most frequently used and most common way of expression of emotions. Hands are human organs which are used to manipulate physical objects. For this very reason hands are used most frequently by human beings to communicate and interact with machines .Hand gestures are now become a source of human computer interaction, such application allows user to control multiple applications through hand gesture. Two relevant application of such systems are Glove-based gesture and Marker based alphabet recognition was proposed by [1] and [2]. These types of application use sign language to understand the action behind every gesture. We also need to understand sign language when we communicate with deaf people.

### 1.2 Problem Description

There exists some situation when we have to use sign language to communicate with people or with machine. Like in a surgery a doctor may gesture to his colleague for assistance or we have to communicate with our friend who is deaf, or we want to communicate with computer through gesture. In all these situation it is required to have knowledge of sign language, for people and for computer also. To cope with this problem there must be an application that helps people to learn/ understand sign language through gesture. So, that if we want to learn sign language we can take help from that application and if we want to communicate with computer through sign language our machine can understand what we wants to perform with it. This is the main problem we want to resolve.

### **1.3 Project Scope**

The project is about the hand gesture recognition, it recognize the gesture using image captured through webcam. Because of its limitation we deal with fix number of gesture to train our system. So, system recognizes limited set of hand gesture. The system is able to recognize the characters from A to M. This software however, would not recognize any other object other than the hand. The background should be clear. Another phase of our project is training. The trainee train the system using image captured through webcam. In a training phase the trainee take 20 pictures per character and store it in local storage and train data. The data stores in xml,jpg file.

## Chapter 2

# Literature Review

### 2.1 Related Work

Background Literature several methods are proposed for both dynamic and static hand gestures. [6] Pujan Ziaie proposed a technique of first computing the similarity of different gestures and then assign probabilities to them using Bayesian Interface Rule. Invariant classes were estimated from these using a modification of KNN (K-Nearest Neighbor). These classes consist of Hu- moments with geometrical attributes like rotation, transformation and scale in variation which were used as features for classification. Performance of this technique was very well and it was giving 95% accurate results. [5] Pujan Ziaie also proposed a similar technique which also uses HU-moments along with modified KNN (K-Nearest Neighbor) algorithm for classification called as Locally Weighted Naive Bayes Classifier. Classification results were this technique were 93% accurate under different lighting conditions with different users. [3] Rajat Shrivastava proposed a method, in which he used HU moments and hand orientation for feature extraction. Baum Welch algorithm was used for recognition. The method has accuracy of 90%. [4] Technique propose by Neha S. Chourasia, Kanchan Dhote and Supratim Saha used a hybrid feature descriptor, combining HU invariant moments and SURF. They used (KNN) K-nearest neighbors and SVM for classification. They Achieved 96% accuracy. [5] Joyeeta Singa proposed a hand gestures recognition system based on K-L Transform. This system was consisting of five steps, which are; skin filtering (Image acquisition, converting RGB to HSV, filtering image, smoothing image, binary image, finding biggest BLOB), palm cropping, Hand edge detection using Canny edge detector, feature extraction using K-L Transform and classification. [2] Huter proposed a system that uses Zernike moments to extract image features and used Hidden Markov Model for recognition. [7] Raheja proposed a technique that scanned the image all directions to find the edges of finger tips. [1] Segan proposed a technique that used edges for feature extraction. This reduces the time complexity and also help for removing noise. Image sensing Image is a two-dimensional function  $f(x,$

y), where x and y are spatial coordinates, and the amplitude of at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. Image creation is based on two main factors which are; Reflection or absorption of energy from the object being imaged and Illumination source. Illumination source can be an electromagnetic energy like; infrared, or X-ray or sources like ultrasound, sunlight or Computer generated illumination pattern. In some cases, the energy that is transmitted or reflected is focused onto converter, this is called photo converter. This photo converter converts energy into visible light. A basic arrangement of sensors is used to convert energy.

**Conclusion:** Method Proposed Method In order to extract features and recognize a gesture following method is proposed: A GUI which allows the user to capture the scene. This phase is called image acquisition. After capturing the image, next step is to detect the hand and separate the hand from the scene, because only hand gesture is needed for accurate classification. If hand is not separated from the scene it will affect the accuracy of the system while extracting and matching the features. Crop hand out of scene. Pre-processing steps, which are:

- Convert RGB to Gray scale.
- Gray filtering using Value.
- Noise removal and smoothing. Feature extraction. Classification using KNN algorithm. Using Euclidean distance formula for calculating distance and having threshold to have better results. Translation (Gestures) in Characters.

## **Chapter 3**

# **Requirement Specifications**

### **3.1 System Overview and Scope**

The project is about the hand gesture recognition, it will recognize the gesture using image captured through webcam. Because of its limitation we deal with fix number of gesture to train our system. So, system will recognize limited set of hand gesture. This software, however, would not recognize any other object other than the hand. Image is captured through webcam. After image acquisition the next step is hand detection as we are dealing only hand gesture so we will use features of hand only. Different gesture contains different angles of fingers and different number of fingers is used in every gesture so after hand detection our system will detect the expanded fingers and their angles.

### **3.2 Existing System**

We have performed our research we have found different hand gesture recognition software but exists none that performs these feature to facilitate the deaf people. Our system is one of its kinds just not because of its feature and requirements to deal with complex background and low resolution of images, but because of its application area.

### **3.3 Proposed System**

The proposed system will be built in Microsoft Visual Studio using windows forms. Our system can be employed by any media outlet organization or education departments who keep interest in communication with deaf people. Application area of our system also be used by the news channels.3.1 provides the interface of the homepage of the proposed system.



Figure 3.1: Home page of HGR

## 3.4 Requirements

### 3.4.1 Functional Requirements

- The system should be able to deal with complex backgrounds.
- The system should be able to process at least three frames per second.
- The systems should be able to put up with differen orientation of Hands
- The system should be able to recognize Hands

### 3.4.2 Non-Functional Requirements

- Extraction of features from image should not take more than 15 seconds.
- The browsing window should open in lesser than 5 seconds.
- The webcam access should be in 5 seconds.
- The result of gesture should be display in 1 second.
- In case of system crash, the reboot must not take more than 2 minutes



### 3.5 Project Extends

The extends of our project include the following features:

- Improving accuracy of Hand detection using classification and KNN algorithm techniques.
- Improving accuracy of gesture recognition to deal with complex background.

### 3.6 Use Cases

In 3.2 the Use Case Diagram for HGR can be seen.

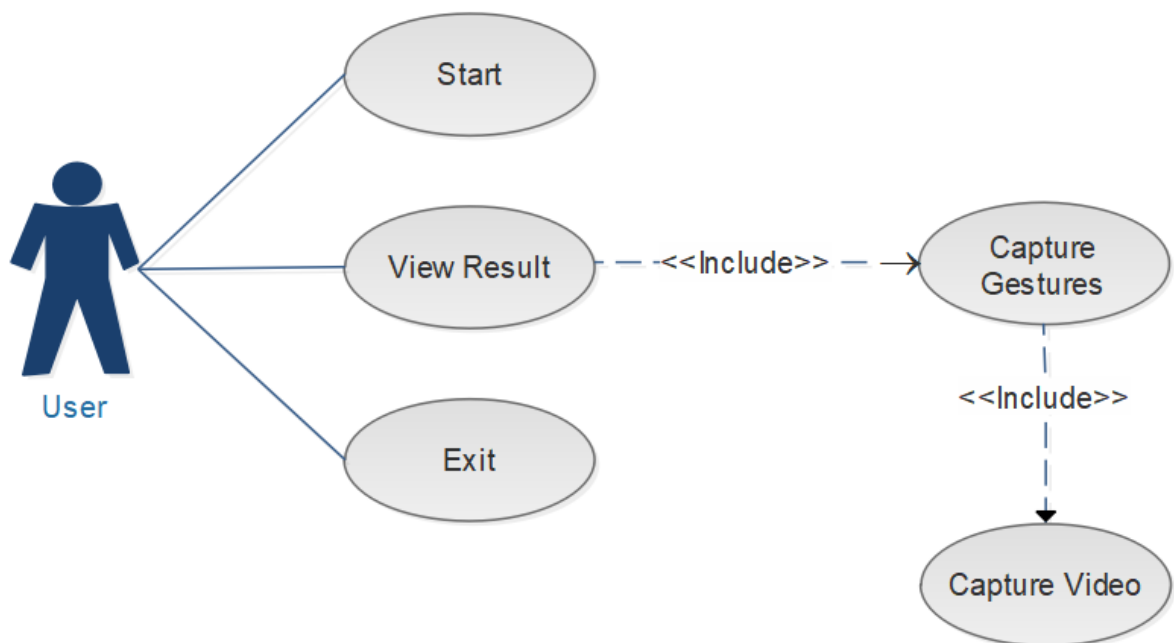


Figure 3.2: Main Use Case Diagram

**Use Case Description:** User starts the application. After that the user capture video and then capture gesture after that the application shown the result of that gesture. After the result user can exit from application.

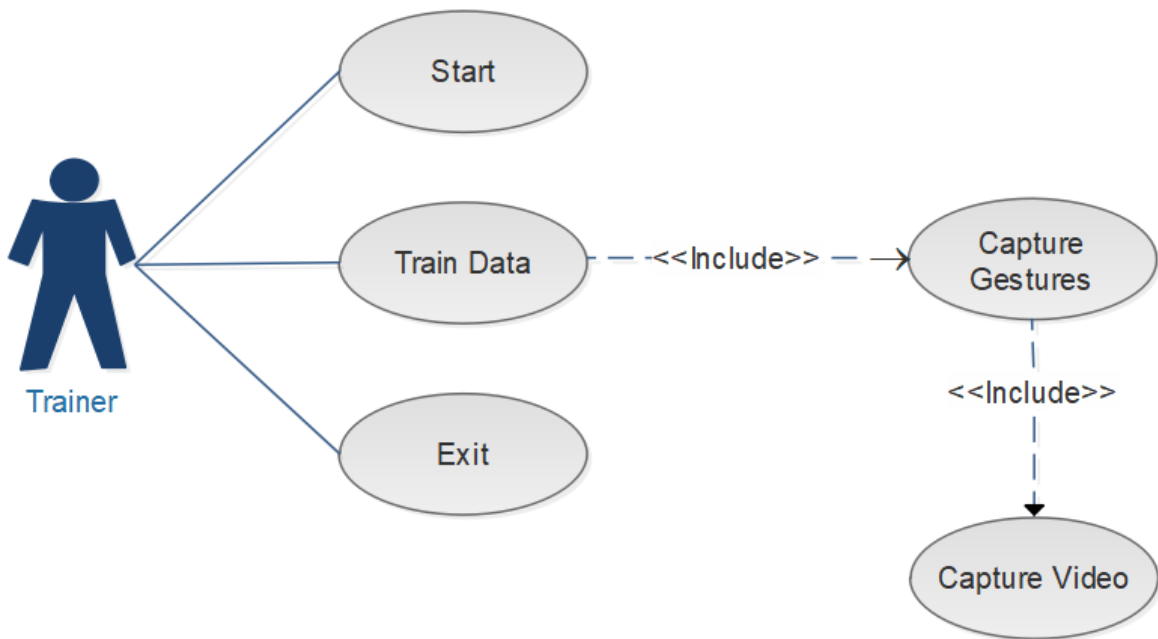


Figure 3.3: Trainer Use Case

**Use Case Description:** The trainer starts the application. After that the trainer capture video and then capture gesture after that the data is store in local storage and then trainer retrieve the data and train. After the training the trainer can exit from application.

In 3.4, use case for Start can be seen.

User Case Section	Comments
Use Case Name	Start
Use Case ID	UC-01
Primary Actor	User
Stakeholders and Interests	User is the stakeholder and He/She wants to start the application.
Preconditions	-
Success Guarantee	The application should start.
Main Success Scenerio	-
Special Requirements	-
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	This use case include 0 use case.

Figure 3.4: Start Video

In 3.5, the use case for View Result can be seen.

Use Case Section	Comments
Use Case Name	View Result
Use Case ID	UC-02
Primary Actor	User
Stakeholders and Interests	User is the stakeholder and He/She wants the result.
Preconditions	Gesture should recognized.
Success Guarantee	The results display in characters .
Main Success Scenerio	-
Special Requirements	Result should not take more then 1 second.
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	This use case include 2 use cases.

Figure 3.5: view Result

In 3.6, the use case for Capture Gesture can be seen.

Use Case Section	Comments
Use Case Name	Capture Gestures.
Use Case ID	UC-03
Primary Actor	User
Stakeholders and Interests	User is the stakeholder .
Preconditions	Video should capturing.
Success Guarantee	-
Main Success Scenerio	-
Special Requirements	Background should subtract.
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	This use case include 1 use case.

Figure 3.6: Capture Gesture

In 3.7, the use case for Capture Video can be seen.

Use Case Section	Comments
Use Case Name	Capture Video.
Use Case ID	UC-04
Primary Actor	User
Stakeholders and Interests	User is the stakeholder .
Preconditions	Webcam should open.
Success Guarantee	-
Main Success Scenerio	-
Special Requirements	The webcam should not take more then 5 second to open.
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	-

Figure 3.7: Capture Video

In 3.8, the use case for Exit can be seen.

Use Case Section	Comments
Use Case Name	Exit
Use Case ID	UC-05
Primary Actor	User
Stakeholders and Interests	User is the stakeholder .
Preconditions	Application should open.
Success Guarantee	-
Main Success Scenerio	-
Special Requirements	-
Technology And Data Variations List	-
Frequency Of Accurrence	Low
Miscellaneous	-

Figure 3.8: Exit

In 3.9, the use case for Start can be seen.

User Case Section	Comments
Use Case Name	Start
Use Case ID	UC-05
Primary Actor	Trainer
Stakeholders and Interests	Trainer is the stakeholder and He/She wants to start the application.
Preconditions	-
Success Guarantee	The application should start.
Main Success Scenerio	-
Special Requirements	-
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	This use case include 0 use case.

Figure 3.9: Start

In 3.10, the use case for Train Data can be seen.

Use Case Section	Comments
Use Case Name	Train Data
Use Case ID	UC-07
Primary Actor	Trainer
Stakeholders and Interests	Trainer is the stakeholder and He/She wants the result of trained data.
Preconditions	Gesture should recognized.
Success Guarantee	-
Main Success Scenerio	-
Special Requirements	Result should not take more then 5 second.
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	This use case include 2 use cases.

Figure 3.10: Start

In [11], the use case for Recognize Face can be seen

Use Case Section	Comments
Use Case Name	Capture Gestures.
Use Case ID	UC-03
Primary Actor	Trainer
Stakeholders and Interests	Trainer is the stakeholder .
Preconditions	Video should capturing.
Success Guarantee	-
Main Success Scenerio	-
Special Requirements	Background should subtract.
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	This use case include 1 use case.

Figure 3.11: Capture Gestures

In [12], the use case for Match Faces can be seen.

Use Case Section	Comments
Use Case Name	Capture Video.
Use Case ID	UC-09
Primary Actor	Trainer
Stakeholders and Interests	User is the stakeholder .
Preconditions	Webcam should open.
Success Guarantee	-
Main Success Scenerio	-
Special Requirements	The webcam should not take more then 5 second to open.
Technology And Data Variations List	-
Frequency Of Accurrence	High
Miscellaneous	-

Figure 3.12: Match Faces

In [13], the use case for Exit can be seen.

Use Case Section	Comments
Use Case Name	Exit
Use Case ID	UC-10
Primary Actor	Trainer
Stakeholders and Interests	Trainer is the stakeholder .
Preconditions	Application should open.
Success Guarantee	-
Main Success Scenerio	-
Special Requirements	-
Technology And Data Variations List	-
Frequency Of Accurrence	Low
Miscellaneous	-

Figure 3.13: Exit

# Chapter 4

## Design

### 4.1 Objective

The objective of this chapter is to describe the design of HDR using Visual Models. The design document includes Use Case diagram, Fully Dressed, Package Model, Deployment Diagram, Data flow Diagram, and the Domain Model.

### 4.2 System Architecture

The Fig 4.1 describes the architecture of our system.

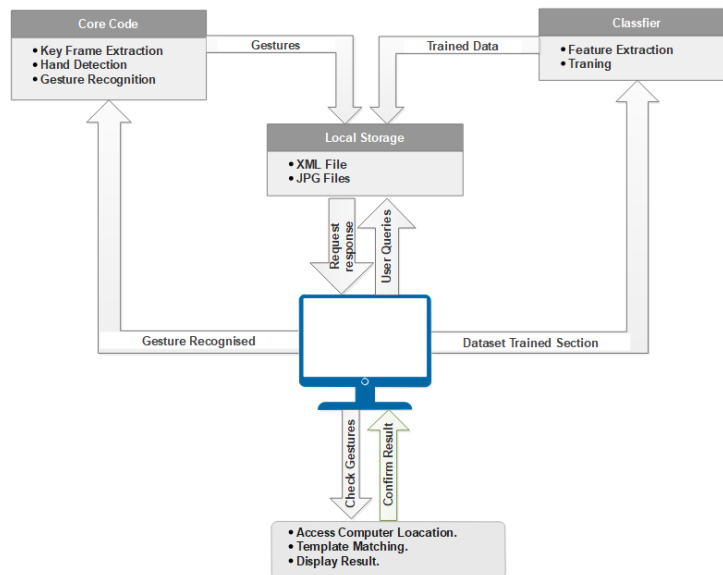


Figure 4.1: System Architecture

**Description:** In the figure 4.1 we can see in the core code section there are interact with the local storage with input of gestures and classifier also with local storage with



trained data input and local storage response to the system and system response with others.

### 4.3 Package Diagram

The package diagram of HDR can be seen in 4.2 .

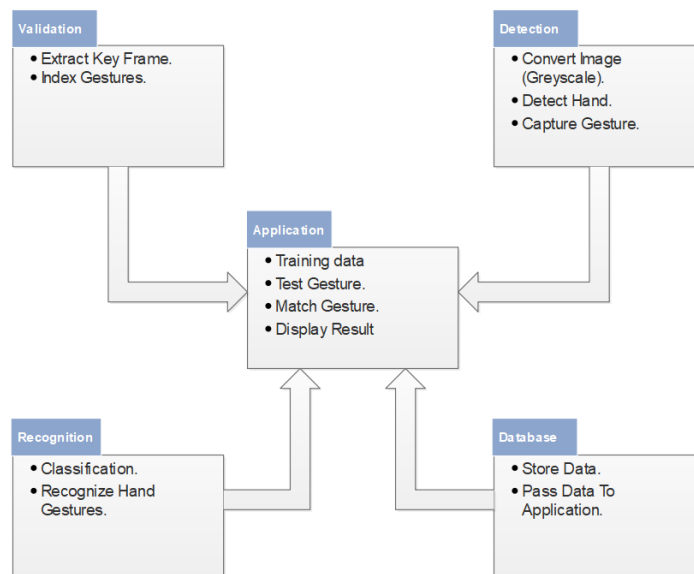


Figure 4.2: Package Diagram

**Description:** There are five packages Validation, Detection, Recognition, Database and Application. Validation, Detection, Recognition and Database are interact with application.

### 4.4 Data Flow Diagram

The Data Flow diagram of HDR can be seen in fig 4.3

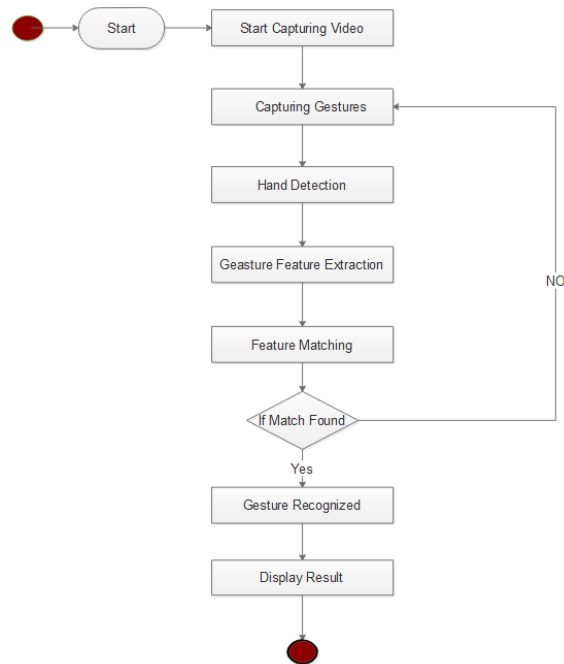


Figure 4.3: Data Flow Diagram

**Description:** User dataflow diagram shows the steps. First the user starts the application after that he Capture gesture after this the application detect hand and subtract the background and then extract feature and if gesture recognized then display result.

The Package Diagram of HDR can be seen in fig 4.4

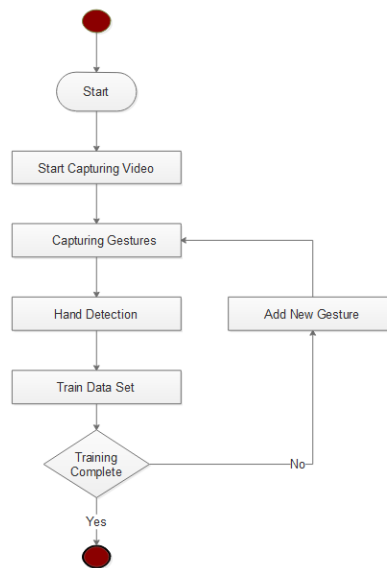


Figure 4.4: Package Diagram

**Description:** Trainer dataflow diagram shows the steps. First the trainer starts the application after that he Capture gesture after this the application detect hand and subtract the background. We used supervise learning technique for training. The trainer takes the picture for specific gesture there are 13 characters and each has 20 pictures. The trainer takes 260 pictures and then trains those pictures.

## 4.5 Deployment Diagram

The deployment diagram of HGR can be seen in Fig 4.5

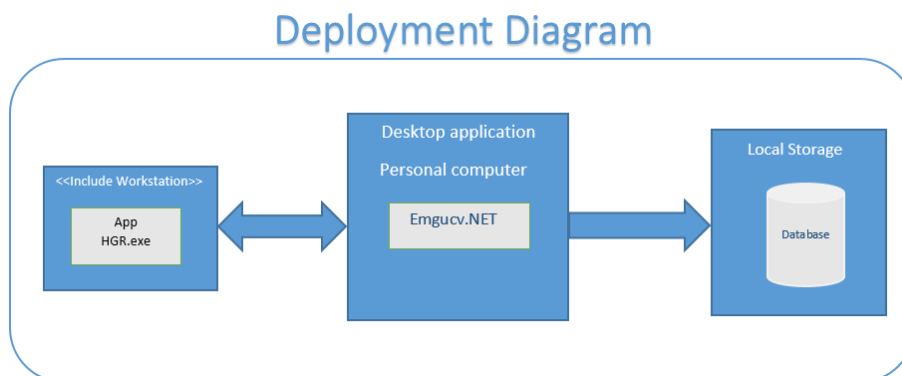


Figure 4.5: Deployment Diagram

## 4.6 Sequence Diagram

The sequence diagram of HDR can be seen in fig 4.6.

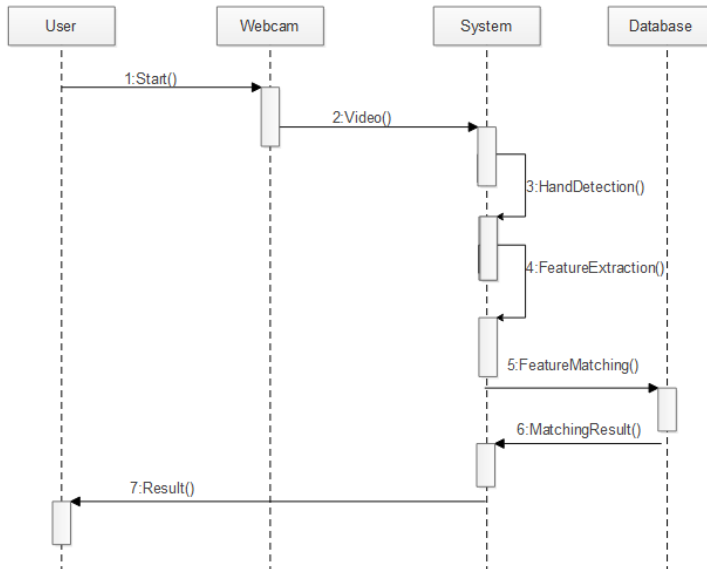


Figure 4.6: User Sequence Diagram

**Description:** In this user sequence diagram we can see there are four objects that is USER, WEBCAM, SYSTEM, and DATABASE. The user starts and webcam starts video and then the system detect hand; extract feature and then system display the result to the user.

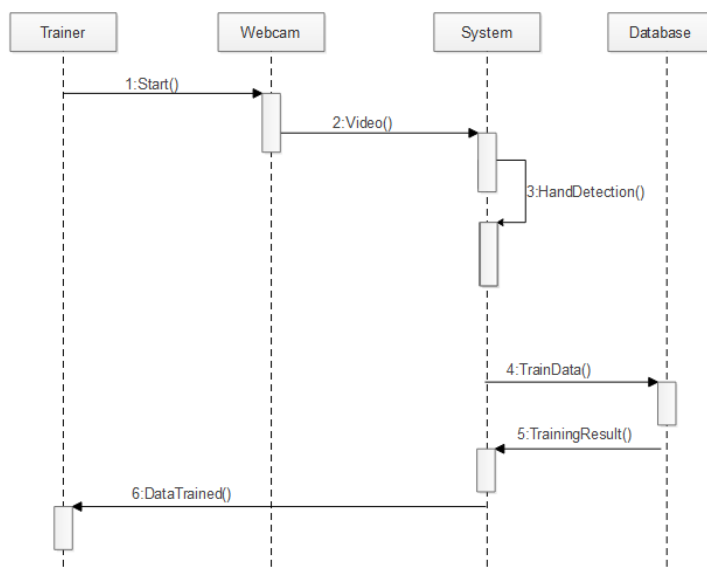


Figure 4.7: Trainer Sequence Diagram

## **Chapter 5**

# **System Implementation**

### **5.1 Objective**

The objective of this chapter is to provide an overview of the tools and technologies and the libraries used to develop this system. This chapter also provides the system architecture of the HGR.

### **5.2 Tools and Technologies Used**

The following are the tools and technologies used in the development of HGR

#### **5.2.1 Environment**

Microsoft Visual Studio 2013 (an IDE from Microsoft Corporations) was used as a coding environment in the development phase of the system. The software provides a very interactive coding environment as is widely used in the development of such systems.

#### **5.2.2 Language**

We have used C-sharp programming language for the development of our system. The basic idea was to take advantage of the Windows Forms that the C-sharp provides us to be more interactive with the user, for browsing a file and opening image queries, windows forms provide a very practical user interface.

#### **5.2.3 Libraries**

The Emgucv.net framework is used for the image processing in the system. Emgucv.net, operating under the name Accord.Net, provides APIs for Computer Vision, Image Processing, and Video Processing. The libraries from Emgucv.net were used for the retrieval of frames from videos, Hand detection, and Hand recognition.

### 5.2.4 Metro Framework

Metro Framework is a Windows Modern UI for .NET WinForms Applications. It provides a Windows 8 like the theme for a very elegant user interface. The Framework was used by adding additional libraries to the system.

### 5.2.5 Methodology

I am going to use some classes that convert image RGB to Gray scale image. After that we draw rectangle that help us to subtract the background and to detect hand. I am using KNN (nearest neighbour) for feature extraction. The extraction of features from the images (of Hand) can be used to get the measure for the similarity between the images. Xmlserializer is use to save the feature of the gesture in xml file and retrieve them for template matching. The result of the gesture is in the accii we convert it into character and display to the user [5].

## 5.3 Block Diagram

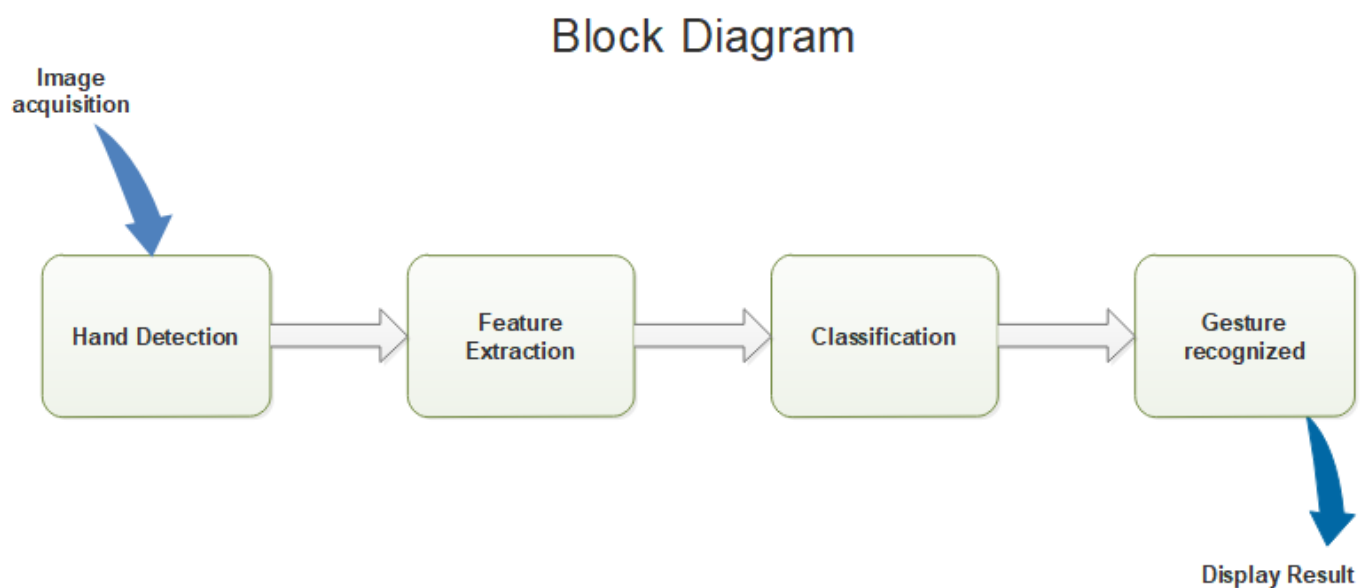


Figure 5.1: system Architecture

**Description:** In the fig: [5.1] we can see the block diagram. Image acquisition is an input. When the system take input the next step is to detect the hand the hand should be in particular box and the back ground should clear. Transforming the input data into the set of features is called feature extraction. Feature extraction is performed on raw data prior to applying k-NN algorithm on the transformed data in feature space. We used

K-nearest neighbour algorithm. The input of our project is an image. The output should be in characters. The application will recognize the gesture and translate it into characters

## **Chapter 6**

# **System Testing and Evaluation**

### **6.1 Objective**

The objective of this project is to test and evaluate the functionalities of the system. We use different techniques to find the restrictions of the software.

### **6.2 Usability Testing**

Usability testing delivers to specify the time a software system requires to execute an explicit task of the system. Usability testing is assessed by the objective audience of the software application. These tests express if the system is executing the tasks of its intentions or not. The criteria for usability testing is whether the system is easy to use, easy to learn to use, safe to use, and consistent. All these criterion have been tested in the following subtopics.

#### **6.2.1 Usability**

The system is very easy to use as the User Interface is not very complex and the texts on the buttons are self-explained.

#### **6.2.2 Learn-ability**

The UI of the system is very easy to comprehend, even for a new user.

#### **6.2.3 Safety**

The system is safe to use.



### 6.2.4 Consistency

The UI of the system is consistent so it will provide no difficulty for user to perform the functionalities provided by the system.

### 6.2.5 GUI Testing

The Graphical User Interface of our system is self-explanatory. Our system's GUI contains the buttons, scroll bar, picture box. All the buttons are performing their functionalities efficiently. Scroll bar is used to select the threshold limit. The GUI has a consistent colour scheme so the user can easily understand the system. The Figures 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6 show the graphical user interface of HGR.



Figure 6.1: home page

**Description:** In figure 6.1 we can see the home page of HGR we can see there are two main buttons that are Training and start. The start button is for user and the training button is for trainer.

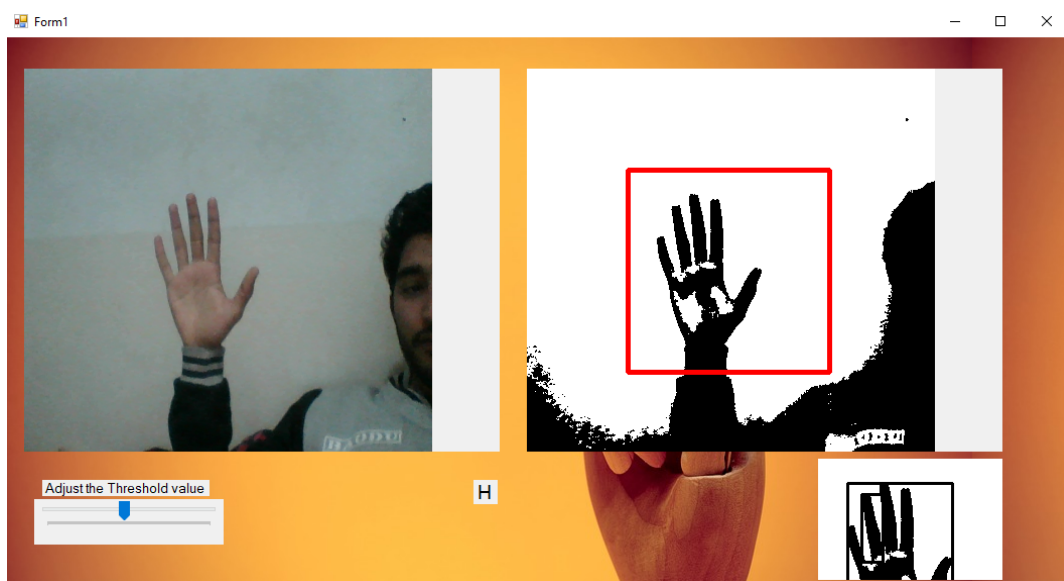


Figure 6.2: home page

**Description:** In figure 6.2 we can see there is the gesture of H we can also see the result of this gesture. We have to adjust the threshold for the subtraction of background. Make threshold value until you get the hand black and the rest of area should white.



Figure 6.3: home page

**Description:** In figure 6.3 we can see the training section of HGR.

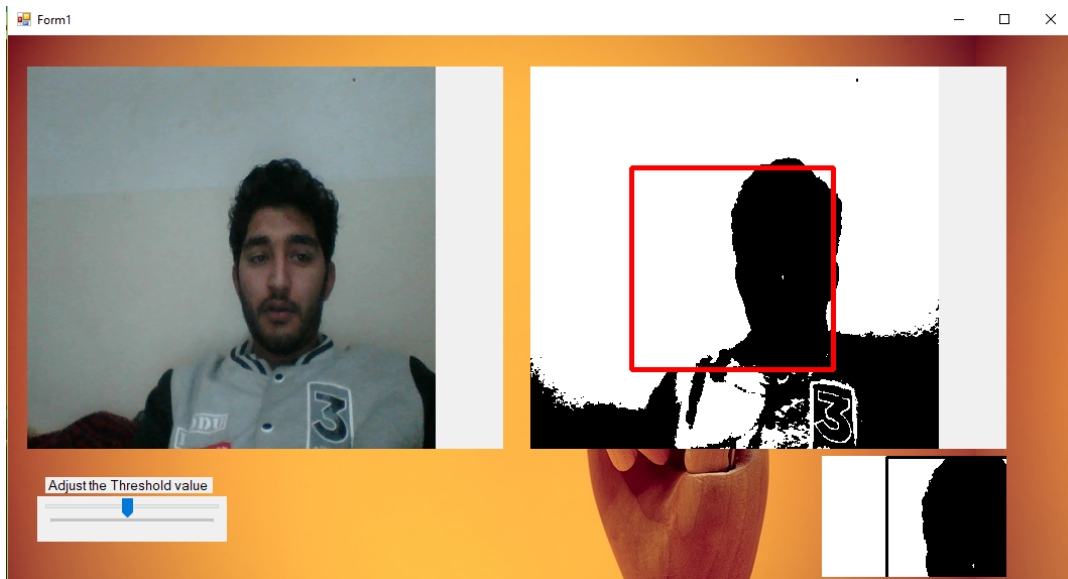


Figure 6.4: home page

**Description:** In figure 6.4 we can see there is no hand gesture and the system didn't display the result of this gesture.

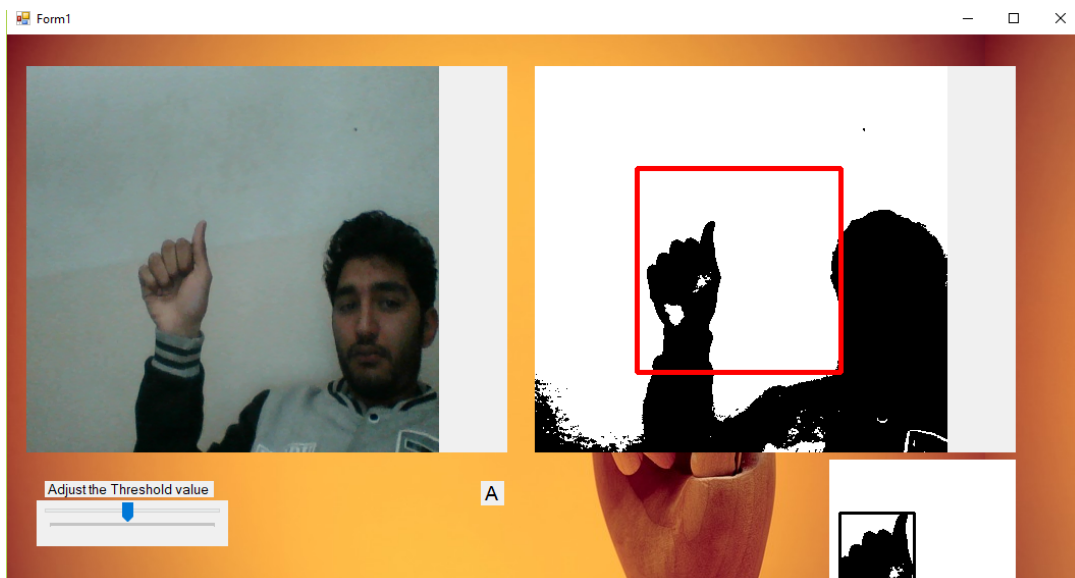


Figure 6.5: home page

**Description:** In figure 6.5 we can see the gesture of A.

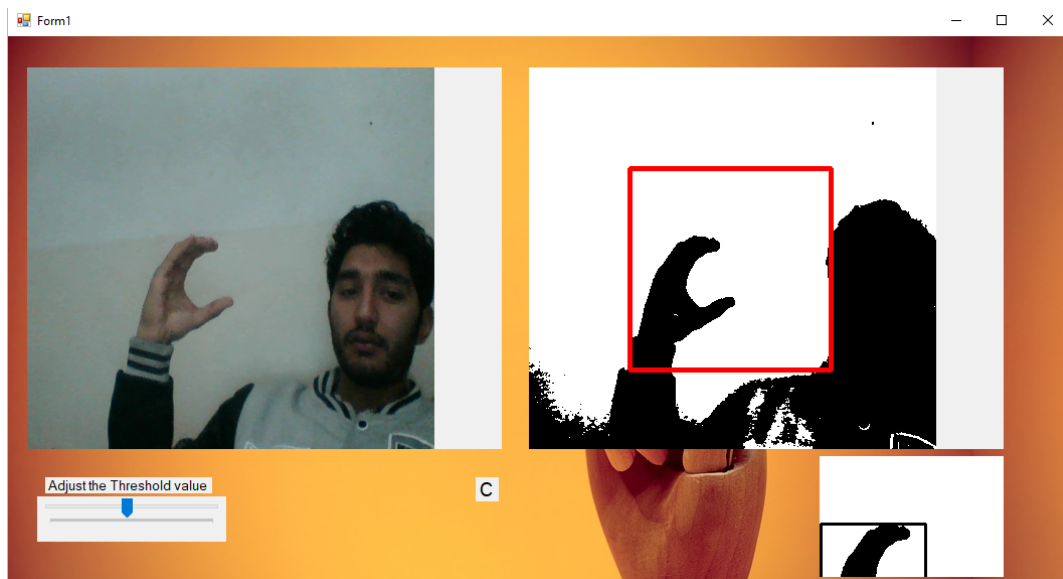


Figure 6.6: home page

**Description:** In figure 6.6 we can see the gesture of C.

### 6.3 Performance Testing

The software performance testing is castoff to evaluate the performance of the system. This technique yields the outcomes that how much reliable the system is. System performance testing carried out the following points.

- The system will take some in Hand detection as, in Hand detection, the system will process individual pixel of the frame.
- The system will take an adequate amount of time (3-4 minutes) for the training of the system.

### 6.4 Compatibility Testing

Compatibility testing is a form of non-functional testing which is used to evaluate if the system is compatible with the device it is going to be used on or not. The requirements to run HGR efficiently are mentioned below.

### 6.4.1 Device

- Intel Core i3, 2.8 GHz or equivalent.
- 4GB Ram
- 1 GB free storage space on disk.

### 6.4.2 Operating System

- Windows 7 or greater version.
- Microsoft Visual Studio 2010 or greater version.
- .NET framework 4.5 or greater.

## 6.5 Installation Testing

To run this system Microsoft OS must be installed on the device. This system is built on .NET framework 4.5, so .NET framework 4.5 or a greater version must be installed.

## 6.6 Test Cases

Test cases for the evaluation of HGR are displayed under.

### 6.6.1 Test Cases 1

The first test case states that the application should be installed on the device. For the application to be successfully installed, the device should be running on a Microsoft OS. To carry out this task, the user must copy the whole project into the desired location and run the setup file.

<b>Test Case Id</b>	TC-1
<b>Description</b>	The application is to be installed on Device.
<b>Application for</b>	Microsoft Operating Systems.
<b>Requirements</b>	The application should be installed in device having Microsoft Operating System
<b>Steps to be Taken</b>	copy the whole project to the respective location on device. run the setup file.
<b>Expected Result</b>	The application should be installed on device completely.
<b>Actual Result</b>	Application installed successfully.
<b>Status</b>	Success
<b>Remarks</b>	N/A

Figure 6.7: test case

### 6.6.2 Test Cases 2

The second test case states that the application should not be installed on the device. For the application to be successfully installed, the device should not be running on an Os except for the Microsoft OS.

Test Case Id	TC-2
Description	The application is not to be installed on Device.
Application for	Microsoft Operating Systems.
Requirements	The application should not be installed in device having opearting system other than Microsoft operating system
Steps to be Taken	copy the whole project to the respective location on device. run the setup file.
Expected Result	The complete application should not be installed on device.
Actual Result	Application installed unsuccessfully.
Status	Success
Remarks	N/A

Figure 6.8: test case

### 6.6.3 Test Cases 3

This test case states that in order to run the application, the user must have .NET Framework 4.5 or higher version running on the device.

Test Case Id	TC-3
Description	To run the application .NET framework 4.5 or greater must be installed on device.
Application for	Microsoft Operating Systems.
Requirements	.NET Framework 4.5 or greater should be installed on device to run application
Steps to be Taken	Run the application exe file from the respective location on device.
Expected Result	The complete application should be run on device.
Actual Result	Application run successfully.
Status	Success
Remarks	N/A

Figure 6.9: test case 3



#### 6.6.4 Test Cases 4

This test case provides the information if the device is not running on .NET Framework 4.5 or a greater version.

<b>Test Case Id</b>	<b>TC-4</b>
<b>Description</b>	<b>Device don't have .NET Framework installed.</b>
<b>Application for</b>	<b>Microsoft Operating Systems.</b>
<b>Requirements</b>	<b>The application should not be run in device.</b>
<b>Steps to be Taken</b>	<b>Run the application exe file from the respective location on device.</b>
<b>Expected Result</b>	<b>The complete application should not be run on device.</b>
<b>Actual Result</b>	<b>Application run unsuccessfully.</b>
<b>Status</b>	<b>Success</b>
<b>Remarks</b>	<b>N/A</b>

Figure 6.10: test case 4

#### 6.6.5 Test Cases 5

This test case provides the solution to a valid format of the input file

<b>Test Case Id</b>	<b>TC-5</b>
<b>Description</b>	<b>Input the file with valid format to the application.</b>
<b>Application for</b>	<b>Microsoft Operating Systems.</b>
<b>Requirements</b>	<b>The application should accept the file with the valid format.</b>
<b>Steps to be Taken</b>	<b>Press the Browse button and select the file.</b>
<b>Expected Result</b>	<b>The application should accept the file.</b>
<b>Actual Result</b>	<b>Application accepted the file successfully.</b>
<b>Status</b>	<b>Success</b>
<b>Remarks</b>	<b>N/A</b>

Figure 6.11: test case 5

## **Chapter 7**

# **Conclusions And Future Work**

The project, Hand Gesture Recognition, is intended for communication base system with deaf people which will assist them keyboard less application for typing for future use. The project is also applicable for intelligence agencies. The main objective of the project is to provide the facilities to the user to interact with the computer and we can also use this application for surgery purpose. It has been a prodigious and fruitful effort. After developing the project, we recognize the basic image processing and its application. We also gain a perspective on how the image processing is implemented on a desktop application. We have now a great idea on how the system testing is performed on a project and have also learned how to solve the problems generated by integrating the units of the system.



# Bibliography

- [1] Fernando Magno Quintão Pereira Roberto S. Bigonha Andrei Rimsa, Marcelo D'amorim. "efficient static checker for tainted variable attacks", February. Cited on p. 3.
- [2] Ahmed Toumanari. El Moussaid, Nadya ElBachir. Web application attacks detection: A survey and classification., nov 2014. Cited on p. 3.
- [3] K. Sathiyakumari. Krishnaveni. Multiclass classification of xss web page attack using machine learning techniques., November. Cited on p. 3.
- [4] Nuno F. Neves Medeiros, Ibéria and Miguel Correia. Automatic detection and correction of web application vulnerabilities using data mining to predict false positives. Cited on p. 3.
- [5] R. Sekar. An efficient black-box technique for defeating web application attacks, nov 2009. Cited on pp. 3 and 20.
- [6] Rajat Shrivastava. "a hidden markov model based dynamic hand gesture recognition system using opencv: Dept. of electronics and communication engineering maulana azad national institute of technology bhopal-462001, india.", February. Cited on p. 3.
- [7] Pujan Ziaie and Alois Knoll. Pujan ziaie and alois knoll. an invariant-based approach to static hand-gesture recognition: Technical university of munich. Cited on p. 3.