**SAVITRIBAI PHULE PUNE UNIVERSITY**

**A PRELIMINARY PROJECT REPORT ON**

**SIGN LANGUAGE TRANSLATOR USING 3D SENSOR**

SUBMITTED TOWARDS THE

PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

### BACHELOR OF ENGINEERING (Computer Engineering)

### BY

Sneha Arun Sonawane Exam No: BCM19074 Sakshi Ananda Borgude Exam No: BCM19013

Aditya Ramesh Kanawade Exam No: BCM19038 Sandesh Arun Shardul Exam No: BCM19068

**Under The Guidance of**

Prof.K.R.Patil



### DEPARTMENT OF COMPUTER ENGINEERING

**Late G.N. Sapkal College of Engineering**

**Anjeneri, Trimbakeshwar Nashik.**



### Late G.N. Sapkal College of Engineering DEPARTMENT OF COMPUTER ENGINEERING

**CERTIFICATE**

This is to certify that the Project Entitled

**SIGN LANGUAGE TRANSLATOR USING 3D SENSOR**

Submitted by

Sneha Arun Sonawane Exam No: BCM19074 Sakshi Ananda Borgude Exam No: BCM19013

Aditya Ramesh Kanawade Exam No: BCM19038 Sandesh Arun Shardul Exam No: BCM19068

It is a bonafide work carried out by Students under the supervision of Prof.K.R.Patil and it is submitted towards the partial fulfilment of the requirement of Bachelor of Engineering (Computer Engineering) Project.

Prof.K.R.Patil Dr.N.R Wankhede

Internal Guide H.O.D

Dept. of Computer Engg. Dept. of Computer Engg.

### Abstract

All over the world, deaf people use sign language as the only reliable source of communication with each other as well as with normal people. These communicating signs are made up of the shape of the hand and movement. In Pakistan, deaf people use Pakistan sign language (PSL) as a means of communication with people. In scientific literature, many studies have been done on PSL recognition and classification. Most of these work focused on colored-based hands while some others are sensors and Kinect-based approaches. These techniques are costly and also avoid user-friendliness. In this paper, a technique is proposed for the recognition of thirty-six static alphabets of PSL using bare hands. The dataset is obtained from the sign language videos. At a later step, four vision-based features are extracted i.e. local binary patterns, a histogram of oriented gradients, edge-oriented histogram, and speeded up robust features. The extracted features are individually classified using Multiple kernel learning (MKL) in support vector machine (SVM). We employed a one-to-all approach for the implementation of basic binary SVM into the multi-class SVM. A voting scheme is adopted for the final recognition of PSL. The performance of the proposed technique is measured in terms of accuracy, precision, recall, and F-score. The simulation results are promising as compared with existing approaches.

**Keywords-** Indian Sign Language, American Sign Language, Kinect Camera.

### Acknowledgments

*It gives us great pleasure in presenting the preliminary project report on* ***‘SIGN LANGUAGE TRANSLATOR USING 3D SENSOR’****.*

*I would like to take this opportunity to thank my internal guide* ***Prof.K.R.Patil*** *for giving me all the help and guidance I needed. I am really grate- ful to them for their kind support. Their valuable suggestions were very helpful.*

*I am also grateful to* ***Dr.N.R Wankhede****, Head of Computer Engineering Department, L.G.N.Sapkal College of Engineering,Anjeneri Nashik for his indispensable support, suggestions.*

*In the end our special thanks to* ***Vishal Mate sir*** *for providing various resources such as laboratory with all needed software platforms, contin- uous Internet connection, for Our Project.*

*Sneha..A.Sonawane*

*Sakshi.A.Borgude*

*Aditya.R.Kanawade*

*Sandesh.A.Shardul*

(B.E. Computer Engg.)

**INDEX**

1. [Synopsis](#_bookmark0) 1

[1.1 Project Title](#_bookmark1) . . . . . . . . . . . . . . . . . . . . . . . . 2

[1.2 Project Option](#_bookmark2) . . . . . . . . . . . . . . . . . . . . . . 2

[1.3 Internal Guide](#_bookmark3) . . . . . . . . . . . . . . . . . . . . . . . 2

[1.4 Sponsorship and External Guide](#_bookmark4) . . . . . . . . . . . . . 2

[1.5 Technical Keywords](#_bookmark5) . . . . . . . . . . . . . . . . . . . 2

[1.6 Problem Statement](#_bookmark6) . . . . . . . . . . . . . . . . . . . . 2

[1.7 Abstract](#_bookmark7) . . . . . . . . . . . . . . . . . . . . . . . . . . 3

[1.8 Goals and Objectives](#_bookmark8) . . . . . . . . . . . . . . . . . . . 3

* 1. [Relevant mathematics associated with the Project](#_bookmark9) . . . . 3
  2. [Names of Conferences / Journals where papers can be](#_bookmark10) [published](#_bookmark10) . . . . . . . . . . . . . . . . . . . . . . . . . 4
  3. [Review of Conference/Journal Papers supporting Project](#_bookmark11)

[idea](#_bookmark11) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

* 1. [Plan of Project Execution](#_bookmark13) . . . . . . . . . . . . . . . . . 5

1. [Technical Keywords](#_bookmark15) 7

[2.1 Area of Project](#_bookmark16) . . . . . . . . . . . . . . . . . . . . . . 8

[2.2 Technical Keywords](#_bookmark17) . . . . . . . . . . . . . . . . . . . . 8

1. [Introduction](#_bookmark18) 9
   1. [Project Idea](#_bookmark19) 10
   2. [Motivation of the Project](#_bookmark20) 10
   3. [Literature Survey](#_bookmark21) 10
2. [Problem Definition and scope](#_bookmark22) 13
   1. [Problem Statement](#_bookmark23) 14
      1. [Goals and objectives](#_bookmark24) 14
      2. [Statement of scope](#_bookmark25) 14
   2. [Software context](#_bookmark26) 14
   3. [Major Constraints](#_bookmark27) 15
   4. [Methodologies of Problem solving and efficiency issues](#_bookmark28) . 15 [4.5 Outcome](#_bookmark29) 16
   5. [Applications](#_bookmark30) 16
   6. [Hardware Resources Required](#_bookmark31) 16
   7. [Software Resources Required](#_bookmark33) 17
3. [Project Plan](#_bookmark34) 18
   1. [Project Estimates](#_bookmark35) 19
      1. [Reconciled Estimates](#_bookmark37) 19
      2. [PROJECT ESTIMATES USING COCOMO-II](#_bookmark39) [MODEL :](#_bookmark39) 21
      3. [Project Resources](#_bookmark41) 22
   2. [Risk Management w.r.t. NP Hard analysis](#_bookmark42) 22
      1. [Risk Identification](#_bookmark43) 23
      2. [Risk Analysis](#_bookmark44) 24
      3. [Overview of Risk Mitigation, Monitoring, Man-](#_bookmark45) [agement](#_bookmark45) 25
   3. [Project Schedule](#_bookmark47) 27
      1. [Project task set](#_bookmark48) 27
      2. [Task network](#_bookmark49) 27
      3. [Timeline Chart](#_bookmark51) 28
   4. [Team Organization](#_bookmark53) 28
      1. [Team structure](#_bookmark54) 28
   5. [Introduction](#_bookmark56) 29
      1. [Purpose and Scope of Document](#_bookmark57) 29
   6. [Usage Scenario](#_bookmark58) 29
      1. [User profiles](#_bookmark59) 29
      2. [Use-cases](#_bookmark60) 30
      3. [Use Case View](#_bookmark62) 30
   7. [Data Model and Description](#_bookmark64) 30
      1. [Data objects and Relationships](#_bookmark65) 30
   8. [Functional Model and Description](#_bookmark67) 31
      1. [Data Flow Diagram](#_bookmark68) 31
      2. [Description of functions](#_bookmark71) 33
      3. [Activity Diagram:](#_bookmark72) 33
      4. [Non Functional Requirements:](#_bookmark74) 34
      5. [State Diagram:](#_bookmark75) 34
      6. [Design Constraints](#_bookmark77) 35
      7. [Software Interface Description](#_bookmark78) 35
4. [Detailed Design Document using Appendix A and B](#_bookmark79) 36
   1. [Introduction](#_bookmark80) 37
   2. [Architectural Design](#_bookmark81) 37
   3. [Compoent Design](#_bookmark83) 38
      1. [Class Diagram](#_bookmark84) 38
5. [Summary and Conclusion](#_bookmark87) 40
6. [References](#_bookmark88) 42

[Annexure A Laboratory assignments on Project Analysis of Al-](#_bookmark89) [gorithmic Design](#_bookmark89) 44

[Annexure B Laboratory assignments on Project Quality and Re-](#_bookmark91) [liability Testing of Project Design](#_bookmark91) 48

[Annexure C Project Planner](#_bookmark92) 53

[Annexure D Reviewers Comments of Paper Submitted](#_bookmark93) 55

[Annexure E Plagiarism Report](#_bookmark94) 57

**List of Figures**

|  |  |  |
| --- | --- | --- |
| [1.1](#_bookmark12) | [Review of papers](#_bookmark12) . . . . . . . . . . . . . . . . . . . . . | 5 |
| [1.2](#_bookmark14) | [Spiral Model](#_bookmark14) . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| [5.1](#_bookmark36) | [Spiral Model](#_bookmark36) . . . . . . . . . . . . . . . . . . . . . . . | 19 |
| [5.2](#_bookmark38) | [Cost Estimate](#_bookmark38) . . . . . . . . . . . . . . . . . . . . . . . | 21 |
| [5.3](#_bookmark40) | [Time Estimates](#_bookmark40) . . . . . . . . . . . . . . . . . . . . . . | 22 |
| [5.4](#_bookmark46) | [Risk Impact Defination](#_bookmark46) . . . . . . . . . . . . . . . . . . | 26 |
| [5.5](#_bookmark50) | [Task Network](#_bookmark50) . . . . . . . . . . . . . . . . . . . . . . . | 28 |
| [5.6](#_bookmark52) | [Timeline Chart](#_bookmark52) . . . . . . . . . . . . . . . . . . . . . . | 28 |
| [5.7](#_bookmark55) | [Team structure](#_bookmark55) . . . . . . . . . . . . . . . . . . . . . . | 28 |
| [5.8](#_bookmark61) | [usecase](#_bookmark61) . . . . . . . . . . . . . . . . . . . . . . . . . . | 30 |
| [5.9](#_bookmark63) | [use case diagram](#_bookmark63) . . . . . . . . . . . . . . . . . . . . . | 30 |
| [5.10](#_bookmark66) | [ER diagram](#_bookmark66) . . . . . . . . . . . . . . . . . . . . . . . . | 31 |
| [5.11](#_bookmark69) | [DFD Level 0 diagram](#_bookmark69) . . . . . . . . . . . . . . . . . . . | 32 |
| [5.12](#_bookmark70) | [DFD Level 1 diagram](#_bookmark70) . . . . . . . . . . . . . . . . . . . | 32 |
| [5.13](#_bookmark73) | [Activity diagram](#_bookmark73) . . . . . . . . . . . . . . . . . . . . . | 33 |
| [5.14](#_bookmark76) | [State transition diagram](#_bookmark76) . . . . . . . . . . . . . . . . . . | 35 |
| [6.1](#_bookmark82) | [Architecture diagram](#_bookmark82) . . . . . . . . . . . . . . . . . . . | 38 |
| [6.2](#_bookmark85) | [Class Diagram](#_bookmark85) . . . . . . . . . . . . . . . . . . . . . . . | 38 |
| [6.3](#_bookmark86) | [Interaction Diagram](#_bookmark86) . . . . . . . . . . . . . . . . . . . . | 39 |
| [A.1](#_bookmark90) | [Idea Matrix](#_bookmark90) . . . . . . . . . . . . . . . . . . . . . . . . | 46 |
| [E.1](#_bookmark95) | [plagarism check 1](#_bookmark95) . . . . . . . . . . . . . . . . . . . . . | 58 |
| [E.2](#_bookmark96) | [plagarism check 2](#_bookmark96) . . . . . . . . . . . . . . . . . . . . . | 58 |

# List of Tables

[4.1 Hardware Resources](#_bookmark32) 1

## CHAPTER 1 SYNOPSIS

**1.1 PROJECT TITLE**

Sign Language Translator using 3D sensor.

**1.2 PROJECT OPTION**

Sponsered

**1.3 INTERNAL GUIDE**

Prof. K.R.Patil

**1.4 SPONSORSHIP AND EXTERNAL GUIDE**

Mr.Prakash Ekhande sir.

**1.5 TECHNICAL KEYWORDS**

1.Indian Sign Language.

2.American Sign Language

3.Kinect Camera

**1.6 PROBLEM STATEMENT**

To design and develop an application for impaired people which will help them to communicate with ordinary people and to also interact with the society. The application will take either audio input or actions based on the user of the system and produces its corresponding Output.

**1.7 ABSTRACT**

All over the world, deaf people use sign language as the only reliable source of communication with each other as well as with normal people. These communicating signs are made up of the shape of the hand and movement. In Pakistan, deaf people use Pakistan sign language (PSL) as a means of communication with people. In scientific literature, many studies have been done on PSL recognition and classification. Most of these work focused on colored-based hands while some others are sensors and Kinect-based approaches. These techniques are costly and also avoid user-friendliness. In this paper, a technique is proposed for the recognition of thirty-six static alphabets of PSL using bare hands. The dataset is obtained from the sign language videos. At a later step, four vision-based features are extracted i.e. local binary patterns, a histogram of oriented gradients, edge-oriented histogram, and speeded up robust features. The extracted features are individually classified using Multiple kernel learning (MKL) in support vector machine (SVM). We employed a one-to-all approach for the implementation of basic binary SVM into the multi-class SVM. A voting scheme is adopted for the final recognition of PSL. The performance of the proposed technique is measured in terms of accuracy, precision, recall, and F-score. The simulation results are promising as compared with existing approaches.

**Keywords**— Sign Language Translator using 3D sensor.

**1.8 GOALS AND OBJECTIVES**

• To promote the use of Indian Sign Language as educational mode for deaf students at primary, secondary and higher education levels.

• To collaborate with organizations of the deaf and other institutions in the field of disability to promote and propagate Sign Language.

• To develop manpower for using Indian Sign Language (ISL) and teaching and conducting research in ISL.

• Work in partnership with Deaf and Deafblind associations on sign language interpreting issues

**1.9 RELEVANT MATHEMATICS ASSOCIATED WITH THE PROJECT**

1. Algorithm 1 Kernel Selection Algorithm (KSA):

INPUT: feature vector set, fvk : k ∈ {SURF, LBP, HOG, EOH}, representing {Dataset Minus TestData}

OUTPUT: selected kernel function, SKFu : u ∈ {linear, Gaussian, polynomial}

1: initialize accuracy for each of the kernel as zero i.e., acc\_KFm = {0}

2: for (1 ≤ i ≤ 10) do

3: Randomly Divide fvk into training and validation subset i.e., fvk(train) , fvk(valid)

4: for (each KFm : m ∈ linear, Gaussian, polynomial) do

5: Train KFm based multiclass SVM over fvk(train)

6: Validate the model using fvk(valid) and compute its accuracy i.e., tempAcc\_KFm

7: acc\_KFm = acc\_KFm + tempAcc\_KFm

8: end for

9: end for

10: Compute average validation accuracy for each of the kernel,

KFm : m ∈ linear, Gaussian, polynomial, after 10 iterations

i.e., avgAcc\_KFm = acc\_KFm/10

11: Find kernel function corresponding to maximum average validation accuracy i.e., SKFu = MAX(avgAcc\_KFm : ∀m ∈ {linear, Gaussian, polynomial})

12: return SKFu

**1.10 NAMES OF CONFERENCES / JOURNALS WHERE PAPERS CAN BE PUBLISHED**

• International Journal for Research in Applied science and Engineering Technology (IJRASET)

**1.11 REVIEW OF CONFERENCE/JOURNAL PAPERS SUPPORTING PROJECT IDEA**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR.NO** | **Paper / Publication** | **Author** | **year** |
| **1.** | Sign Language Recognition Using Multiple Kernel Learning: A Case Study of Pakistan Sign Language | Awais Manzoor | 2017 |
| **2** | A Web-basedSign Language Translator Using 3D Video Processing | Kin Fun Li | 2011 |
| **3** | Sign language Translator using Microsoft Kinect XBOX 360. | Dr.Hairong Qi  Dr.Fabrice Meriaudeau | 2012 |
| **4** | ‘‘Shape based pakistan sign language categorization using  statistical features and support vector machines,’ | S. M. S. Shah, | 2018 |
| **5** | ‘‘An automatic arabic sign  language recognition system | N. B. Ibrahim | 2018 |

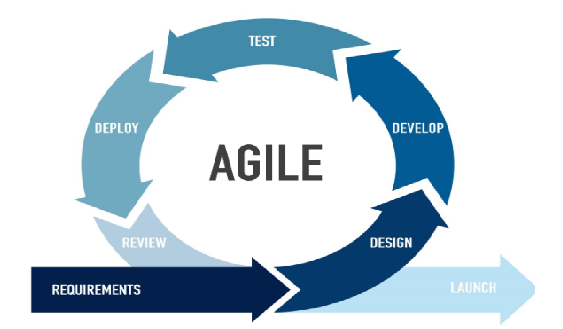
Figure 1.1: Review of papers

**1.12 PLAN OF PROJECT EXECUTION**

**Agile Model**

• Agile process model refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

• Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.



**Figure 1.2: Spiral Model**

The model starts with determining objectives and constraints of the software at the start of one iteration. Next phase is of prototyping the software. This includes risk analysis. Then one standard SDLC model is used to build the software. In the fourth phase of the plan of next iteration is prepared.

**CHAPTER 2**

**TECHNICAL KEYWORDS**

**2.1 AREA OF PROJECT**

Machine Learning.

**2.2 TECHNICAL KEYWORDS**

1. Indian Sign Language.

2. American Sign Language

3. Kinect Camera.

**CHAPTER 3**

**INTRODUCTION**

**3.1 PROJECT IDEA**

• Some investigators have argued that the failure to reduce the difference between deaf and hearing students' academic achievement indicates that we have not yet fully elucidated differences in what they know and how they learn, and thus we have been unable to adjust our instructional methods to match.

• This perspective has spawned renewed research into the cognitive underpinnings of learning in students with hearing loss. Meanwhile, other lines of research have focused on particular interventions designed to facilitate communication and learning in the classroom.

• Perhaps most apparent among these are recent efforts to provide deaf students with alternative or multiple forms of communication in the classroom.

**3.2 MOTIVATION OF THE PROJECT**

• According to the Research Annual Survey of Deaf and Hard-of-hearing Children and Youth, which included just over 40,000 children in the United States, only 27% of the children identified attended a special school or center for deaf children. This compares to 46% of the children who were fully mainstreamed in regular public school classrooms. The latter is likely to be an underestimate, however, as many of the children not identified by the survey would likely be those who are in general education classrooms where they are the only deaf or hard-of-hearing child. It is generally agreed by those in deaf education that more than 75% of deaf students at the K–12 level now are enrolled in regular education classrooms.

• At the postsecondary level, there are over 31,000 deaf and hard-of-hearing students enrolled in “mainstream” educational institutions in the United States, more than 90% of them attending classes with hearing peers. In fact, almost 50% of all 2- and 4-year institutions in the United States have identified themselves as serving at least one deaf or hard-of-hearing student, and among larger colleges and universities this number rises to around 95%.

**3.3 LITERATURE SURVEY**

• This system was successfully trained on all ISL static alphabets with a training accuracy of 99.93% and with testing and validation accuracy of 98.64%. Facial expression and context analysis are the other part not included in this project.

• Considering the challenges of sign language recognition, on targeted embedded platforms, authors have proposed the novel architecture of a binarized neural network with binary values of weights and activations using bitwise operations. The advantage is using this architecture achieves an overall accuracy of 98.8% which is higher than other existing methods while the disadvantage is This system misclassifies some signs of M, N, E because of their similar kind of shapes, and also, the proposed BNN architecture is limited with small no of classes of gestures.

• In this system it takes input through in-built web camera. They have used Camshift method for Hand tracking and Genetic Algorithm for gesture recognition. Then final result is converted into text and voice. The proposed system consists of 4 modules: Hand Tracking, Segmentation, Feature Extraction and Gesture Recognition.

1. Hand Tracking: It involves tracing of hand gestures using camshift algorithm.

2. Segmentation: The purpose behind the HSV Color model is to segment the hands from the background.

3. Feature Extraction: Several general-purpose features are extracted and relationship between features and classes is inferred by an appropriate classifier.

4. Gesture Recognition: After extracting features of the input character, features are searched into the database and consider the most similar features as the result. Then Genetic algorithm is used for hand gesture analysis.

**CHAPTER 4**

**PROBLEM DEFINITION AND SCOPE**

**4.1 PROBLEM STATEMENT**

To design and develop an application for impaired people which will help them to communicate with ordinary people and to also interact with the society. The application will take either audio input or actions based on the user of the system and produces its corresponding Output.

**4.1.1 Goals and objectives**

Goal and Objectives:

• To promote the use of Indian Sign Language as educational mode for deaf students at primary, secondary and higher education levels.

• To collaborate with organizations of the deaf and other institutions in the field of disability to promote and propagate Sign Language.

• To develop manpower for using Indian Sign Language (ISL) and teaching and conducting research in ISL.

• Work in partnership with Deaf and Deafblind associations on sign language interpreting issues

**4.1.2 Statement of scope**

• In this system it takes input through in-built web camera. They have used Camshift method for Hand tracking and Genetic Algorithm for gesture recognition. Then final result is converted into text and voice. The proposed system consists of 4 modules: Hand Tracking, Segmentation, Feature Extraction and Gesture Recognition.

1. Hand Tracking: It involves tracing of hand gestures using camshift algorithm.

2. Segmentation: The purpose behind the HSV Color model is to segment the hands from the background.

3. Feature Extraction: Several general-purpose features are extracted and relationship between features and classes is inferred by an appropriate classifier.

4. Gesture Recognition: After extracting features of the input character, features are searched into the database and consider the most similar features as the result. Then Genetic algorithm is used for hand gesture analysis.

**4.2 SOFTWARE CONTEXT**

1. Deaf to Normal

It convert the hand gesture to the text.

2. Normal to Deaf

It convert the voice message into the sign.

**4.3 MAJOR CONSTRAINTS**

1. Internet connectivity:

We require working internet connection for sending real time noti- fication.

2. Hardware failure:

We need system with good hardware configuration if system hard- ware configuration is not good there may be the chances that system may collapse.

3. Privacy security:

Here we are using open source hardware and software so there is no need of security mechanism.

**4.4 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY IS- SUES**

• The first part consist of Kinect Camera Sensor which is Connected to the laptop for

capturing the hand gesture and controlling purpose.

• In Mode I the system is initiated by the impaired person. When impaired person want to communicate with normal person then he gives the input as sign to System through Kinect camera. These actions are captured by the kinect camera. System performs Processing on captured sign and produces text and audio as output. That is understood by normal people.

• In mode 2 the system is initiated by the normal person. When normal person wants to communicate with impaired person then he gives input as audio. This input is accepted by the microphone which is fitted on the base of kinect camera.

• System performs processing on audio and converts it into text. Based on this text pattern matching is done. Once the match is found then the relevant images are displayed. In this way the system converts sign to text and vice versa.

**4.5 OUTCOME**

• System to be developed here is consisting of shutter lock an alarm which leads to catch the thieves at real time and real time notifica- tion and alert will be sent to authority.

• With the wide use of internet of things system is focused to im- plement the internet technology to establish a system which would communicate through internet for ATM secure monitoring system Internet of things is expected to rule the world in various fields.

**4.6 APPLICATIONS**

• ATM Security.

• Banking Locker Security.

• Other private security.

**4.7 HARDWARE RESOURCES REQUIRED**

1. Kinect Camera:

(a) Kinect sensor captures the human body and draw the skeleton of it. (b) Provide accuracy while performing Actions. (e) Kinect XBOX 360 captures the Global points of human body.

2. Operating System

(a) Windows 7 and Windows 8 are supported.

**4.8 SOFTWARE RESOURCES REQUIRED**

1. Kinect Studio v1.8.0

This studio helps to make connectivity of Kinect camera with C sharp net, this Studio also provides Developer Toolkit Browser v1.8.0. In which sample template is provided, in this built-in packages and classes are given. This template can be used for Kinect learning.

2. C#. net

To design front end of this System WPF (Windows Presentation Foundation) is

used, and Implementation is done in C sharp net.

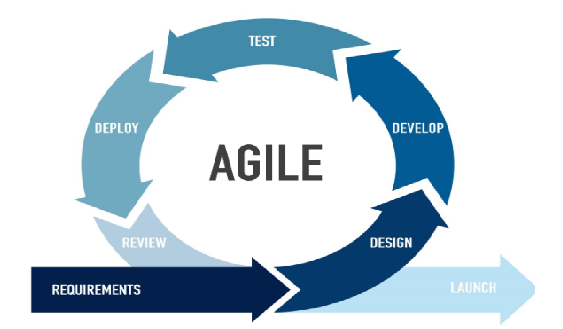
**CHAPTER 5**

**PROJECT PLAN**

**5.1 PROJECT ESTIMATES**

• Agile process model refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

• Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.



**Figure 5.1: Agile Model**

**5.1.1 Reconciled Estimates**

Cost Estimate

• COCOMO (COnstructive COst MOdel):

The Constructive Cost Model is a procedural software cost esti- mation model developed by Barry W. Boehm. This application de- rives the COCOMO software engineering metric as found in Robert

Pressman’s “Software Engineering, A Practitioner’s Approach”, (McGraw- Hill,97). The specific version utilized here is the “basic” model.

• PURPOSE:

The purpose of this site is to help the end user making an estimate using the COCOMO model and verify the same.

• INSTRUCTIONS:

Enter the lines of code in KLOC as per the type of the project and estimated lines of code.

The result in the below table shows different values of Effort, Du- ration, and Staffing based on the type of the project.

The values of a,b,c,d are also shown in the table.

• ORGANIC:

Relatively small, simple software projects in which small teams with good application experience work to a set of less than rigid requirements.

• SEMI-DETACHED:

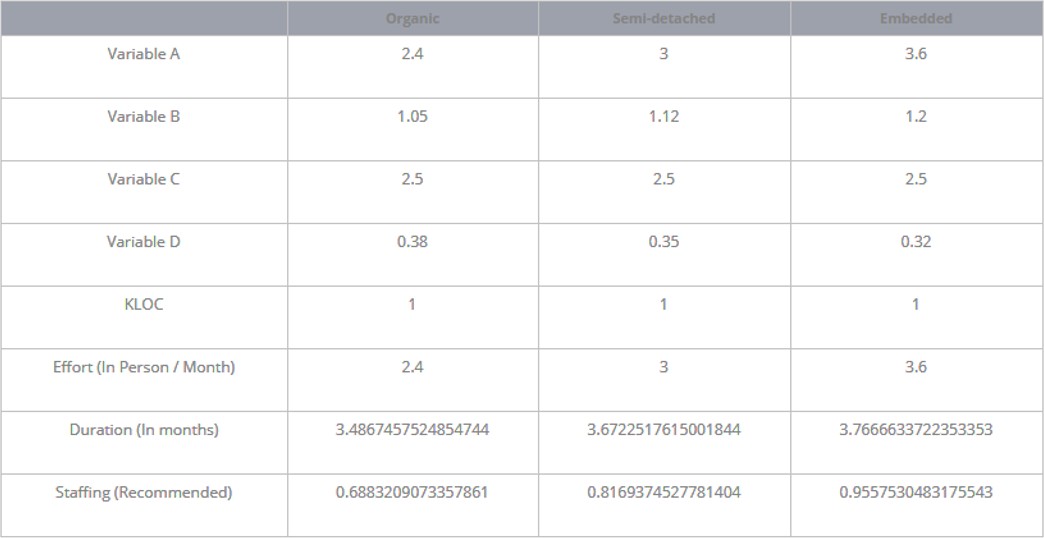
An intermediate, (in size and complexity), a software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

• EMBEDDED:

A software project that must be developed within a set of tight hard- ware, software and operation constraints.

Enter estimated kilo line of codes : 1.5 Effort = a\*locb

Duration = c\*effortd Staffing = effort/duration



**Figure 5.2: Cost Estimate**

**5.1.2 PROJECT ESTIMATES USING COCOMO-II MODEL :**

Costing is the proposed or estimated cost of producing or undertaking something of a software product. It can be calculated using COCOMO, COCOMO II models. For costing of our system, Basic COCMO model is used. The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project charac- teristics. Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in esti- mated thousands of source lines of code (SLOC). COCOMO applies to three classes of software projects:

• Organic projects - ”small” teams with ”good” experience working with ”less than rigid” requirements.

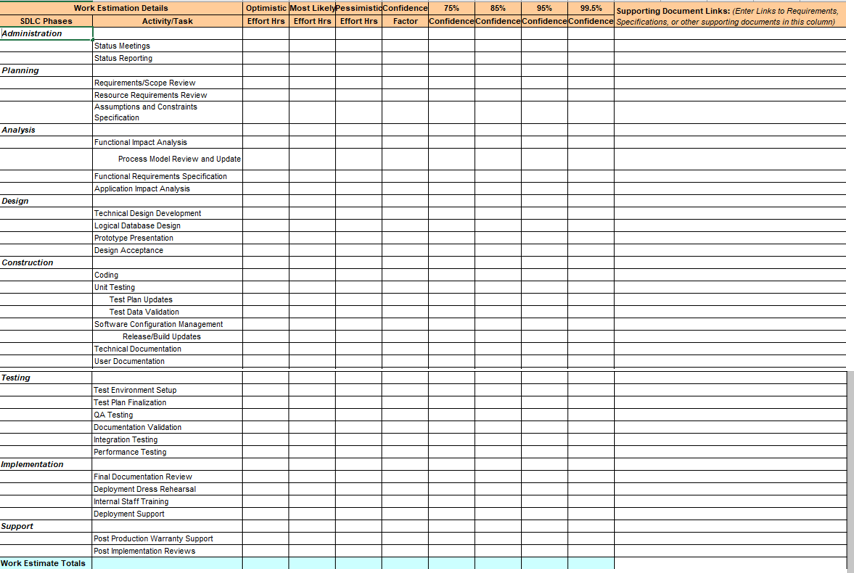
• Semi-detached projects - ”medium” teams with mixed experience working with a mix of rigid and less than rigid requirements.

• Embedded projects - developed within a set of ”tight” constraints.

It is also combination of organic and semi-detached projects.(hardware, software, operational, ...).

**5.1.2.1 Time Estimates**

**Format Sem-I/time.png**



**Figure 5.3: Time Estimates**

**5.1.3 Project Resources**

Project resources [People, Hardware, Software, Tools and other resources] based on Memory Sharing, IPC, and Concurrency derived using appen- dices to be referred.

**5.2 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS**

NP-hard (Non-deterministic Polynomial-time hard), in computational complexity theory, is a class of problems that are, informally, ”at least as hard as the hardest problems in NP”. More precisely, a problem H is NP- hard when every problem L in NP can be reduced in polynomial time to

H. As a consequence, finding a polynomial algorithm to solve any NP- hard problem would give polynomial algorithms for all the problems in NP, which is unlikely as many of them are considered hard.

A common mistake is to think that the NP in NP-hard stands for non- polynomial. Although it is widely suspected that there are no polynomial- time algorithms for NP-hard problems, this has never been proven. More- over, the class NP also contains all problems which can be solved in polynomial time.

**Definition**

A decision problem H is NP-hard when for any problem L in NP, there is a polynomial-time reduction from L to H An equivalent definition is to require that any problem L in NP can be solved in polynomial time by an oracle machine with an oracle for H. Informally, we can think of an algorithm that can call such an oracle machine as a subroutine for solving H, and solves L in polynomial time, if the subroutine call takes only one step to compute. Another definition is to require that there is a polynomial-time reduction from an NP-complete problem G to H. As any problem L in NP reduces in polynomial time to G, L reduces in turn to H in polynomial time so this new definition implies the previous one. It does not restrict the class NP-hard to decision problems, for instance it also includes search problems, or optimization problems.

Consequences

• If P NP, then NP-hard problems cannot be solved in polynomial time;

• If an optimization problem H has an NP-complete decision version L, then H is NP-hard.

• our system comes in the category of NP HARD problem as:

If hardware gets crashed then system fails If internet connection is lost then real time notification cannot be sent

**5.2.1 Risk Identification**

Risk analysis and management are the series of steps that help a project team to understand and manage uncertainty. The team establishes a plan for managing the risks. The primary objective to avoid risk, the team works to develop a contingency plan that will enable it to respond in a controlled and effective manner. In our project complexity of the project and hardware uncertainty were dened as major project risk factor.

Each risk is categorized as per the categories mentioned in . Please refer table 5.4for all the risks.

• Technical Risks:

The threaten the quality and timeliness of the software/hardware to be produced. If the technical risks become reality, implementa- tion may become difcult or impossible .Technical risks involved in the project are potential design, implementation, interfacing, veri- cation and maintained problems.

• Software Risks:

Software risks involve the characteristics like uncertainty and the loss. Uncertainty is the event that characterize the risk may or may not happen i.e there are no 100.

• Business Risks:

This threaten the viability of the system to be build. Business risks often jeopardize the project or the product. These risks are simply unpredictable in advance. The project may suffer from strategic risk meaning that product may no longer fit into the overall business strategy for the company.

**5.2.2 Risk Analysis**

Risk Analysis is very essential for software testing. In software test- ing, risk analysis is the process of identifying risks in applications and prioritizing them to test. A risk is a potential for loss or damage to an organization from materialized threats. Risk Analysis attempts to iden- tify all the risks and then quantify the severity of the risks. A threat as we have seen is a possible damaging event. If it occurs, it exploits vulnerability in the security of a computer based system.

Items with higher risk values should be tested early and often.

Items with lower risk value can be tested later, or not at all.

• How to perform Risk Analysis during software testing

When a test plan has been created, risks involved in testing the product are to be taken into consideration along with the possibil- ity of their occurrence and the damage they may cause along with solutions; if any. Detailed study of this is called Risk Analysis.

Some of the risks could be:

1. New Hardware

2. New Technology

3. New Automation Tool

4. Sequence of code delivery

5. Availability of application test resources

Risk assessment may be the most important step in the risk man- agement process, and may also be the most difficult and prone to error. Once risks have been identified and assessed, the steps to properly deal with them are much more programmatically.

Part of the difficulty of risk management is that measurement of both of the quantities in which risk assessment is concerned can be very difficult itself. Uncertainty in the measurement is often large in both cases. Also, risk management would be simpler if a single metric could embody all of the information in the measurement. However, since two quantities are being measured, this is not possible. A risk with a large po- tential loss and a low probability of occurring must be treated differently than one with a low potential loss but a high likelihood of occurring. In theory both are of nearly equal priority in dealing with first, but in practice it can be very difficult to manage when faced with the scarcity of resources, especially time, in which to conduct the risk management process. Expressed mathematically,

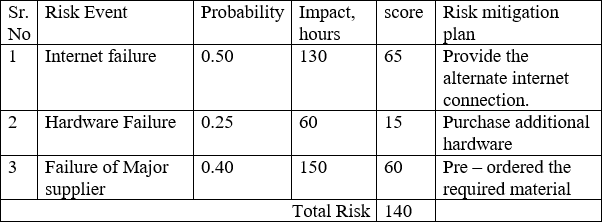
Financial decisions, such as insurance, often express loss terms in dollars. When risk assessment is used for public health or environmental decisions, there are differences of opinions as to whether the loss can be quantified in a common metric such as dollar values or some numerical measure of quality of life. Often for public health and environmental decisions, the loss term is simply a verbal description of the outcome, such as increased cancer incidence or incidence of birth defects. In that case, the ”risk” is expressed.

If the risk estimate takes into account information on the number of individuals exposed, it is termed a ”population risk” and is in units of expected increased cases per a time period. If the risk estimate does not take into account the number of individuals exposed, it is termed an ”individual risk” and is in units of incidence rate per a time period. Pop- ulation risks are of more use for cost/benefit analysis; individual risks are of more use for evaluating whether risks to individuals are ”acceptable”.

**5.2.3 Overview of Risk Mitigation, Monitoring, Management**

Following are the details for each risk.

**Format Sem-I/Risk impact.png**



**Figure 5.4: Risk Impact Defination**

|  |  |
| --- | --- |
| Risk ID | 1 |
| Risk Description | Internet connectivity |
| Category | Development Environment. |
| Source | Software Requirement Specification . |
| Probability | Moderate |
| Impact | High |
| Response | Accept |
| Strategy | Provide alternate connection |
| Risk Status | Identified |

|  |  |
| --- | --- |
| Risk ID | 2 |
| Risk Description | Hardware Failure |
| Category | Development Environment |
| Source | Software Requirement Specification |
| Probability | Moderate |
| Impact | Low |
| Response | Accept |
| Strategy | Purchase additional Hardware |
| Risk Status | Identified |

|  |  |
| --- | --- |
| Risk ID | 3 |
| Risk Description | Failure of major supplier |
| Category | Development Environment |
| Source | Software Requirement Specification |
| Probability | Low |
| Impact | High |
| Response | Accept |
| Strategy | Pre-ordered the require material |
| Risk Status | Identified |

**5.3 PROJECT SCHEDULE**

**5.3.1 Project task set**

Major Tasks in the Project stages are:

• Task 1: Topic Finalization

• Task 2: Market Analysis

• Task 3: Requirement Gathering

• Task 4: Detailed Design

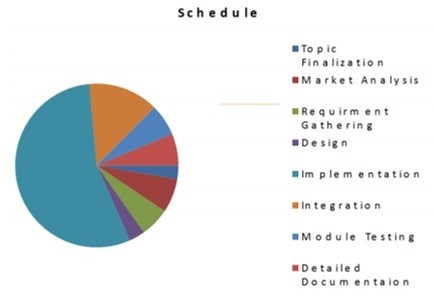
• Task 5: Implementation

• Task 6: Module Testing

• Task 8: Detailed Documentation

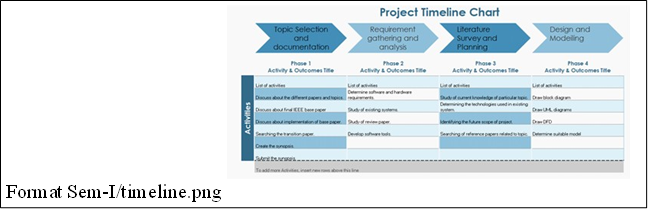
5.3.2 Task network

Project tasks and their dependencies are noted in this diagrammatic form.



**Figure 5.5: Task Network**

**5.3.3 Timeline Chart**



**Figure 5.6: Timeline Chart**

**5.4 TEAM ORGANIZATION**

**5.4.1 Team structure**

The team structure for the project is identified. Roles are defined.

|  |  |  |
| --- | --- | --- |
| Sr. No. | Name | Exam No |
| 1 | Sneha Arun Sonawane | BCM19074 |
| 2 | Sakshi Ananda Borgude | BCM19013 |
| 3 | Aditya Ramesh Kanawade | BCM19038 |
| 4 | Sandesh Arun Shardul | BCM19068 |

**Figure 5.7: Team structure**

**5.5 INTRODUCTION**

**5.5.1 Purpose and Scope of Document**

The main objective of this project is to recognizing the gestures and displaying the correspondent word. The first phase involves capturing the gesture using a webcam along with pose estimation library. The webcam captures the image and image is processed with pose estimation algorithm in tensor-flow utility. how the webcam is reading the image and the skeleton mapped on the image is the result of the pose estimation library. The skeleton obtained provides the values for creating the data set; the data set is a collection of the values of the coordinates of the end points of the skeleton. These values are labelled accordingly and are appended to the machine for predicting when the input is taken. The block diagram explains how the work is carried out in the system. webcam along with pose estimation library. The webcam captures the image and image is processed with pose estimation algorithm in tensor flow utility. how the webcam is reading the image and the skeleton mapped on the image is the result of the pose estimation library. The skeleton obtained provides the values for creating the data set; the data set is a collection of the values of the coordinates of the end points of the skeleton. These values are labelled accordingly and are appended to the machine for predicting when the input is taken.

**5.6 USAGE SCENARIO**

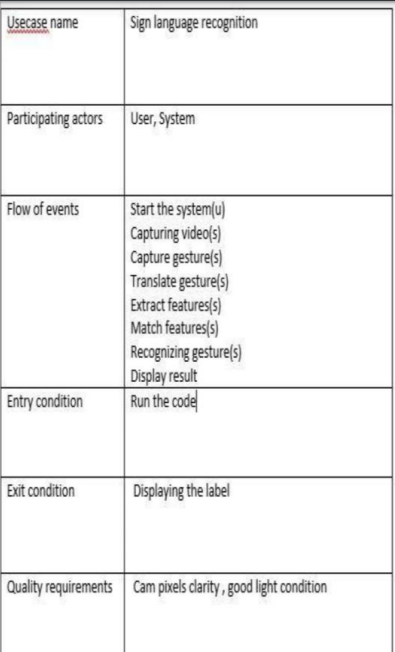
• To communicate with deaf peoples:

To communicate or do partnership work with deaf people.

**5.6.1 User profiles**

Admin: When any deaf person try to communicate with admin then admin will control and monitor the system.

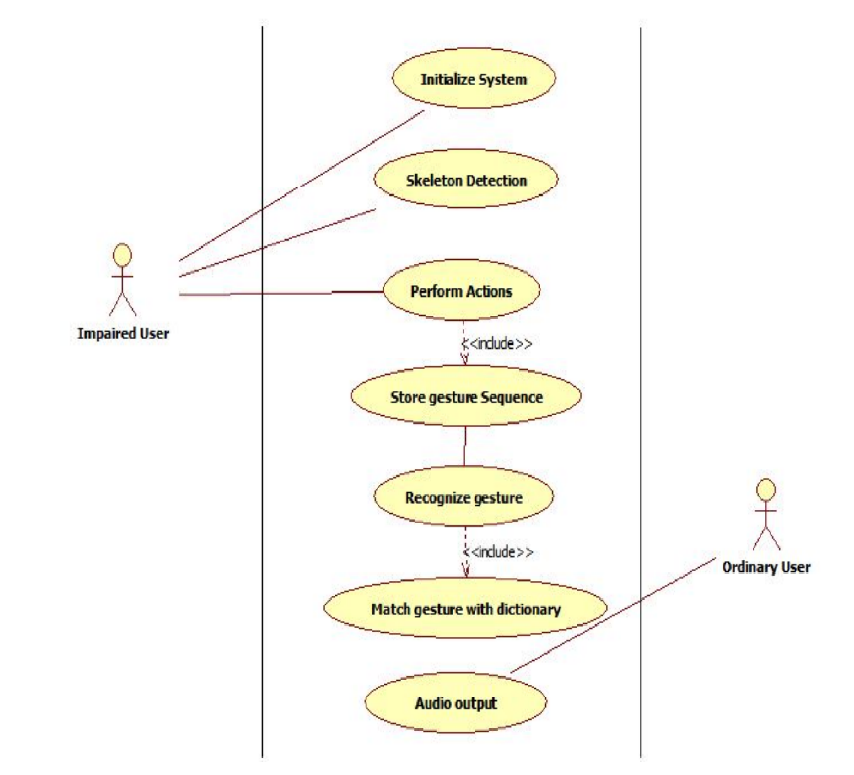
**5.6.2 Use-cases**

****

**Figure 5.8: use case**

**5.6.3 Use Case View**

Use Case Diagram. Example is given below

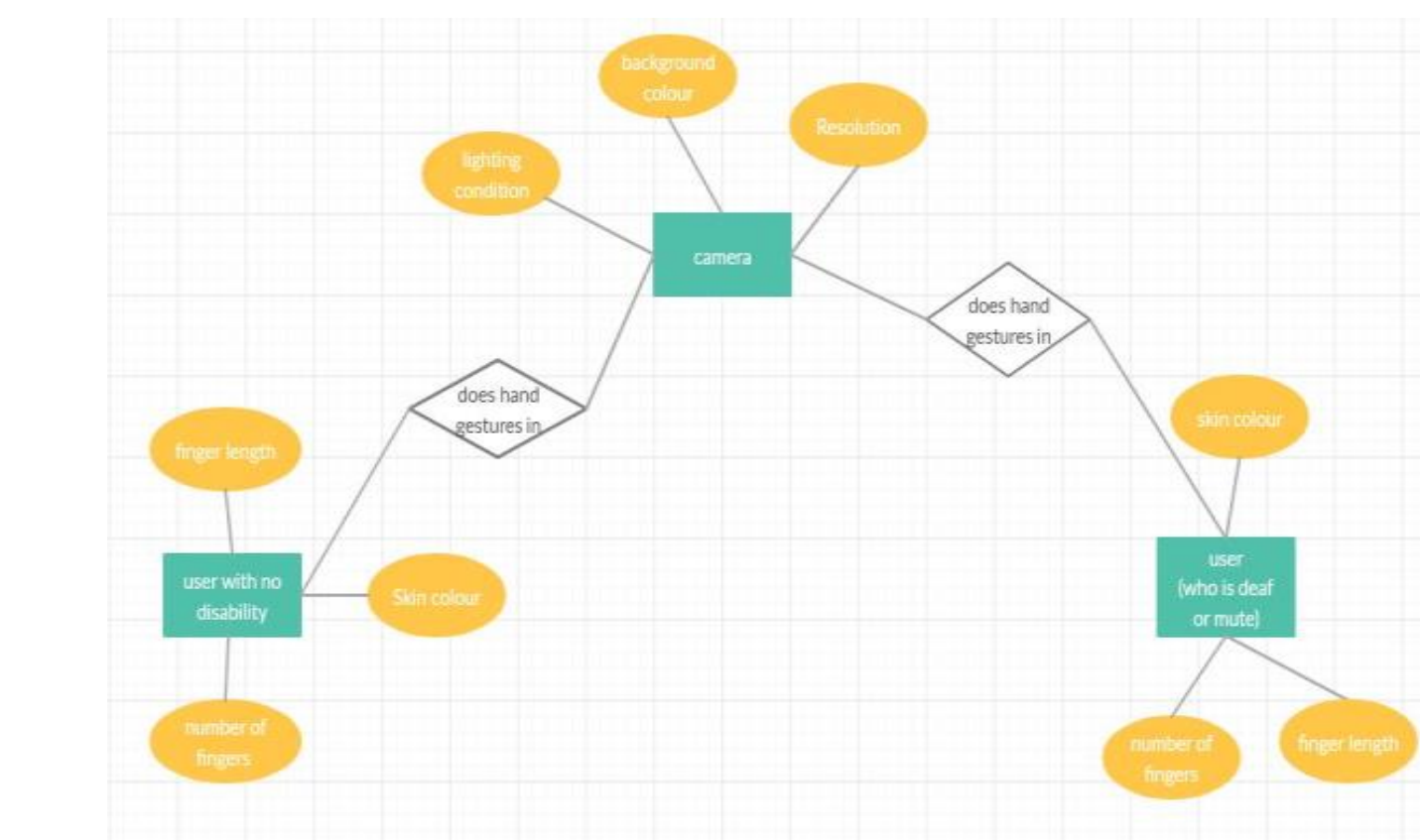


**Figure 5.9: use case diagram**

**5.7 DATA MODEL AND DESCRIPTION**

**5.7.1 Data objects and Relationships**

Data objects and their major attributes and relationships among data ob- jects are described using an ERD- like form.



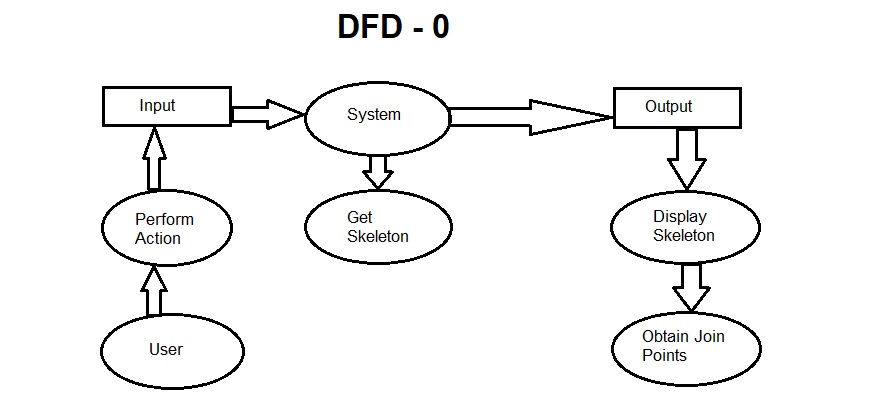
**Figure 5.10: ER diagram**

**5.8 FUNCTIONAL MODEL AND DESCRIPTION**

**5.8.1 Data Flow Diagram**

**5.8.1.1 Level 0 Data Flow Diagram**

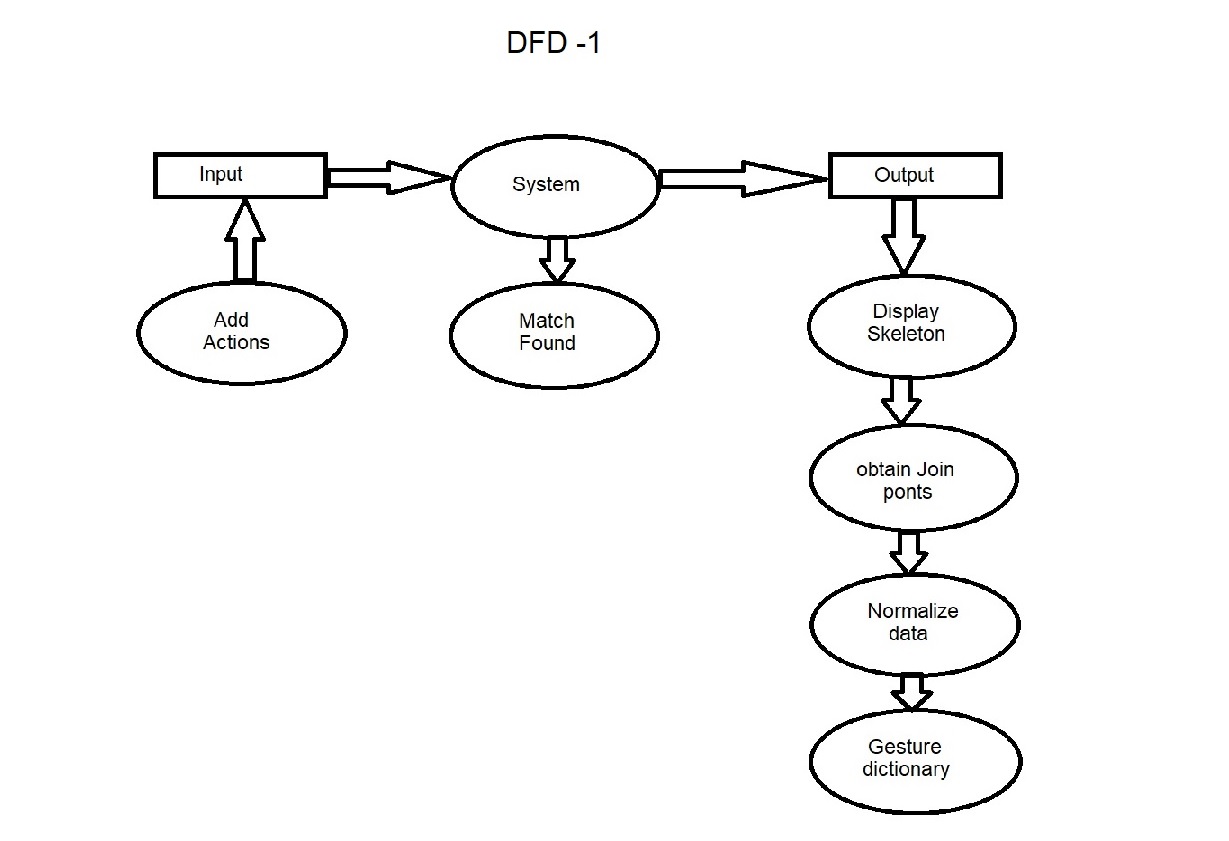
It is also known as context diagram. Its designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.



**Figure 5.11: DFD Level 0 diagram**

**5.8.1.2 Level 1 Data Flow Diagram**

In 1-level DFD, context diagram is decomposed into multiple bubbles/processes.in this level we highlight the main functions of the system and breakdown the high level process of 0-level DFD into subprocesses.



**Figure 5.12: DFD Level 1 diagram**

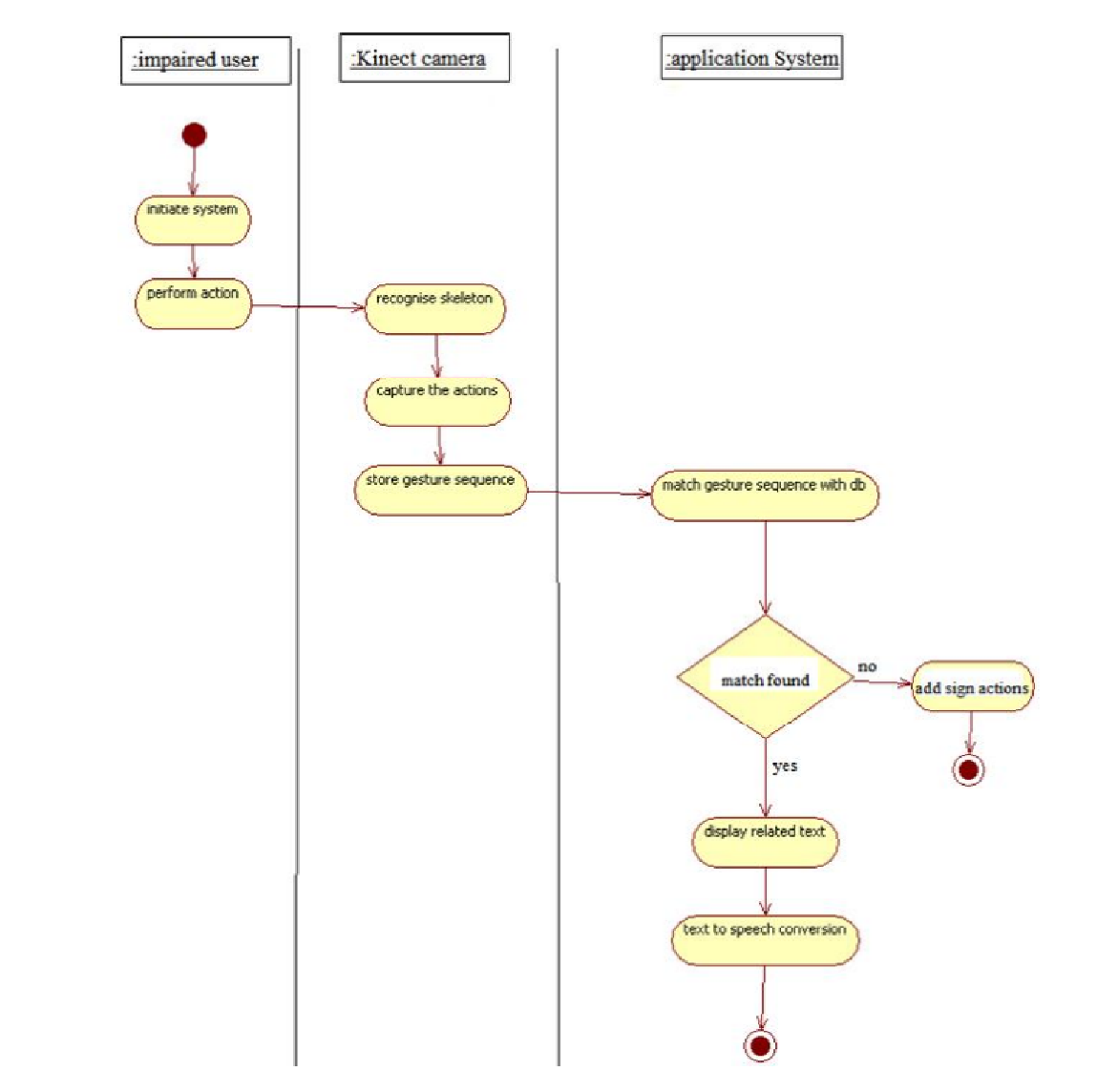
**5.8.2 Description of functions**

A description of each software function is presented. A processing narra- tive for function n is presented.(Steps)/ Activity Diagrams. For Example Refer 5.12

**5.8.3 Activity Diagram:**

• The Activity diagram represents the steps taken:

We use Activity Diagrams to illustrate the ow of control in a sys- tem. We can also use an activity diagram to refer to the steps in- volved in the execution of a use case. We model sequential and con- current activities using activity diagrams. So, we basically depict workows visually using an activity diagram. an activity diagram fo- cuses on condition of ow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.



**Figure 5.13: Activity diagram**

**5.8.4 Non Functional Requirements:**

• Interface Requirements

The Interface requirements either consist of the Application or Any Interface Application.

• Performance Requirements

System should notify in real-time without any delay.

• Safety Requirement

Container should not be dip in water.

• Security Requirements

Here we are using opensource hardware and software so there is no need of security mechanism.

• Software quality attributes:

– Reliability

The reliability that anyone can buy the product and install eas- ily.

– Availability

System will available at all time until battery or internet prob- lem. If we provide backup power supply then we can say that our system availability is 100 percent.

– Maintainability

Our system usually not required maintenance until some mod- ification to be done.

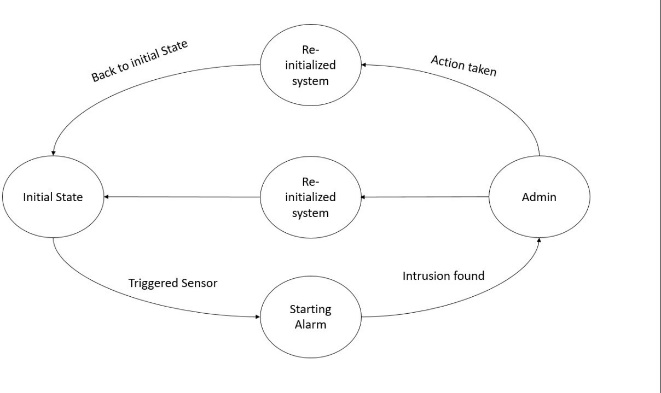
– Portability

The application is works on windows, iOS and Linux platform.

**5.8.5 State Diagram:**

State Transition Diagram

Fig.5.11 example shows the state transition diagram of ATM theft detec- tion system. The states are represented in ovals and state of system gets changed when certain events occur. The transitions from one state to the other are represented by arrows. The Figure shows important states and events that occur while creating new project



**Figure 5.14: State transition diagram**

**5.8.6 Design Constraints**

ESP8266 Development Board (NODEMCU) ,sensors,power supply,Actuators (Siren, Solenoid Lock, Shutter Motor.)

**5.8.7 Software Interface Description**

• User Interfaces

The user must have an Android or iOS application installed in Authorized person’s Pc/Laptop/Tablet with internet connection or Desktop with any operating system.

• Hardware Interfaces

Hardware interfaces needed except smartphone or computer system with wi-fi.

• Software Interfaces

Our project is based on Windows or Linux android or iOS operating system so user must have this operating system.

• Communications Interfaces

The internet connection plays important role which is sending the notification.

**CHAPTER 6**

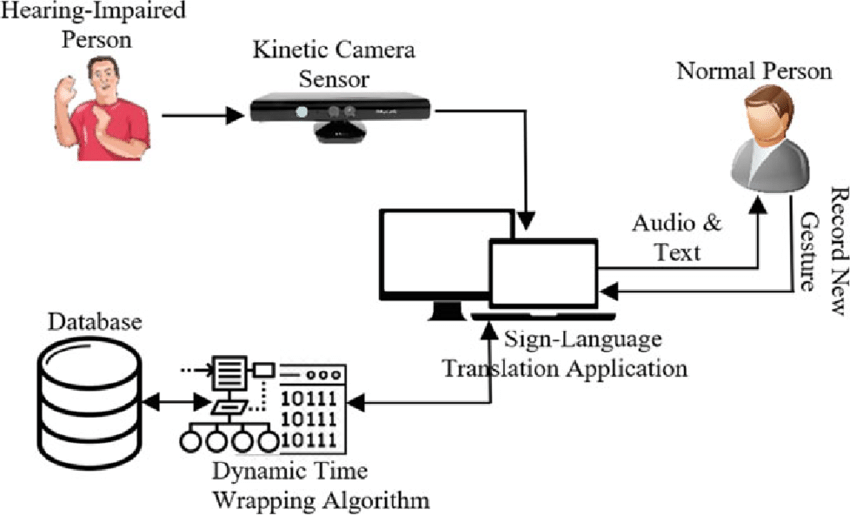
**DETAILED DESIGN DOCUMENT USING APPENDIX A AND B**

**6.1 INTRODUCTION**

Sign language is the way of communication and interaction for deaf people all around the world. This kind of communication is accomplished over some hand gestures, facial expressions, or movement of arm/body. The sign language recognition system aims to enable the deaf community to communicate with normal society appropriately. It is a highly structured symbolic set that provides the human-computer interaction (HCI). Sign language is very beneficial as a communication tool, and every day millions of deaf people around the world use sign language to communicate and express their ideas. This facilitation and assistance to deaf persons enable and encourage them to be a healthy part of society and integrate them into society. As you move from one country/ region to another country/region, sign language changes like American, Japanese, Chinese, and Arabic sign language. Pakistan has the sign language known as Pakistan sign language (PSL), and the alphabets of PSL are the alphabets of the national language of Pakistan ‘Urdu’. Sign language has two broad categories which include static and dynamic sign language. In static, a fixed pose of the hand is considered while dynamic sign language includes motion or movement of the hand. In past, many researchers presented several approaches for the recognition of different sign languages, used all around the world. Most of the work has been done for American sign language, Chinese sign language, Arabic sign language.

**6.2 ARCHITECTURAL DESIGN**

A description of the program architecture is presented. Subsystem de- sign or Block diagram,Package Diagram,Deployment diagram with de- scription is to be presented.

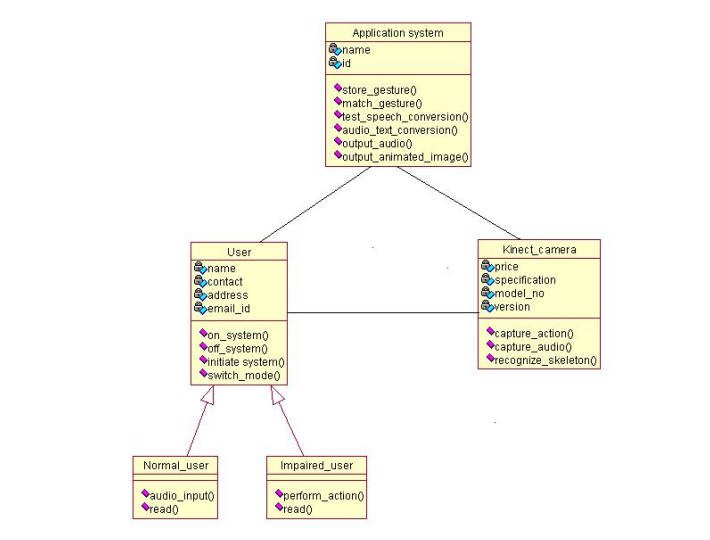


**Figure 6.1: Architecture diagram**

**6.3 COMPOENT DESIGN**

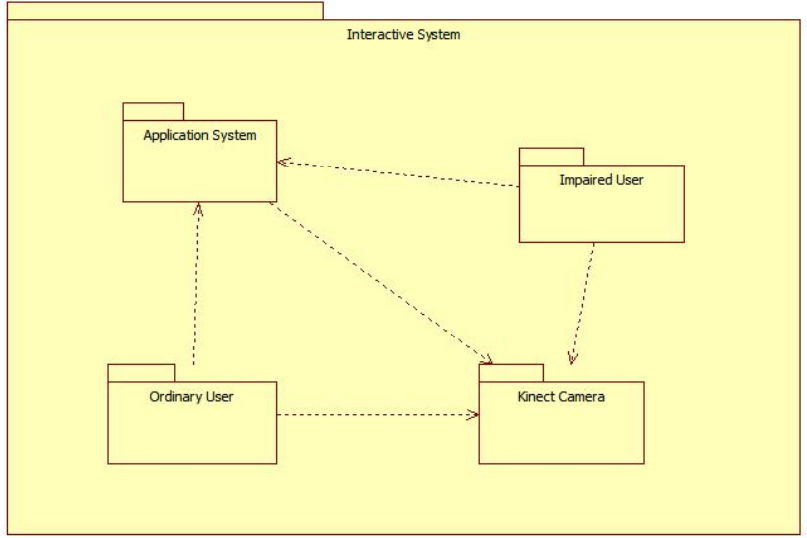
Most widely use UML diagram is the class diagram. It is the building block of all object oriented software systems. We use class diagrams to depict the static structure of a system by showing systems classes, their methods and attributes. Class diagrams also help us identify relationship between dierent classes or objects.

**6.3.1 Class Diagram**

****

**Figure 6.2: Class Diagram**

**6.3.1.1 Interaction Diagrams**



**Figure 6.3: Interaction Diagram**

**CHAPTER 7 SUMMARY AND CONCLUSION**

The proposed Interactive System can handle different types of words. Also, it is suitable for dynamic signs. This system helps for easier interaction and communication with impaired people. It acts as mediator between impaired User and Ordinary User. They can easily convey the messages to each other by this System. While communicating, User can also add as many signs into the dictionary along with its corresponding meaning. The experimental results show that the system is working system for native Indian sign language recognition. The system is designed to support recognition of words in ISL only.

This proposed system can be enhanced to recognize for continuous sentences also. Examples Stories, News.

**CHAPTER 8 REFERENCES**

1. Geetha M, Manjusha Cy, Unnikrishnan Pz and Harikrishnan Rz, "A Vision Based Dy- namic Gesture Recognition of Indian Sign Language on Kinect based Depth Images, Dept. of Computer Science and Engg. Amrita School of Engineering Amritapuri, Kol- lam, Kerala, India, Vol.58, No.20, pp-2059-2062, June 2012.
2. D.Martinez, "Sign Language Translator using Microsoft Kinect XBOX 360TM, VIBOT 5. Department of Electrical Engineering and Computer Science, Computer Vision Lab, University of Tennessee, Vol.23, No.20, pp-2059-2062, June 2011.

3. Geetha M and Manjusha U C. "A vision based Recogniton of IndianSign language Al- phabets and Numerals using Bspline approximation." INTERNATIONAL JOURNAL ON COMPUTER SCIENCE AND ENGINEERING(LICSE), VOL. 4, NO. 3, MARCH 2012.

4. Matthew Tang. "Recognising hand gestures with Microsoft Kinect," STANFORD UNI- VERSITY, Vol.38. No.20, pp-2059-2062, June 2013.

5. Xiaodong Yang and YingLi Tian, "Eigen Joints Based Action Recognition Using Nave Bayes Nearest Neighbour, THE CITY COLLEGE OF NEW YORK, NEW YORK.

6. J. S. Johson, E. L. Newport (1989), Cognitive Psychology 21, Critical Period Effects in Second Language Learning: The Influence of Maturational State on the Acquisition of English as a Second Language, University of Illinois, USA, 60-99

7. J. Wachs, M. Kolsch, H. Stern, Y. Edan (2011), Communications of the ACM Vol ¨

54, No.2, Vision-Based Hand-Gesture Applications, 62-71

8. F. Chen, C. Fu, C. Huang (2003), Image and Vision Computer 21, Hand gesture

recognition using a real-time tracking method and hidden Markov models, Elsevier,

745-758

9. H. Wang, M. Leu, C. Oz (2005), American Sign Language Recognition Using MultiDimensional Hidden Markov Models, Journal of Information Science and Engineering 22, 1109-1123

10. F. Weichert, D. Bachmann, B. Rudak and D. Fisseler (2013), Sensors 13, Analysis

of the Accuracy and Robustness of the Leap Motion Controller, Technical University

Dortmund, Germany, May 14, 6380-6393

11. M. Khademi, H. M. Hondori, A. McKenzie, L. Dodakian, C. V. Lopes, S. C. Cramer

(2014), CHI ’14 Extended Abstracts on Human Factors in Computing Systems, Freehand interaction with leap motion controller for stroke rehabilitation, University of

California, USA, 1663-1668

12. J.Sutton (2013), ACM SIGGRAPH 2013 Studio Talks, no. 21, Airpainting with

Corel Painter Freestyle and the Leap Motion Controller: A Revolutionary New Way

to Paint

13. N. Xu, W. Wang, X. Qu (2015), Image and Graphics: 8th International Conference, Recognition of In-Air Handwritten Chinese Character Based on Leap Motion

Controller, Tianjin, China, August 13-16 160-168

**ANNEXURE A**

**LABORATORY ASSIGNMENTS ON PROJECT ANALYSIS OF ALGORITHMIC DESIGN**

• To develop the problem under consideration and justify feasibilty using concepts of knowledge canvas and IDEA Matrix.

Refer fig A.1 for IDEA Matrix and Knowledge canvas model. Case studies are given in this book. IDEA Matrix is represented in the following form. Knowledge canvas represents about identification of opportunity for product. Feasibility is represented w.r.t. business perspective.

• Feasibility Knowledge Canvas:

Knowledge canvas is one that depicts the knowledge forces and knowledge flow across the organization. It captures the current knowledge state and knowledge forces in the environment. It tries to build the bigger and broader knowledge scenario for you and your environment. It is simple representation of knowledge oppor- tunities with reference to the environment.

• IDEA Matrix:

– A SYSTEMIC KNOWLEDGE INNOVA TION:

“Innovation is not merely idea + value,but it is the applica- tion of knowledge to change the world and deliver knowledge value”

What separates knowledge organizations from ordinary orga- nizations is the ability to innovate knowedge. While ordinary organizations run for projects,trying to hoard customers,play value, and competitive tactics-knowledge organizations inno- vate knowledge, and expand knowledge horizons. They create new knowledge opportunities. Knowledge organizations fun- damentally approach business in different ways, what we call Systematic Knowledge Innovation.

– A SYSTEMIC KNOWLEDGE INNOVATION-BASED FRAME- WORK:

Organisation reinvention requires disciplined framework,A solid framework for SKI is required; otherwise,following SKI would become more difficult. To build and practice Systemic Knowl- edge Innovation strategy,we have developed a new framework ‘IDEA’ .

This framework is applied to more than a dozen successful organizations and over dozen emerging organization. ‘IDEA’ framework focuses on the highest leverage points of knowl- edge building, and knowledge flow optimization in the Organization. The framework is based on flexibility and simplicity,

while applying knowledge concepts.

• Project problem statement feasibility assessment using NP-Hard, NP-Complete or satisfy ability issues using modern algebra and/or relevant mathematical models.

• NP-hard in computational complexity theory, is a class of prob- lems that are, informally, ”at least as hard as the hardest problems in NP”. More precisely, a problem H is NP-hard when every prob- lem L in NP can be reduced in polynomial time to H. As a conse- quence, finding a polynomial algorithm to solve any NP-hard prob- lem would give polynomial algorithms for all the problems in NP, which is unlikely as many of them are considered hard.

A common mistake is to think that the NP in NP-hard stands for non-polynomial. Although it is widely suspected that there are no polynomial-time algorithms for NP-hard problems, this has never been proven. Moreover, the class NP also contains all problems which can be solved in polynomial time.

• An interesting example is the graph isomorphism problem, the graph theory problem of determining whether a graph isomorphism ex- ists between two graphs. Two graphs are isomorphic if one can be

transformed into the other simply by renaming vertices. Consider these two problems:

• Graph Isomorphism: Is graph G1 isomorphic to graph G2?

• Subgraph Isomorphism: Is graph G1 isomorphic to a subgraph of graph G2?

The Subgraph Isomorphism problem is NP-complete. The graph isomorphism problem is suspected to be neither in P nor NP-complete, though it is in NP. This is an example of a problem that is thought

to be hard, but is not thought to be NP-complete. The easiest way to prove that some new problem is NP-complete is first to prove that it is in NP, and then to reduce some known NP-complete prob- lem to it. Therefore, it is useful to know a variety of NP-complete problems.

• Mathematical Model

S= {I,O,F,Succes,Failure}

Input:{i1,i2,i3,i4} i1=Temperature Sensor i2=Vibration Sensor i3=Knocking Sensor i4=Motion Sensor

Output:{o1,o2,o3,o4} o1= Siron

o2= Alert Notification o3=Glass door lock o4= Shutter lock

Functions: {f1,f2}

f1= Alogorithm for prediction of theft detection

f2= It is the algorithm, when theft occurs notification from Cloud is sent

Success condition:

When theft occurs it gets detected.

Failure condition:

When theft occurs it is not detected

**ANNEXURE B**

**LABORATORY ASSIGNMENTS ON PROJECT QUALITY AND RELIABILITY TESTING OF PROJECT DESIGN**

It should include assignments such as

• Use of divide and conquer strategies to exploit distributed/parallel/concurrent processing of the above to identify object, morphisms, overloading

in functions (if any), and functional relations and any other depen- dencies (as per requirements). It can include Venn diagram, state diagram, function relations, i/o relations; use this to derive objects, morphism, overloading

• In computer science, divide and conquer (DC) is an algorithm de- sign paradigm based on multi-branched recursion. A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type (di- vide), until these become simple enough to be solved directly (con- quer). The solutions to the sub-problems are then combined to give a solution to the original problem.

• This divide and conquer technique is the basis of efficient algo- rithms for all kinds of problems, such as sorting (e.g., quicksort, merge sort), multiplying large numbers (e.g. Karatsuba), syntac- tic analysis (e.g., top-down parsers), and computing the discrete Fourier transform (FFTs).Understanding and designing DC algo- rithms is a complex skill that requires a good understanding of the nature of the underlying problem to be solved.

• As when proving a theorem by induction, it is often necessary to replace the original problem with a more general or complicated problem in order to initialize the recursion, and there is no system- atic method for finding the proper generalization. These DC com- plications are seen when optimizing the calculation of a Fibonacci number with efficient double recursion.

• The correctness of a divide and conquer algorithm is usually proved by mathematical induction, and its computational cost is often de- termined by solving recurrence relations.

OOPS Concepts:

1. Classes

2. Objects

3. Morphism

4. Overloading in Functions.

5. Functional Relations others.

• Classes

In object-oriented programming, a class is an extensible program- code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member func- tions or methods). In many languages, the class name is used as the name for the class (the template itself), the name for the default constructor of the class (a subroutine that creates objects), and as the type of objects generated by instantiating the class; these dis- tinct concepts are easily conflated.

• When an object is created by a constructor of the class, the resulting object is called an instance of the class, and the member variables specific to the object are called instance variables, to contrast with the class variables shared across the class.

• In some languages, classes are only a compile-time feature (new classes cannot be declared at runtime), while in other languages classes are first-class citizens, and are generally themselves objects (typically of type Class or similar). In these languages, a class that creates classes is called a metaclass.

• Object

In computer science, an object is a location in memory having a value and possibly referenced by an identifier. An object can be a variable, a data structure, or a function. In the class-based object- oriented programming paradigm, ”object” refers to a particular in- stance of a class where the object can be a combination of variables, functions, and data structures. In relational database management, an object can be a table or column, or an association between data and a database entity (such as relating a person’s age to a specific person).

• Morphism

In many fields of mathematics, Morphism refers to a structure- preserving map from one mathematical structure to another. The notion of Morphism recurs in much of contemporary mathemat- ics. In set theory, morphisms are functions; in linear algebra, linear transformations; in group theory, group homomorphisms; in topol- ogy, continuous functions, and so on. In category theory, Mor-phism is a broadly similar idea, but somewhat more abstract: the mathematical objects involved need not be sets, and the relation- ship between them may be something more general than a map.

• The study of morphisms and of the structures (called objects) over which they are defined is central to category theory. Much of the terminology of morphisms, as well as the intuition underlying them, come from concrete categories, where the objects are simply sets with some additional structure, and morphisms are structure- preserving functions. In category theory, morphisms are sometimes also called arrows.

• A category C consists of two classes, one of objects and the other of morphisms.There are two objects that are associated to every mor- phisms, the source and the target. For many common categories, objects are sets (usually with more structure) and morphisms are functions from an object to another object. Therefore the source and the target of morphisms are often called respectively domain and codomain. A morphisms f with source X and target Y is writ- ten f : X Y. Thus Morphism is represented by an arrow from its source to its target.

• Morphisms are equipped with a partial binary operation, called composition. The composition of two Morphism f and g is defined if and only if the target of g is the source of f, and is denoted fg. The source of fg is the source of g, and the target of fg is the target of f. The composition satisfies two axioms:

Identity: for every object X, there exists a morphism idX : X X called the identity

Morphism on X, such that for every morphism f : A B we have idB f = f = f idA.

Associativity: h (g f) = (h g) f whenever the operations are de- fined, that is when the target of f is the source of g, and the target of g is the source of h. For a concrete category (that is the ob- jects are sets with additional structure, and of the morphisms as structure-preserving functions), the identity morphisms is just the identity function, and composition is just the ordinary composition of functions. Associativity then follows, because the composition of functions is associative.

• The composition of morphisms is often represented by a commu- tative diagram. For example,

• The collection of all morphisms from X to Y is denoted homC(X, Y) or simply hom(X, Y) and called the hom-set between X and Y. Some authors write MorC(X, Y), Mor(X, Y) or C(X, Y). Note that the term hom-set is a bit of a misnomer as the collection of mor- phisms is not required to be a set. A category where home (X, Y) is a set for all objects X and Y is called locally small.

• Use of above to draw functional dependency graphs and relevant Software modeling methods, techniques including UML diagrams or other necessities using appropriate tools.

• Testing of project problem statement using generated test data (us- ing mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML dia- gram’s reliability. Write also test cases [Black box testing] for each identified function. You can use Mathematica or equivalent open -source tool for generating test data.

**ANNEXURE C PROJECT PLANNER**

Using planner or alike project management tool.

**ANNEXURE D**

**REVIEWERS COMMENTS OF PAPER SUBMITTED**

1. Paper Title:

Sign Language Recognition Using Multiple Kernel Learning: A Case Study of Pakistan Sign Language.

2. Name of the Conference/Journal where paper submitted: International Journal for Research in Applied science and Engineering Technology (IJRASET)

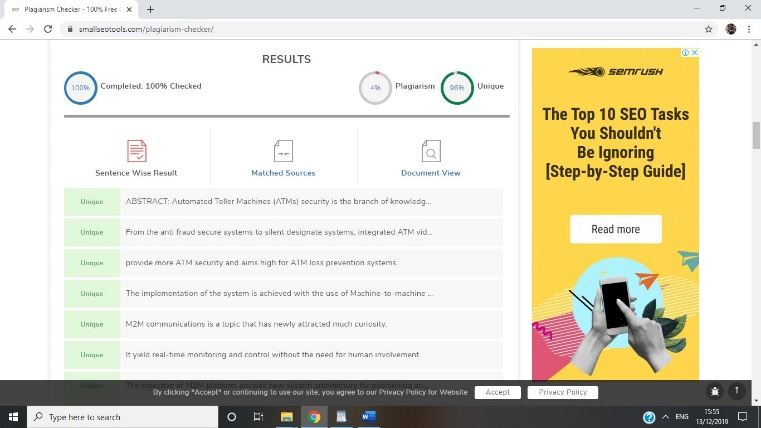
3. Paper accepted/rejected: Accepted

4. Review comments by reviewer:

5. Corrective actions if any: NO Correction

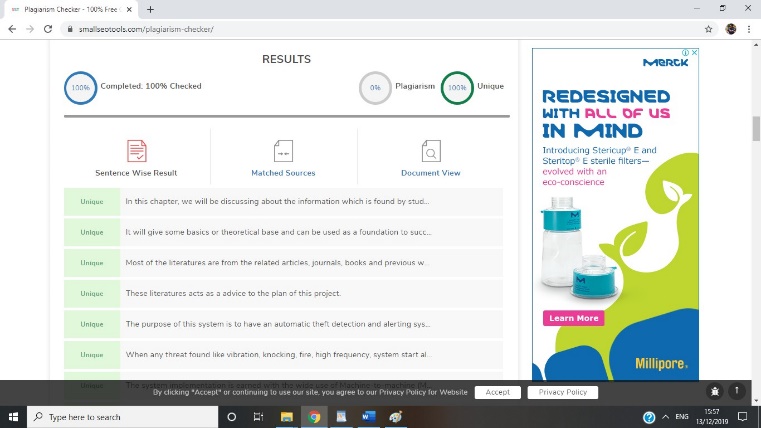
**ANNEXURE E PLAGIARISM REPORT**

Plagiarism report



Format Sem-I/plagarism check 1.png

**Figure E.1: plagarism check 1**

****

**Figure E.2: plagarism check 2**