VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

A Project work Phase-I Report

on

VIRTUAL CURSOR USING HAND GESTURES AND VOICE RECOGNITION

Submitted in partial fulfillment of the requirements for the final year degree in
Bachelor of Engineering in Computer Science and Engineering
of Visvesvaraya Technological University, Belagavi

Submitted by

NITHEESH SHETTY	**1RN19CS091**
NITISH K	**1RN19CS092**
RAKSHITH BK	**1RN19CS108**
RAJATHA BANGERA	**1RN20CS411**

Under the Guidance of:

Internal Guide:
Mrs. Yashasvi B N
Assistant Professor
Dept. of CSE

Department of Computer Science and Engineering

RNS Institute of Technology

(Accredited by NBA upto 30-06-2025)

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

2022-2023

RNS INSTITUTE OF TECHNOLOGY

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(NBA ACCREDITED FOR ACADEMIC YEARS 2018-19, 2019-20, 2020-21, 2021-22, 22-25)

CERTIFICATE

Certified that the Project Phase 1 Report work entitled **Virtual Cursor using Hand Gestures and Voice Recognition** has been successfully carried out at **RNSIT** by **Nitheesh Shetty** bearing USN **1RN19CS091**, **Nitish K** bearing USN **1RN19CS092**, **Rakshith BK** bearing USN **1RN19CS108** and **Rajatha Bangera** bearing USN **1RN20CS411** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements of final year degree in **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year **2022-2023**. The project work phase-I has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Signature of the Guide	Signature of the HoD	Signature of the Principal
Mrs. Yashasvi B N	**Dr. Kiran P**	**Dr. M K Venkatesha**
Assistant Professor	Professor	Principal
Dept. of CSE	Dept. of CSE	RNSIT

Name of the Examiners	**Signature with Date**
1.	
2.	

Acknowledgement

At the very onset, I would like to place on record my gratitude to all those people who have helped me in making this work a reality. Our Institution has played a paramount role in guiding in the right direction. I would like to profoundly thank **Sri. Satish R Shetty**, Managing Director, RNS Group of Companies, Bengaluru for providing such a healthy environment for the successful completion of this Project work.

I would like to thank our beloved Principal, **Dr. M K Venkatesha**, for providing the necessary facilities to carry out this work. I am extremely grateful to **Dr. Kiran P**, Professor and Head, Department of Computer Science and Engineering for having accepted to patronize me in the right direction with all his wisdom.

I would like to express my sincere thanks to our Coordinator and guide, **Mrs. Yashasvi B N** , Assistant Professor, for her constant encouragement that motivated me for the successful completion of this work. Last but not the least, I am thankful to all the teaching and non-teaching staff members of the Computer Science and Engineering Department for their encouragement and support throughout this work.

Nitheesh Shetty 1RN19CS091

Nitish K 1RN19CS092

Rakshith BK 1RN19CS108

Rajatha Bangera 1RN20CS411

Abstract

Gesture-controlled laptops and computers have recently gained a lot of traction. Leap motion is the name for this technique. This paper proposes a way to control the position of the cursor with the bare hands without using any electronic device. The mouse is one of the wonderful inventions of Human-Computer Interaction (HCI) technology. Currently, wireless mouse or a Bluetooth mouse still uses devices and is not free of devices completely since it uses a battery for power and a dongle to connect it to the PC. The Virtual Mouse provides an infrastructure between the user and the system using only a camera. It allows users to interface with machines without the use of mechanical or physical devices, and even control mouse functionalities.

In the proposed AI virtual mouse system, this limitation can be overcome by employing webcam or a built-in camera for capturing of hand gestures and hand tip detection using computer vision. The algorithm used in the system makes use of the machine learning algorithm. Hand gestures are the most natural and effortless manner of communicating. The proposed system will only require a webcam as an input device.

The software's that will be required to implement the proposed system are OpenCV and python. The output of the camera will be displayed on the system's screen so that it can be further calibrated by the user. Based on the hand gestures, the computer can be controlled virtually and can perform left click, right click, scrolling functions, and computer cursor function without the use of the physical mouse. The python dependencies that will be used for implementing this system are NumPy, math, wx and mouse. For voice recognition we make use of Speech Recognition, PyAudio, pyttsx3, pywhatkit.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 PURPOSE, AIM AND OBJECTIVES:

The aim of this project is to develop a Virtual Mouse application that targets a few aspects of significant development. For starters, this project aims to eliminate the needs of having a physical mouse while able to interact with the computer system through webcam by using various image processing techniques. Other than that, this project aims to develop a Virtual Mouse application that can be operational on all kind of surfaces and environment. The goal is to manage computers and other devices with gestures rather than pointing and clicking a mouse or touching a display directly. Backers believe that the approach can make it not only easier to carry out many existing chores but also take on trickier tasks such as creating 3-D models, browsing medical imagery during surgery without touching anything. Also, to reduce the cost of having hardware.

1.2 BACKGROUND OF PROJECT:

A mouse, in computing terms is a pointing device that detects twodimensional movements relative to a surface. This movement is converted into the movement of a pointer on a display that allows to control the Graphical User Interface (GUI) on a computer platform. There are a lot of different types of mouses that have already existed in the modern day's technology, there's the mechanical mouse that determines the movements by a hard rubber ball that rolls around as the mouse is moved. Years later, the optical mouse was introduced that replace the hard rubber ball to a LED sensor to detects

table top movement and then sends off the information to the computer for processing.

However, no matter how accurate can it be, there are still limitations exist within the mouse itself in both physical and technical terms. For example, a computer mouse is a consumable hardware device as it requires replacement in the long run, either the mouse buttons were degraded that causes inappropriate clicks, or the whole mouse was no longer detected by the computer itself. Despite the limitations, the computer technology still continues to grow, so does the importance of the human computer interactions.

However, even though the touch screen technology for the desktop system is already exist, the price can be very steep. Therefore, a virtual human computer interaction device that replaces the physical mouse or keyboard by using a webcam or any other image capturing devices can be an alternative way for the touch screen. This device which is the webcam will be constantly utilized by a software that monitors the gestures given by the user in order to process it and translate to motion of a pointes, as similar to a physical mouse.

1.3 SCOPE OF PROJECT:

Virtual Mouse that will soon to be introduced to replace the physical computer mouse to promote convenience while still able to accurately interact and control the computer system. To do that, the software requires to be fast enough to capture and process every image, in order to successfully track the user's gesture. Therefore, this project will develop a software application with the aid of the latest software coding technique and the open-source computer vision library also known as the OpenCV. The scope of the project is as below:

- Real time application
- User friendly application
- Removes the requirement of having a physical mouse

Chapter 2

LITERATURE SURVEY

- Title: VirtualMouse47-GSJ2500.pdf

  Author: Vijay Kumar Sharma, Vimal Kumar, Md. Iqbal, Sachin Tawara, Vishal Jayaswal

  This paper proposes a way to control the position of the cursor with the bare hands without using any electronic device. While the operations like clicking and dragging of objects will be performed with different hand gestures. The proposed system will only require a webcam as an input device. The software's that will be required to implement the proposed system are OpenCV and python. The output of the camera will be displayed on the system's screen so that it can be further calibrated by the user. The python dependencies that will be used for implementing this system are NumPy, math, wx and mouse.[DOI: 10.1109/ICOEI.2019.8862764]

  Advantages:

  - The proposed system is a practically viable option in the real world.
  - Has made use of hand gestures for controlling the mouse

  Disadvantages

  - Absence of voice recognition system to turn on or turn off the virtual mouse

- Title: AJSAT-Vol.5-No.2-July-Dece-2016-pp.23-30.pdf

  Author: Pala Mahesh Kumar

  Abstract - In an effort to provide a more efficient representation of the speech signal, the

application of the wavelet analysis is considered. This research presents an effective and robust method for extracting features for speech processing. Here, we proposed a new human voice recognition system using the combination of decimated wavelet (DW) and Relative Spectra Algorithm with Linear Predictive coding. First, we will apply the proposed techniques to the training speech signals and then form a train feature vector which contains the low level features extracted, wavelet and linear predictive coefficients. Afterwards, the same process will be applied to the testing speech signals and will form a test feature vector. Now, we will compare the two feature vectors by calculating the Euclidean distance between the vectors to identify the speech and speaker. If the distance between two vectors is near to zero then the tested speech/speaker will be matched with the trained speech/speaker. Simulation results have been compared with LPC scheme, and shown that the proposed scheme has performed superior to the existing technique by using the fifty preloaded voice signals from six individuals, the verification tests have been carried and an accuracy rate of approximately 90 % has been achieved. [DOI: https://doi.org/10.51983/ajsat-2016.5.2.931]

Advantages:

- By using the fifty preloaded voice signals from six individuals, the verification tests have been carried and an accuracy rate of approximately 90 % has been achieved by proposed algorithm where the LPC has achieved only 66.66%.

Disadvantages:

- Accuracy can be further increased by increasing the number of preloaded voices to approx 95%

- Title: Survey on Vision based Hand Gesture Recognition.pdf

  Author: Gaurab Baruah; Anjan Kr. Talukdar; Kandarpa Kumar Sarma

  Abstract: The use of hand gesture for Human Computer Interface (HCI) is very common nowadays with the help of smart devices such as smart TV, smart phone, etc. and in recent years vision-based gesture recognition system has received a great attention. The task of recognition in real time and with high accuracy is becoming the main priority. In gesture recognition system one commonly encountered problem is detecting the gesture when made from side

angles. In this work, a solution to that problem is derived. In this paper, a hand gesture recognition system for multiple viewing angles using a single camera based on combination of Affine Transform and Discrete Fourier Transform (DFT) is presented. Experimental results show that the system can efficiently detect gestures made from multiple angles with an average recognition rate of 95.28% and 90.30% for gestures made at ±30° and ±45° respectively. [DOI: 10.1109/HYDCON48903.2020.9242677]

Advantages:

- Simple to use and cost effective.
- System performance is good enough and can be used.

Disadvantage:

- Absence of voice recognition system to turn on or turn off the virtual mouse

- Title: Vision-Based Hand Gesture Recognition Using Eccentric Approach[9]

  Author: Vishal Bhame R.Sreemathy Hrushikesh Dhumal

  Abstract: There has been growing interest in the development of new approaches and technologies for bridging the human-computer barrier. Hand gesture recognition is considered as an interaction technique having potential to communicate with machines. Human computer interaction (HCI) was never an easy task and lots of approaches are available to build such systems. Hand gesture recognition (HGR) using wearable data glove provides a solution to build a HCI system, but lags in terms of its computational time and poor interface. Pattern matching is one more solution which uses vision based techniques and provides strong interface to build HCI systems. But again, it requires complex algorithms which takes lots of computational time and hence limits its use in real time HCI applications. In this paper, we presented an eccentric approach for hand gesture recognition which is simple, fast and user independent and can be used to develop real time HCI applications. Based on proposed algorithm we built a system for Indian Sign Language recognition which converts Indian Sign numbers into text. The algorithm first captures the image of single handed gesture of speech/hearing impaired person using simple webcam and then using our proposed algorithm it classifies the gesture into its appropriate class. It uses

simple logical conditions for gesture classification which make its use in real time HCI applications.[DOI:10.1109/ICACCI.2014.6968545]

Advantages:

- The system is completely autonomous and easy to use as the users

- Title: Virtual Mouse Using Hand Gesture

  Author: Neha Ramakrishnan, Mrs. Jena Catherine Bel

  Abstract: Image Processing is a strategy to extract certain information from an image, so as a result it gets an upgraded image or some data can be extracted from the image. It is a kind of sign handling where input is an image and yield might be an image or qualities/highlights related with that image. These days, image processing is one among the quickly developing advancements. In this project hand gestures are used as a replacement of mouse. We will be using digital image processing in this project to retrieve necessary data from the image captured through the user's web camera and implement it to the mouse interface of the computer according to predefined functions. User can be calibrate the color ranges of the three finger caps individually. It can be calibrated separately for all the three colors at the starting of the program. The user is provided with a default setting as well. It detects the centroid of the three color and decides what mouse action has to be done on their relative positions. Based on the detection, corresponding actions will be choosed and the mouse functions will be performed using the Python specific library PyAutoGUI. We can perform actions like left click, right click, scroll up, scroll down and drag. A camera catches your video and relying upon your hand motions, you can move the cursor and perform left-click, right-click, drag, select and look here and there. The predefined motions utilize just three fingers set apart by various hues.[DOI: 10.1109/ICTAI53825.2021.9673251].

Chapter 3

PROBLEM STATEMENT

To track fingertips as a movable object, and to utilize it for mouse functions. A virtual mouse instead of a physical mouse works based on webcam captured frames tracking colored fingertips and also to implement a simple voice recognition system.This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process.The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

3.1 EXISTING SYSTEM

The existing system consists of the generic mouse and track pad system of monitor controlling and the no availability of a hand gesture system. The remote accessing of monitor screen using the hand gesture is unavailable. Even-though there exists a virtual mouse which is largely trying to implement the scope it is simply restricted in the field of virtual mouse.

There are many software or application are present on the internet which claims that they can control the mouse without touching it but on the ground check reality they do not the same as they say instead of they use controlling of mouse virtually by using coloured tape on the finger or using colour monitoring techniques that inappropriate for anyone to manage those kinds of stuff.

3.2 LIMITATIONS OF EXISTING SYSTEM

Even-though there are a number of systems which are used for hand recognition, the system they made used is the static hand recognition which is simply recognition of the shape made by hand and by defining an action for each shape made, which is limited to a number of defined actions and a large amount of confusion.The remote accessing of monitor screen using the hand gesture is unavailable. Always a need of operating a physical device. Not convenient and not a cost-effective solution.

3.3 PROPOSED SYSTEM

Using our project we could make use of the laptop's web-cam and by recognizing the hand gesture we could control mouse and perform basic operations like mouse pointer controlling, select and deselect using left click, and some additional features like scrolling, drag and drop, multiple item selection, volume control etc.

The project done is a "Zero Cost" hand recognition system for laptops, which uses simple algorithms to determine the hand, hand movements and by assigning an action for each movement. The system we are implementing which is been written in python code be much more responsive and is easily implemented since python is a simple language and is platform independent with flexibility and is also portable.

3.4 ADVANTAGES OF PROPOSED SYSTEM

- Any new product should either make human life more comfortable, more productive or more fun.
- Provides greater flexibility than the existing system.
- Easy to modify and adapt.
- Less prone to physical damage due to absence of a fixed physical device.
- Avoid the mouse-related wrist damage like CTS RSI.

Chapter 4

REQUIREMENT ANALYSIS AND FEASIBILITY STUDY

4.1 FUNCTIONAL REQUIREMENT

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed; the general approach taken by the organization when writing requirements. The functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on. Functional requirements are as follows:

- Capture hand and draw line of interest
- Process image frame by frame
- Perform a gesture to make certain action
- Detect gesture.
- Perform corresponding event on screen.

4.2 NON FUNCTIONAL REQUIREMENT

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define

constraints on the system such as capabilities of I/O devices and the data representations used in system interfaces. The non-functional requirements are as follows:

- The program must be self-contained so that it can easily be moved from one Computer to another. It is assumed that network connection will be available on the computer on which the program resides.
- Capacity, scalability, and availability.
- The system shall always achieve 100% availability. The system shall be scalable to support additional clients and volunteers.
- Maintainability. The system should be optimized for supportability, or ease of maintenance as far as possible. This may be achieved through the use documentation of coding standards,naming conventions, class libraries and abstraction.
- Randomness, verifiability, and load balancing. The system should be optimized for supportability, or ease of maintenance as far as possible.

4.3 HARDWARE REQUIREMENT

- Processor : Intel or AMD.
- Ram : 2 GB or above.
- Hard disk : 40GB or above.
- Peripheral device : webcam with at least 30 frames per second,640 x 480 resolution

4.4 SOFTWARE REQUIREMENT

- Technology/Language : Python
- IDE : PyCharm, VScode, Sublime
- Operating System : Windows, Linux, Mac.
- Python version : 3.6x and later versions Library : Open CV

4.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

4.5.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.5.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Chapter 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system.It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough as shown in Figure 5.1.

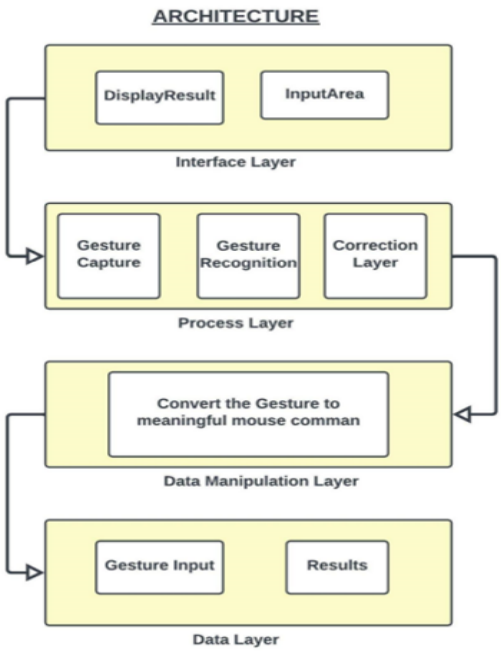

Figure 5.1: Architecture of System

5.2 DATA FLOW DIAGRAMS (DFD)/ FLOW CHARTS

Level: 0 describes the overall process of the project. We are using captures video frames as input. System will use the opencv to detect hand and the fingers and capture the movements in frames as shown in Figure 5.2.

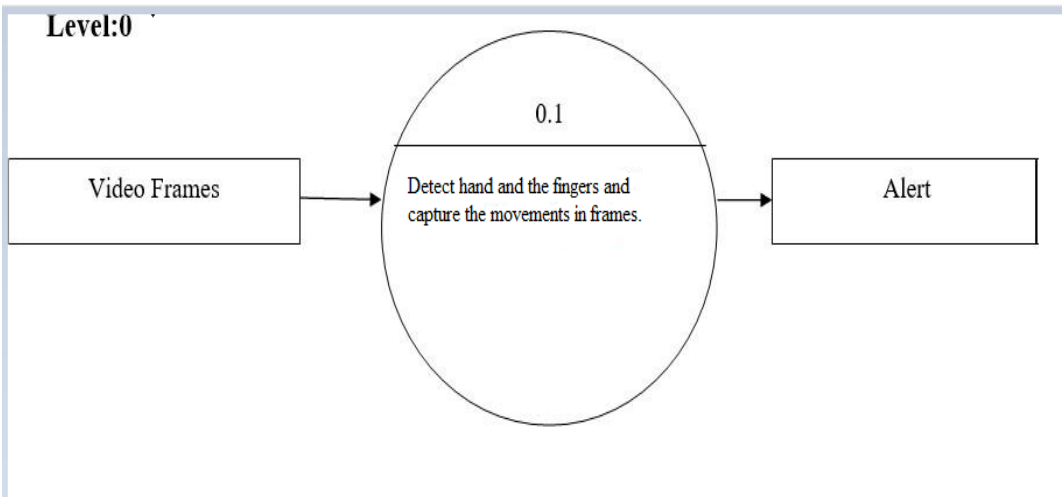

Figure 5.2: Data Flow Diagram level 0.

Level: 1 describes the initial process of the project. We are using captures video frames as input. System will preprocess and extract the face and eye features as shown in Figure 5.3.

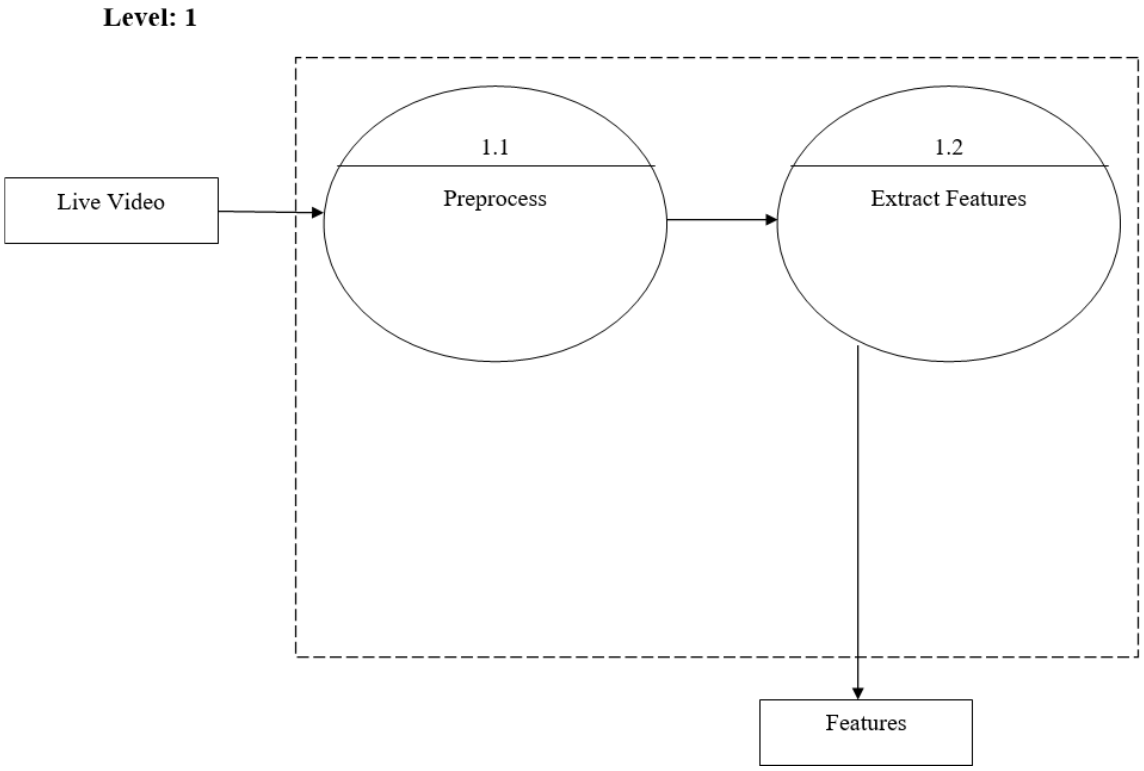

Figure 5.3: Data Flow Diagram level 1.

Level: 2 describes the final process of the project. We are using captures video frames as input. System detect the drowsiness and generate alarm to the driver using CNN as shown in Figure 5.4.

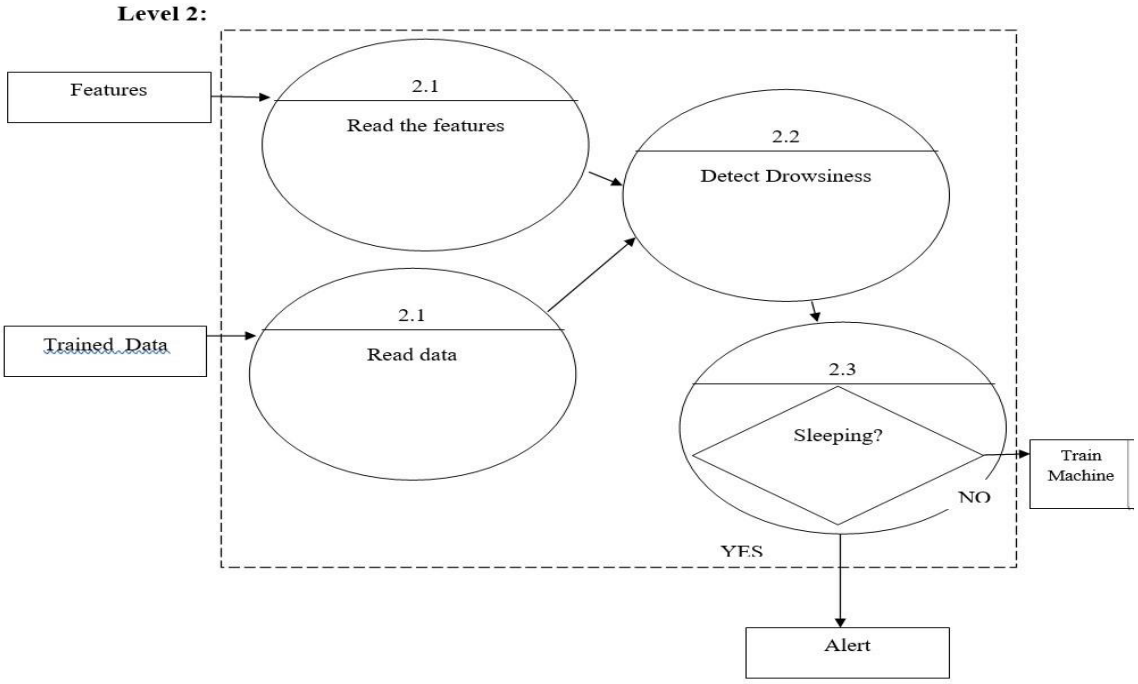

Figure 5.4: Data Flow Diagram level 2.

Another feature which we will try to implement in our virtual cursor is the use of voice recognition system the flowchart for which is as shown in figure 5.5

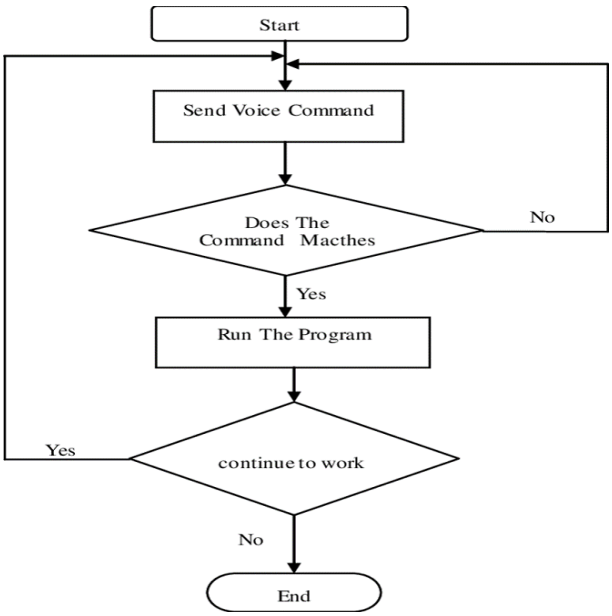

Figure 5.5: Voice Recognition Flow Diagram

5.3 SEQUENCE DIAGRAM

Sequence diagram are used to represent the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram as shown in figure 5.6.

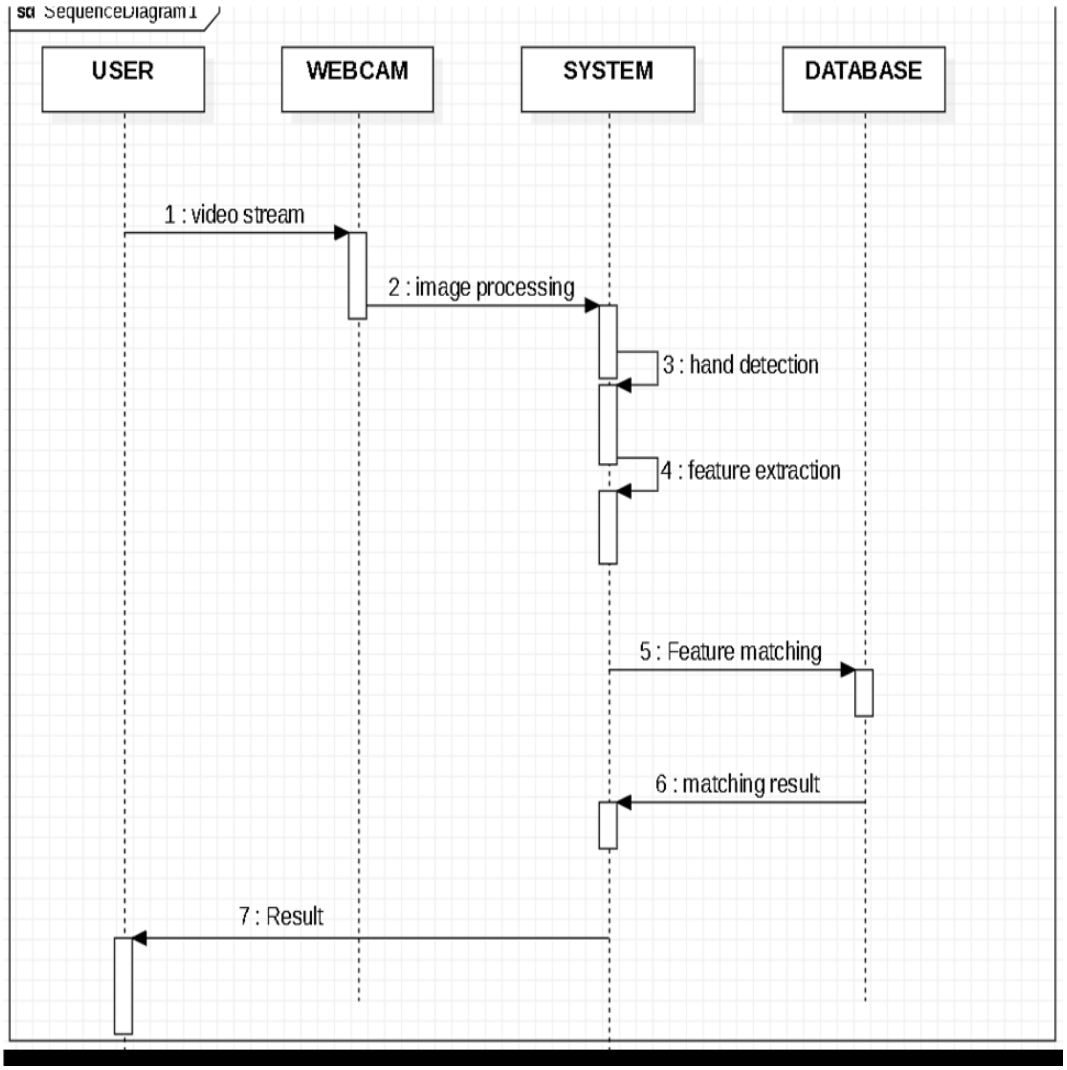

Figure 5.6: Sequence Diagram

Chapter 6

IMPLEMENTATION

6.1 PSEUDO CODE / ALGORITHM

6.1.1 Procedure for Image Capturing

Input: Web Camera Turned on

- Start
- Capture Hand and draw line of Interest
- Process Image frame by frame
- Perform a gesture to make certain action
- Detect the gesture
- Perform corresponding event on screen
- Stop

6.1.2 Procedure for Voice recognition

Input: Send Voice Commands

- Start
- Send Voice Command

- Does the command matches
- If yes
- Run the Program
- If no repeat Step 2
- Run the Program
- Continue to work
- If yes
- Repeat Step 2
- If no
- End

6.2 CONFIGURATION

- Install anaconda
- Install python interpreter
- Install Pycharm
- Make sure that all the above softwares installed are the latest versions
- Link the python interpreter to Pycharm
- Install the required libraries such as MediaPipe, pyautogui etc.
- Run the file Gesture_Controller.py
- The required output will be as shown on the screen

6.3 CODE

```
import cv2
import mediapipe as mp
import pyautogui
import math
from enum import IntEnum
from ctypes import cast, POINTER
from comtypes import CLSCTX_ALL
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume
from google.protobuf.json_format import MessageToDict
import screen_brightness_control as sbcontrol
pyautogui.FAILSAFE = False
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands

class Gest(IntEnum):
    FIST = 0
    PINKY = 1
    RING = 2
    MID = 4
    LAST3 = 7
    INDEX = 8
    FIRST2 = 12
    LAST4 = 15
    THUMB = 16
    PALM = 31
    # Extra Mappings
    V_GEST = 33
    TWO_FINGER_CLOSED = 34
```

```
    PINCH_MAJOR = 35
    PINCH_MINOR = 36

class HLabel(IntEnum):
    MINOR = 0
    MAJOR = 1

class HandRecog:
    def __init__(self, hand_label):
        self.finger = 0
        self.ori_gesture = Gest.PALM
        self.prev_gesture = Gest.PALM
        self.frame_count = 0
        self.hand_result = None
        self.hand_label = hand_label

    def update_hand_result(self, hand_result):
        self.hand_result = hand_result

    def get_signed_dist(self, point):
        sign = -1
        if self.hand_result.landmark[point[0]].
        y < self.hand_result.landmark[point[1]].y:
            sign = 1
        dist = (self.hand_result.landmark[point[0]].x
        - self.hand_result.landmark
        [point[1]].x) ** 2
        dist += (self.hand_result.landmark[point[0]]
        .y - self.hand_result.landmark[point[1]].y) ** 2
```

```
        dist = math.sqrt(dist)
        return dist * sign

    def get_dist(self, point):
        dist = (self.hand_result.landmark[point[0]].x
        - self.hand_result.landmark[point[1]].x) ** 2
        dist += (self.hand_result.landmark[point[0]].y
        - self.hand_result.landmark[point[1]].y) ** 2
        dist = math.sqrt(dist)
        return dist

    def get_dz(self, point):

        return abs(self.hand_result.landmark[point[0]].z
        - self.hand_result.landmark[point[1]].z)

    def set_finger_state(self):
        if self.hand_result == None:
            return
        points = [[8, 5, 0], [12, 9, 0], [16, 13, 0], [20, 17, 0]]
        self.finger = 0
        self.finger = self.finger | 0  # thumb
        for idx, point in enumerate(points):
            dist = self.get_signed_dist(point[:2])
            dist2 = self.get_signed_dist(point[1:])
            try:
                ratio = round(dist / dist2, 1)
            except:
                ratio = round(dist / 0.01, 1)
```

```
            self.finger = self.finger << 1
            if ratio > 0.5:
                self.finger = self.finger | 1


class Controller:
    tx_old = 0
    ty_old = 0
    trial = True
    flag = False
    grabflag = False
    def scrollHorizontal(self):
        pyautogui.keyDown('shift')
        pyautogui.keyDown('ctrl')
        pyautogui.scroll(-120 if Controller.pinchlv > 0.0 else 120)
        pyautogui.keyUp('ctrl')
        pyautogui.keyUp('shift')

    def get_position(hand_result):
        point = 9
        position = [hand_result.
        landmark[point].x, hand_result.landmark[point].y]
        sx, sy = pyautogui.size()
        x_old, y_old = pyautogui.position()
        x = int(position[0] * sx)
        y = int(position[1] * sy)
        if Controller.prev_hand is None:
            Controller.prev_hand = x, y
        delta_x = x - Controller.prev_hand[0]
        delta_y = y - Controller.prev_hand[1]
```

```
        distsq = delta_x ** 2 + delta_y ** 2
        ratio = 1
        Controller.prev_hand = [x, y]
        if distsq <= 25:
            ratio = 0
        elif distsq <= 900:
            ratio = 0.07 * (distsq ** (1 / 2))
        else:
            ratio = 2.1
        x, y = x_old + delta_x * ratio, y_old + delta_y * ratio
        return (x, y)


    def handle_controls(gesture, hand_result):
        x, y = None, None
        if gesture != Gest.PALM:
            x, y = Controller.get_position(hand_result)
        if gesture != Gest.FIST and Controller.grabflag:
            Controller.grabflag = False
            pyautogui.mouseUp(button="left")
        if gesture != Gest.PINCH_MAJOR and Controller.pinchmajorflag:
            Controller.pinchmajorflag = False
        if gesture != Gest.PINCH_MINOR and Controller.pinchminorflag:
            Controller.pinchminorflag = False
        if gesture == Gest.V_GEST:
            Controller.flag = True
            pyautogui.moveTo(x, y, duration=0.1)
        elif gesture == Gest.FIST:
            if not Controller.grabflag:
                Controller.grabflag = True
```

```
                pyautogui.mouseDown(button="left")
            pyautogui.moveTo(x, y, duration=0.1)
        elif gesture == Gest.MID and Controller.flag:
            pyautogui.click()
            Controller.flag = False
        elif gesture == Gest.INDEX and Controller.flag:
            pyautogui.click(button='right')
            Controller.flag = False


        elif gesture == Gest.TWO_FINGER_CLOSED and Controller.flag:
            pyautogui.doubleClick()
            Controller.flag = False
        elif gesture == Gest.PINCH_MINOR:
            if Controller.pinchminorflag == False:
                Controller.pinch_control_init(hand_result)
                Controller.pinchminorflag = True
            Controller.pinch_control(hand_result
            , Controller.scrollHorizontal, Controller.scrollVertical)
        elif gesture == Gest.PINCH_MAJOR:
            if Controller.pinchmajorflag == False:
                Controller.pinch_control_init(hand_result)
                Controller.pinchmajorflag = True
            Controller.pinch_control(hand_result, Controller.
            changesystembrightness
            , Controller.changesystemvolume)


class GestureController:
    gc_mode = 0
    cap = None
```

```
    CAM_HEIGHT = None
    CAM_WIDTH = None
    hr_major = None  # Right Hand by default
    hr_minor = None  # Left hand by default
    dom_hand = True

    def __init__(self):
        """Initilaizes attributes."""
        GestureController.gc_mode = 1
        GestureController.cap = cv2.VideoCapture(0)
        GestureController.CAM_HEIGHT = GestureController.cap
        .get(cv2.CAP_PROP_FRAME_HEIGHT)
        GestureController.CAM_WIDTH = GestureController.cap
        .get(cv2.CAP_PROP_FRAME_WIDTH)

    def classify_hands(results):
        left, right = None, None
        try:
            handedness_dict = MessageToDict(results
            .multi_handedness[0])
            if handedness_dict['classification'][0]['label'] == 'Right':
                right = results.multi_hand_landmarks[0]
            else:
                left = results.multi_hand_landmarks[0]
        except:
            pass
        try:
            handedness_dict = MessageToDict(results.multi_handedness[1])
            if handedness_dict['classification'][0]['label'] == 'Right':
```

```
                right = results.multi_hand_landmarks[1]
            else:
                left = results.multi_hand_landmarks[1]
        except:
            pass
        if GestureController.dom_hand == True:
            GestureController.hr_major = right
            GestureController.hr_minor = left
        else:
            GestureController.hr_major = left
            GestureController.hr_minor = right


    def start(self):
        handmajor = HandRecog(HLabel.MAJOR)
        handminor = HandRecog(HLabel.MINOR)
        with mp_hands.Hands(max_num_hands=2, min_detection_confidence=0.5,
            while GestureController.cap
            .isOpened() and GestureController.gc_mode:
                success, image = GestureController.cap.read()
                if not success:
                    print("Ignoring␣empty␣camera␣frame.")
                    continue
                image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)
                image.flags.writeable = False
                results = hands.process(image)
                image.flags.writeable = True
                image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
                if results.multi_hand_landmarks:
                    GestureController.classify_hands(results)
```

```
                    handmajor.update_hand_result(GestureController
                    .hr_major)
                    handminor.update_hand_result(GestureController
                    .hr_minor)
                    handmajor.set_finger_state()
                    handminor.set_finger_state()
                    gest_name = handminor.get_gesture()
                    if gest_name == Gest.PINCH_MINOR:
                        Controller.handle_controls
                        (gest_name, handminor.hand_result)
                    else:
                        gest_name = handmajor.get_gesture()
                        Controller.handle_
                        controls(gest_name, handmajor.hand_result)
                    for hand_landmarks in results.multi_hand_landmarks:
                        mp_drawing.draw_landmarks
                        (image, hand_landmarks, mp_hands.HAND_CONNECTIONS)
                else:
                    Controller.prev_hand = None
                cv2.imshow('Gesture Controller', image)
                if cv2.waitKey(5) & 0xFF == 13:
                    break
        GestureController.cap.release()
        cv2.destroyAllWindows()

gc1 = GestureController()
gc1.start()
```

Chapter 7

RESULT ANALYSIS

7.1 TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

7.1.1 TESTING OBJECTIVES:

- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation
- To make sure that during the operation, incorrect input, processing and output will be detected
- To see that when correct inputs are fed to the system the outputs are correct
- To verify that the controls incorporated in the same system as intended
- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error. Note that the result of the system testing will prove that the system is working

correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

7.1.2 TESTING METHODOLOGIES:

- White box testing.
- Black box testing.
- Unit testing.
- Integration testing.
- User acceptance testing.
- Output testing.
- Validation testing.
- System testing.

• White box testing: White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

• Black box testing: Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

• Unit testing: Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete

coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

• Integration testing: Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

• User acceptance testing: User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

• Output testing: After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

7.1.3 TESTING STRATEGY :

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module. In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.

7.2 RESULTS

Neutral Gesture is used to halt/stop execution of current gesture. Cursor is assigned to the midpoint of index and middle fingertips. This gesture moves the cursor to the desired location. Speed of the cursor movement is proportional to the speed of hand.Dynamic Gestures for horizontal and vertical scroll. The speed of scroll is proportional to the distance moved by pinch gesture from start point. Vertical and Horizontal scrolls are controlled by vertical and horizontal pinch movements respectively. Gesture for drag and drop functionality. Can be used to move/tranfer files from one directory to other.The major extension to this work can be done to make system able to work at much complex background and compatible with different light conditions. It can be made as an effective user interface and which can include all mouse functionalities.

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

Gesture recognition gives the best interaction between human and machine. Gesture recognition is also important for developing alternative human computer interaction modalities. It enables human to interface with machine in a more natural way. Gesture recognition can be used for many applications like sign language recognition for deaf and dumb people, robot control etc. This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, and biomedical instrumentation. This technology can be used to help patients who don't have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person's motions are tracked and interpreted as commands.

8.2 FUTURE SCOPE:

The major extension to this work can be done to make system able to work at much complex background and compatible with different light conditions. It can be made as an effective user interface and which can include all mouse functionalities. And also, it would be ideal to research into advanced mathematical materials for image processing and investigate on different hardware solutions that would result in more accurate hand detections.

References

[1] Jan Erik Solem , *"Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images Book "*

[2] `https://pypi.org/project/opencv-python/`

[3] `https://google.github.io/mediapipe/getting_started/python.html`

[4] `https://ieeexplore.ieee.org/document/9673251`

[5] Jan Erik Solem , *"J. S. Sonkusare, N. B. Chopade, R. Sor, and S. L. Tade, "A Review on Hand Gesture Recognition System," 2015 Int. Conf Comput. Commun. Control Autom., pp. 790-794,2015. "*

[6] Jan Erik Solem , *"M. Panwar and P. Singh Mehra, "Hand gesture recognition for human computer interaction," IEEE Int. Conf Image In! Process., 2011 no. Iciip, pp. 1-7."*

[7] Jan Erik Solem , *"Q. Y. Zhang, F. Chen and X. W. Liu, "Hand Gesture Detection and Segmentation Based on Difference Background Image with Complex Background," Proceedings of the 2008 International Conference on Embedded Software and Systems, Sichuan, 29-31 July 2008, pp. 338- 343. "*

[8] Jan Erik Solem , *"Abhik Banerjee, Abhirup Ghosh, Koustuvmoni Bharadwaj, "Mouse Control using a Web Camera based on Colour Detection", IJCTT, vol.9, Mar 2014. "*