

# **AI VIRTUAL MOUSE**

## **A PROJECT REPORT**

**Submitted By**

**Apurva Gupta**  
**(University Roll No- 2100290140035)**  
**Chhavi Chaudhary**  
**(University Roll No- 2100290140054)**

**Submitted in partial fulfillment of the  
Requirements for the Degree of**

## **MASTER OF COMPUTER APPLICATIONS**

**Under the Supervision of  
Mr. Ankit Verma  
Assistant Professor**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS  
KIET Group of Institutions, Ghaziabad  
Uttar Pradesh-201206  
(June 2023)**

## **CERTIFICATE**

Certified that **Apurva Gupta (University Roll. No-2100290140035), Chhavi Chaudhary (University Roll. No-2100290140054)**, has carried out the project work having title “**AI VIRTUAL MOUSE**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself /herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Apurva Gupta (University Roll No- 2100290140035)**

**Chhavi Chaudhary (University Roll. No-2100290140054)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

**Mr. Ankit Verma**  
**Assistant Professor**  
**Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner**

**Signature of External Examiner**

**Dr. Arun Kumar Tripathi**  
**Head, Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

## **ABSTRACT**

This project is about how new technologies can be used to develop a Python application that enables the user to perform the task on a desktop machine without interacting with the mouse or using hand gestures. It will analyze the possible utility of one single piece of software as a programming application by looking at examples of intelligent programs with natural language processing that are now available with various categories of support.

Natural Language Processing is used to activate the ability to communicate socially by storing and evaluating information in the context of the user. It is suggested that new technology may soon redefine the concept of using the internet. Experiments conducted on this system, combined with user testing, have provided evidence that a basic programmed with natural language processing algorithms in the form of a virtual mouse task performer application, with basic natural language processing and the ability to function without the need for another type of human input (or programming), may already be viable.

## ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Mr. Ankit Verma Sir** for his guidance, help, and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi, Sir** Professor and Head of the Department of Computer Applications, for his insightful comments and administrative help.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, the completion of this work would not have been possible on time. They keep my life filled with enjoyment and happiness.

**APURVA GUPTA**  
**CHHAVI CHAUDHARY**

## LIST OF CHAPTERS

<b>1. Introduction</b>	<b>8-12</b>
1.1 Introduction	8
1.2 Literature Review	9
1.3 Objective And Issue	10
1.4 Scope	11
<b>2. Concepts And Techniques</b>	<b>13-18</b>
2.1 Methodology	13
2.2 Conceptual Review	15
2.3 Features And Scope	16
<b>3. System Design and Architecture</b>	<b>19-22</b>
3.1 Package Review	19
3.2 Architecture Of Programming	22
<b>4. Feasibility Study</b>	<b>23-25</b>
4.1 Technical Study	23
4.2 Operational Study	23
4.3 Economical Study	24
4.4 Behavioral Study	25
<b>5. Database Design</b>	<b>26-30</b>
5.1 Use Case Diagram	26
5.2 Sequence Diagram	28
5.3 Activity Diagram	29
<b>6. Testing</b>	<b>31-33</b>
6.1 Unit Testing	31
6.2 Integration Testing	31
6.3 System Testing	32
6.4 Acceptance Testing	32
6.5 Test Procedure	33
<b>7. Working</b>	<b>34-36</b>
<b>8. Testing and Result Analysis</b>	<b>37-41</b>
8.1 Overview	37
8.2 Recognition Calls	38
<b>9. Implementation</b>	<b>42-50</b>

9.1 Python	42
9.2 Implemented Code	43
<b>10. Conclusion</b>	<b>51-52</b>
<b>11. Future Work</b>	<b>53</b>
<b>12. Advantages</b>	<b>54</b>
<b>13. References</b>	<b>55</b>

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 INTRODUCTION**

The devices we use every day are getting smaller thanks to the advancement of Bluetooth or wireless technologies, as well as advancements in the field of augmented reality. This study suggests a computer vision-based AI virtual mouse system that uses hand motions and hand tip detection to emulate a mouse on a computer. The major goal of the suggested system is to replace the use of a typical mouse device with a web camera or a built-in camera in the computer to perform computer mouse cursor and scroll tasks. Computer vision is used to identify hand gestures and hand tips as part of human-computer interaction (HCI). By employing a built-in camera or a web camera to monitor the fingertip of the hand gesture, the AI virtual mouse system enables us to move the cursor and conduct mouse cursor actions.

While using a wireless or Bluetooth mouse, some devices such as the mouse, the dongle to connect to the PC, and a battery to power the mouse to operate are used, but in this paper, the user uses his or her built-in camera or a webcam and uses his or her hand gestures to control the computer mouse operations. In the proposed system, the web camera captures and then processes the frames that have been captured, recognizes the various hand gestures and hand tip gestures, and then performs the mouse function.

The AI virtual mouse system is created using the Python programming language, and OpenCV, a library for computer vision, is incorporated into the system. The Media Pipe package is used by the proposed AI virtual mouse system to track the hands and the tips of the hands, and Patagium packages were used to move around the computer's window screen and perform actions like left-clicking, right-clicking, and

scrolling. The suggested model's results demonstrated a very high level of accuracy, and it can operate flawlessly in practical situations using only a CPU and no GPU.

## **1.2 LITERATURE OVERVIEW**

The proposed AI virtual mouse system can be used to solve real-world issues including those where there isn't enough room to use a physical mouse and for people who have hand issues and can't use a physical mouse. The proposed AI virtual mouse can be used to solve these issues because hand gestures and hand Tip detection are used to control the PC mouse functions by using a webcam or a built-in camera. In the COVID-19 situation, it is also not safe to use the devices by touching them because it may result in a possible situation of the virus spreading by touching the devices. To design a virtual mouse that detects hand gesture patterns instead of a physical mouse.

- i. We use colored tips for detection which are captured by webcam.
- ii. Here, the colored fingertip acts as an object that the webcam senses.
- iii. The camera is positioned such that it recognizes the moment of fingertips and performs the operations of the mouse.
- iv. The utilization of virtual mouse appears in space-saving situations or in movement situations.

## **1.3 OBJECTIVES AND ISSUE**

The Objective and Issue related to project development are as follows: -

- i. The main goal of the proposed AI virtual mouse system is to create a replacement for the standard mouse system to perform and control the mouse functions. This can be done with the aid of a web camera that records hand gestures and the tip of the hand and then processes these frames to perform mouse functions like left-clicking, right-clicking, and scrolling.
- ii. The goal of this project is to create a Virtual Mouse application that focuses on a few key development areas. With the use of various image processing algorithms, this project intends to firstly do away with the necessity for a physical mouse while enabling webcam-based computer interaction. In addition, this project seeks to create a virtual mouse program that can be used in a variety of contexts and on a variety of surfaces. The overarching goals of this initiative are as follows:



- iii. To be designed to operate with the help of a webcam. The Virtual Mouse application is operational with the help of a webcam, as the webcam is responsible for capturing the images in real time. The application would not work if there were no webcams detected.to create a universally functional virtual input. If the user faces the webcam while making the motion gesture, the Virtual Mouse application will work on any surface and in any indoor setting.
- iv. Setting up the camera to keep taking pictures so that it can later analyses them using different image processing methods. As stated above, the Virtual Mouse application will continually capture the images while they are being processed, including HSV conversion, binary image conversion, let and pepper noise filtering, and other operations.
- v. To translate a hand motion or gesture into mouse input that is set to a specific screen position. The Virtual Mouse application will be programmed to recognize where the defined colors are in relation to where the mouse pointers will be positioned.

#### **1.4 SCOPE**

To improve convenience while maintaining the ability to precisely interact with and manage the computer system, Virtual Mouse will soon be developed to replace the physical computer mouse. To correctly track the user's gesture, the program must be quick enough to capture and process each image. The open-source computer vision library, often known as OpenCV, and the most recent software development techniques will be used in this project to create a software application. The project's scope is as follows:

- a) For most laptop touch pad is not the most comfortable and convenient.
- b) Virtual mouse, known as Virtual Multitask Mouse.
- c) This is a real time application.
- d) User friendly application.
- e) This project removes the requirement of having a physical.

When a user's gesture is caught in real time via a webcam, the application's workflow can begin. The obtained image is then processed for segmentation to determine which pixels have values that are equivalent to those of the set colour. After segmentation, the

entire image will be converted to a binary image, where the pixels that were identified will appear as white and the remaining pixels as black. In order to simulate the mouse pointer without utilizing a real computer mouse, the position of the white segment in the image will be captured and set as the position of the mouse pointer. The Windows operating system is compatible with the program. The functionality of the software will be programmed in C++, and an external library called OpenCV, which performs image processing, will be included.

The Virtual Mouse application is expected to replace the current methods of utilizing a physical computer mouse where the mouse inputs and positions are done manually. This application offers a more effortless way to interact with the computer system, where every task can be done by gestures. Furthermore, the Virtual Mouse application could assist the motor-impaired users where he/she, could interact with the computer system by just showing the correct combination of colors to the webcam.

## **CHAPTER 2**

### **CONCEPTS AND TECHNIQUES**

#### **2.1 METHODOLOGY**

In order to develop the application for this project, we'll follow the Agile Software Development methodology approach. By using incremental and iterative work, the project team can respond to unpredictability using the described technique, which is an alternative to the conventional waterfall paradigm. It encourages quick and adaptable responses to change and supports adaptive planning, evolutionary development, early delivery, and continual improvement. The guidelines for the Agile Software Development methodology are as follows:

- a. Satisfy the customer by early and continuous delivery of workable software.
- b. Encourage changes in requirements.
- c. Workable software is delivered frequently.
- d. Continuous collaboration between the stakeholders and the developers.
- e. Projects are developed around motivated individuals.
- f. Encourage informal meetings.
- g. Operational software is the principal measure of progress.
- h. Sustainable development, able to maintain a constant pace.
- i. Continuous attention to technical excellence and good design
- j. Simplicity
- k. Self-organizing teams
- l. Regular adaptation to changing circumstances.

The reason for choosing this methodology is that the Virtual Mouse is still considered to be at the introduction stage, which means it still requires a great deal of extensive research and development before it could make it into the market. Therefore, this project requires a thorough yet iterative planning and requirements gathering where the lifecycle will be continually revisited to re-evaluate the direction of the project and eliminate the ambiguities in the process of development, and at the same time welcome changes of requirements, which promotes adaptability and flexibility. Furthermore, due to the Virtual Mouse application being more towards serving the users, this project requires continuous customer collaboration, as they are essential for gathering the proper requirements in all aspects. Therefore, the agile methodology is the ideal approach for developing the project.

The Virtual Mouse Color Recognition requires being able to recognize most of the colors provided by the users with high accuracy, consistency, and minimal performance impact on other processes. However, the recognition results may vary whenever the qualities of the captured frames have changed, as they may be affected by the different situations in terms of environment, brightness, and weather. The following describes the situations which may result in false detection and/or any other problem that may occur during the recognition phase:

- i. The real-time images are taken under dark or bright environment conditions.
- ii. The real-time images are taken in a color conflict background.
- iii. The users interact with the program at near or far distances.
- iv. The real-time images are rotated in a clockwise or anti-clockwise rotation to achieve greater accuracy and consistency throughout the whole recognition cycle, a plan is required to be implemented for the program to perform flawlessly.

In order to help stroke victims, recover from the aftereffects of their condition, this research proposes the implementation of a computer program that employs alternate techniques to control keyboard and mouse cursors. As a result, we suggest a brand-new keyboard and mouse cursor control system that is based on the

vision and recognition technique and makes use of webcam-recorded hand movements.

## **2.2 CONCEPTUAL REVIEW**

As modern technology of human-computer interactions became important in our everyday lives, varieties of mouse of all kinds of shapes, and sizes were invented, from a casual office mouse to a hard-core gaming mouse. However, there are some limitations to this hardware as they are not as environmentally friendly as it seems. Foreexample, the physical mouse requires a flat surface to operate, not to mention that it requires a certain area to fully utilize the functions offered. Furthermore, some of this hardware is completely useless when it comes to interacting with the computers remotely due to the cable length limitations, rendering it inaccessible.

There are traditional approaches for virtual keyboard and mouse systems which are usually based on eye gestures. Our literature review focuses on the research works on virtual keyboards and virtual mice which were published in Elsevier, Springer, ACM (AWS Certificate Manager) Digital Library, IEEE Digital Library etc.

In 2016, S. Shetty et al. constructed a virtual mouse system using color detection. They used a webcam for detecting mouse cursor movement and click events using OpenCV built-in functions. A mouse driver, written in java, is required as well. This system fails to perform well in the rough background. P. C. dhe et al. expanded a method for mouse-free cursor control where mouse cursor operations are controlled by using hand fingers. They have collected hand gestures via webcam using color detection principles. The built-in function of the Image Processing Toolbox in MATLAB and a mouse driver, written in java, is used in this approach. The pointer was not too efficient on the air as the cursor was very sensitive to motion.

G. Sahu et al. built a system for controlling a mouse pointer using a webcam that controls the volume of the media player, and PowerPoint slides and can make or end a call. They used RGB color tapes to recognize the user's finger. In 2019, K. Hassan et al. presented a system to design and develop a hand gesture-based virtual mouse. They captured different gestures via webcam and performed mouse functions according to the gestures. This system achieved 78%-90% accuracy. The system does not work efficiently in a complex or rough background. As we can see from the

reviewed literature, previous systems include either a virtual keyboard or a virtual mouse. Those systems can't fully eliminate the need for a mouse and keyboard completely. This work aims to build an interactive computer system that can be operated without any physical mouse.

The current system is comprised of a generic mouse and trackpad monitor control system, as well as the absence of a hand gesture control system. The use of a hand gesture to access the monitor screen from a distance is not possible. Even though it is primarily attempting to implement, the scope is simply limited in the virtual mouse field. The existing virtual mouse control system consists of simple mouse operations using a hand recognition system, in which we can control the mouse pointer, left-click, right-click, drag, and so on. The use of hand recognition in the future will not be used. Even though there are a variety of systems for hand recognition, the system they used is static hand recognition, which is simply a recognition of the shape made by the hand and the definition of action for each shape made, which is limited to a few defined actions and causes a lot of confusion. As technology advances, there are more alternatives to using a mouse.

## 2.3 FEATURES AND SCOPE

- i. Features such as enlarging and shrinking windows, closing windows, etc. by using the palm and multiple fingers are our future scope for this project.
- ii. In the future, we plan to add more features such as enlarging and shrinking windows, closing windows, etc. by using the palm and multiple fingers.
- iii. We can also open the browser or any drives (C: /D:/E: etc.) with the help of hand gestures instead of moving the cursor.

The Virtual Mouse application is expected to replace the current methods of utilizing a physical computer mouse where the mouse inputs and positions are done manually. This application offers a more effortless way to interact with the computer system, where every task can be done by gestures. Furthermore, the Virtual Mouse application could assist the motor-impaired users where he/she could interact with the computer system by just showing the correct pattern of fingers to the webcam.

The code is written in Python, and it employs the cross-platform image processing module OpenCV as well as the Python-specific library PyAutoGUI to implement mouse actions.

Skin detection can be defined as detecting the skin color pixels in an image. It is a fundamental step in a wide range of image processing applications such as face detection, hand tracking and gesture recognition. Skin detection using color information has recently gained a lot of attention since it is computationally effective and provides robust information against scaling, rotation and partial occlusion. Skin detection using colour information can be a challenging task, since skin appearance in images is affected by illumination, camera characteristics, background, and ethnicity. To reduce the effects of illumination, the image can be converted to a chrominance color space, which is less sensitive to illumination changes. A chrominance color space is one where the intensity information (luminance), is separated from the color information.

## CHAPTER 3

### SYSTEM DESIGN AND ARCHITECTURE

#### 3.1 PACKAGE OVERVIEW

```
import cv2          #Real-time computer vision and image-processing framework
import mediapipe as mp #Open source and cross platform framework for ML solution such as hand recognisation and object detection
import pyautogui    #Is a python automation library used to click,drag,scroll,move or click on exact position
import math         #used for mathematical calculation and functions
from enum import IntEnum #creating enumerated constant
from ctypes import cast, POINTER #imported so that we can create pointers to structures and call their methods on them.
from ctypes import CLSCTX_ALL #comtypes are imported so that we can use COM interfaces with Python 3+.
from pycaw.pycaw import AudioUtilities, IAudioEndpointVolume #used to control device's speaker and its master volume
from google.protobuf.json_format import MessageToDict #Contains routines for printing protocol messages in json format
import screen_brightness_control as sbcontrol #to control brightness of screen
```

Fig 3.1: Imported packages

- i. The code starts by importing the necessary libraries.
- ii. Media Pipe offers cross-platform, customizable ML solutions for live and streaming media. End-to-End acceleration: Built-in fast ML inference and processing accelerated even on common hardware. Build once, deploy anywhere: Unified solution works across Android, iOS, desktop/cloud, web and IoT.
- iii. The cv2 library is used to import the functions for creating and manipulating images, while pyautogui is used to control a mouse cursor.
- iv. Next, from Enum import IntEnum is imported so that we can use enumerations instead of integers when defining colors.
- v. Then from type's import cast, POINTER are imported so that we can create pointers structures and call their methods on them.



Finally, cotyles are imported so that we can use COM interfaces with Python 3+.

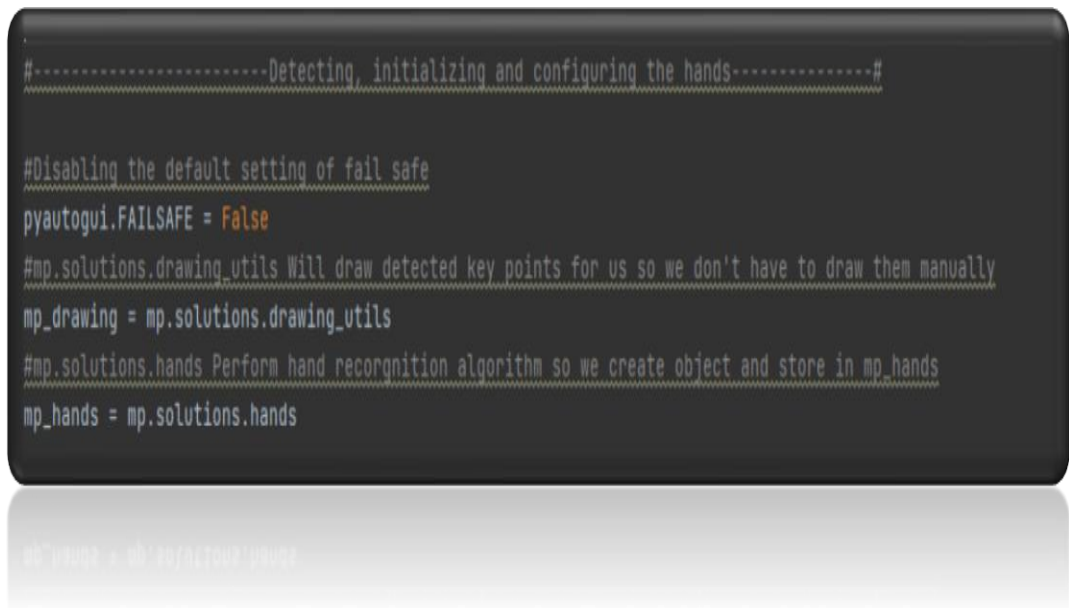


Fig 3.2: Code For Detection Of Hands

- i. Next comes an if statement which checks whether FAILSAFE has been set as False before continuing with the rest of the code block.
- ii. If it has been set as True, then some error handling occurs because there's no point in trying to draw something if it won't work properly without crashing due to memory issues or other errors occurring during execution of this program.
- iii. If FAILSAFE hasn't been set as False then a function called updrawing () is defined which draws shapes onto an image using drawing utils provided by MP Solutions (a company who provides solutions for multimedia applications).
- iv. The function takes two parameters.
- v. The code is a part of the PyCaw Library

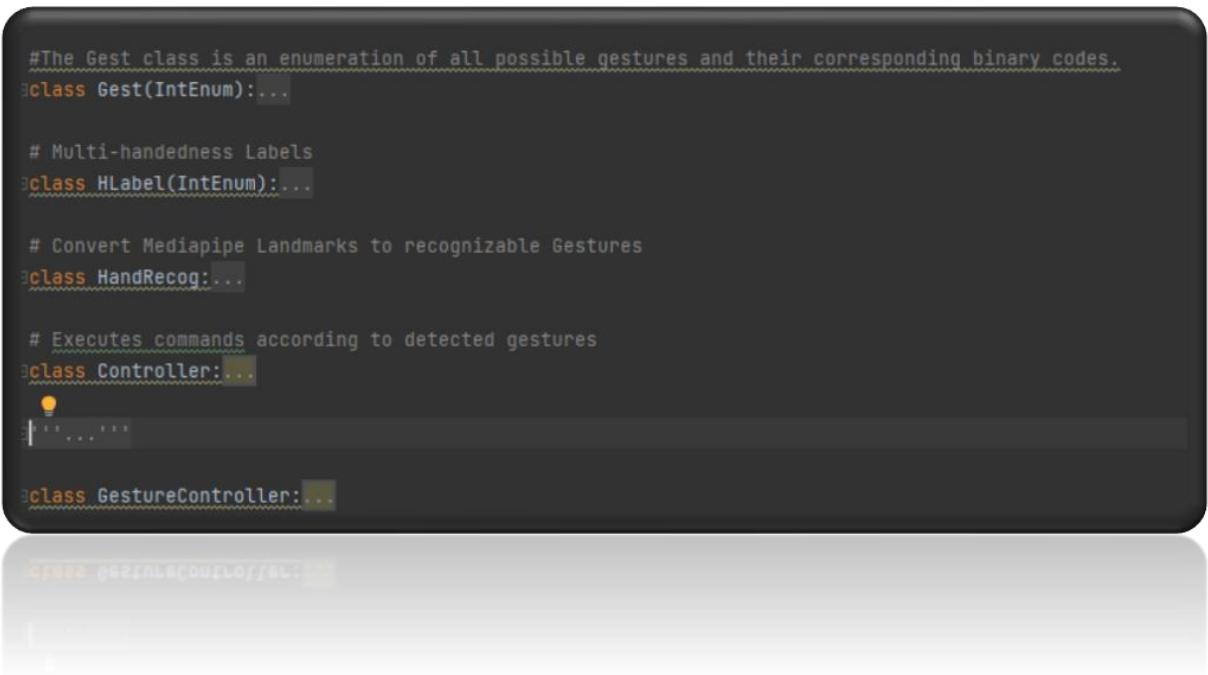


Fig 3.3 Database Recognition of gestures

- i. The gestures will be matched with the database.
- ii. The set functionalities will be matched with the gestures.
- iii. The Virtual Mouse will then perform the desired task.

### 3.2 ARCHITECTURE OF PROGRAMMING

The basic flow is described as shown in the image below-

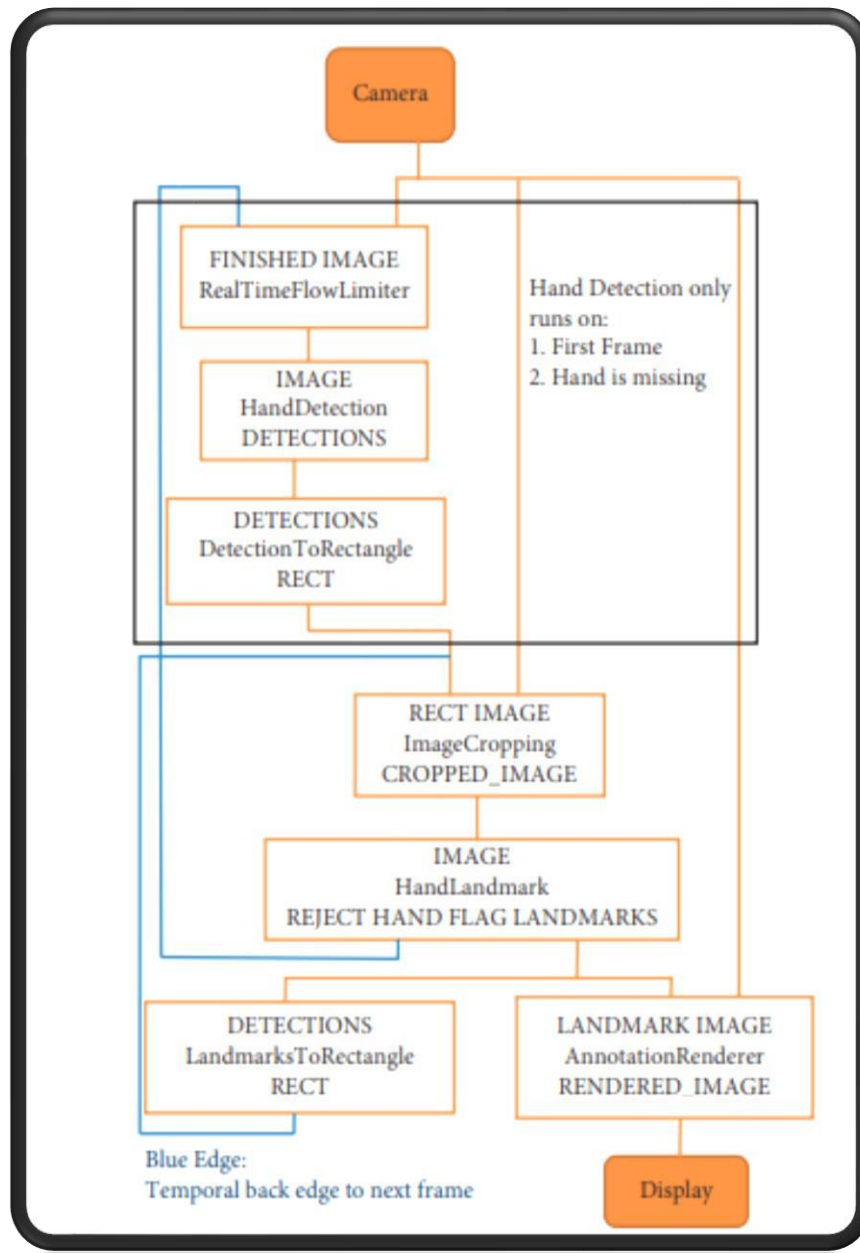


Fig 3.4: Basic Flow of Project

## **CHAPTER 4**

### **FEASIBILITY STUDY**

A feasibility analysis is used to assess an idea's viability, such as confirming that a project is technically and legally possible as well as financially justifiable. It reveals whether an endeavor is worthwhile; in other situations, it may reveal that an endeavor is impossible. There are a variety of potential causes for this, including the need for excessive resources, which not only hinders them from doing other duties but may also end up costing an organization more money than they would get back by taking on an unsuccessful project. A well-designed study should provide a historical background of the company or project, including a description of the good or service, accounting statements, information on operations and management, marketing research and policies, financial statistics, legal needs, and tax duties. These studies typically come before project implementation and technical development.

#### **4.1 TECHNICAL STUDY**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team can convert the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. It includes finding out technologies for the project, both hardware and software. For a virtual assistant, the user must have a microphone to convey their message and a speaker to listen to when the system speaks. These are very cheap nowadays and everyone generally possesses them. Besides, the system needs an internet connection. While using the assistant, make sure you have a steady internet connection. It is also not an issue in this era where almost every home.

## **4.2 OPERATIONAL FEASIBILITY**

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development. It is the ease and simplicity of operation of the proposed system. The system does not require any special skill set for users to operate it. This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all carried out by a single person. That won’t create any management issues and will increase the feasibility of the project.

## **4.3 ECONOMICAL FEASIBILITY**

In the Economic Feasibility study the cost and benefit of the project are analyzed. This means under this feasibility study a detailed analysis is carried out will be the cost of the project for development which includes all required costs final development hardware and software resources required, design and development costs and operational costs, and so on.

After that, it is analyzed whether the project will be beneficial in terms of finance for the organization or not. We find the total cost and benefit of the proposed system over the current system. For this project, the main cost is the documentation cost. Users would also have to pay for the microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, Assistant won’t cost too much.

## **4.4 BEHAVIOURAL STUDY**

It evaluates and estimates the user’s attitude or behavior toward the development of the new system. It helps in determining if the system requires special effort to educate, retrain, transfer, and change an employee's job status on new ways of conducting business. Establishing the cost-effectiveness of the proposed system i.e., if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need for online social networking facilities. Thus, the benefits

of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment is typically involved.

## **CHAPTER 5**

### **DATABASE DESIGN**

A properly designed database provides you with access to up-to-date, accurate information. Because correct design is essential to achieving your goals in working with a database, investing the time required to learn the principles of good design makes sense. In the end, you are much more likely to end up with a database that meets your needs and can easily accommodate change.

This article provides guidelines for planning a desktop database. You will learn how to decide what information you need, how to divide that information into the appropriate tables and columns, and how those tables relate to each other. You should read this article before you create your first desktop database.

Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of an enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there must be a brilliant concept for this.

A properly designed database provides you with access to up-to-date, accurate information. Because correct design is essential to achieving your goals in working with a database, investing the time required to learn the principles of good design makes sense. In the end, you are much more likely to end up with a database that meets your needs and can easily accommodate change.

This article provides guidelines for planning a desktop database. You will learn how to decide what information you need, how to divide that information into the appropriate tables and columns, and how those tables relate to each other. You should read this article before you create your first desktop database.

Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of an enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept for designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

## **5.1 USE CASE DIAGRAM**

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

The purpose of a use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as the other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will investigate some specific purpose, which will distinguish it from the other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case.



## **5.2 SEQUENCE DIAGRAM**

A sequence diagram or system sequence diagram (SSD) shows object interactions arranged in time sequence in the field of software engineering. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

For a particular scenario of a use case, the diagrams show the events that external actors generate, their order, and possible inter-system events. All systems are treated as a black box; the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram should be made for the main success scenario of the use case, and frequent or complex alternative scenarios.

- To model high-level interaction among active objects within a system.
- To model interaction among objects inside realizing a use case.
- It either models' generic interactions or some certain instances.

## **5.3 ACTIVITY DIAGRAM**

The activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

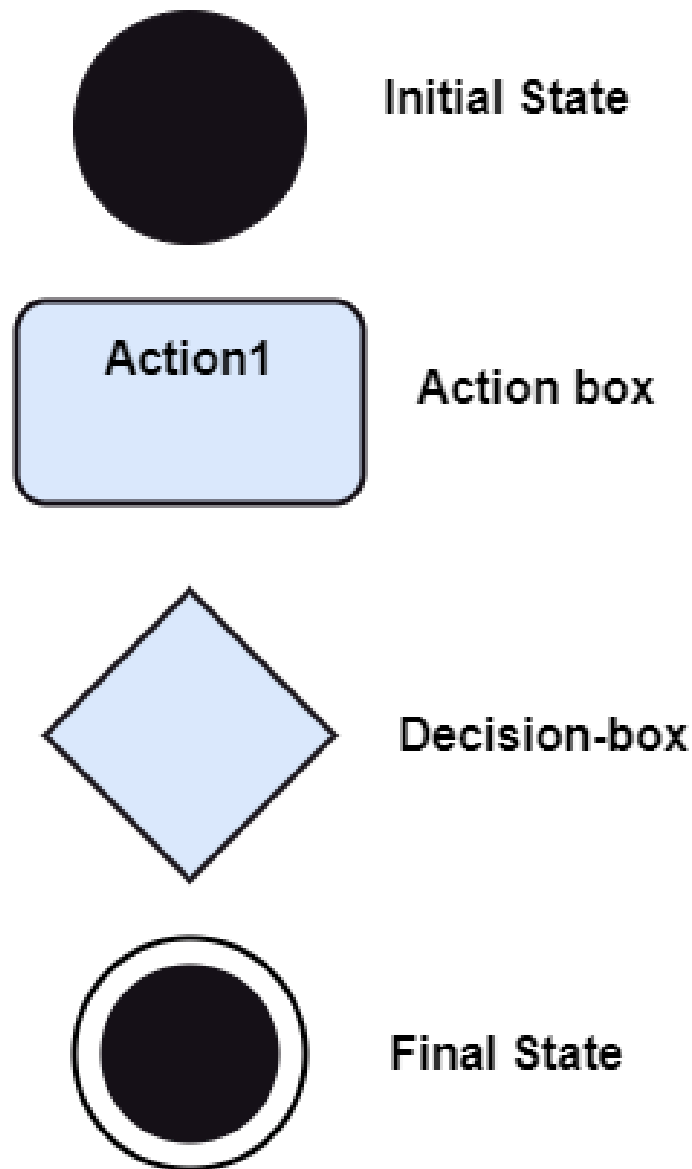


Fig 5.1 Activity Diagram Notations

An event is created as an activity diagram encompassing a group of nodes associated with edges. To model the behavior of activities, they can be attached to any modeling element. It can model use cases, classes, interfaces, components, and collaborations.

It mainly models processes and workflows. It envisions the dynamic behavior of the system as well as constructs a runnable system that incorporates forward

and reverse engineering. It does not include the message part, which means message flow is not represented in an activity diagram.

It is the same as that of a flowchart but not exactly a flowchart itself. It is used to depict the flow between several activities. An activity diagram is a flowchart of activities, as it represents the workflow among various activities. They are identical, but they themselves are not exactly the flowchart. In other words, it can be said that an activity diagram is an enhancement of the flowchart, which encompasses several unique skills.

Since it incorporates swim lanes, branching, parallel flows, join nodes, control nodes, and forks, it supports exception handling. A system must be explored before drawing an activity diagram to provide a clearer view of the user. All of the activities are explored after they are properly analyzed to find out the constraints applied to the activities. Each activity, condition, and association must be recognized. After gathering all the essential information, an abstract or a prototype is built, which is then transformed into the actual diagram.

## **CHAPTER 6**

### **TESTING**

#### **6.1 UNIT TESTING**

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance. A unit is a single testable part of a software system and tested during the development phase of the application software.

The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers. Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one.

#### **6.2 INTEGRATION TESTING**

Integration testing is the second level of the software testing process that comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Unit Testing uses modules for testing purposes, and these modules are combined and tested in integration testing. The Software is developed with several software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

### **6.3 SYSTEM TESTING**

System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system. In other words, a computer system consists of a group of software to perform the various tasks, but only software cannot perform the task; for that software must be interfaced with compatible hardware. System testing is a series of different types of tests with the purpose of exercising and examining the full working of an integrated software computer system against requirements.

To check the end-to-end flow of an application or the software as a user is known as System testing. In this, we navigate (go through) all the necessary modules of an application and check if the end features or the end business works fine and test the product as a whole system. It is end-to-end testing where the testing environment.

### **6.4 ACCEPTANCE TESTING**

Acceptance testing is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not. It is conducted as a kind of Black Box testing where the number of required users is involved in testing the acceptance level of the system. It is the fourth and last level of software testing.

User acceptance testing (UAT) is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer (domain expert) for their satisfaction, and checks whether the application is working according to given business scenarios, real-time scenarios. In this, we concentrate only on those features and scenarios which are regularly used by the customer or mostly user scenarios for the business or those scenarios which are used daily by end-user customer.

## **6.5 TEST PROCEDURES**

A test procedure is a formal specification of test cases to be applied to one or more target program modules. Test procedures are executable. A process called the VERIFIER applies a test procedure to its target modules and produces an exception report indicating which test cases, if any, failed.

Test procedures facilitate thorough software testing by allowing individual modules or arbitrary groups of modules to be thoroughly tested outside the environment in which they will eventually reside. Test procedures are complete, self-contained, self-validating and executed automatically. Test procedures are a deliverable product of the software development process and are used for both initial checkout and subsequent regression testing of target program modifications.

Test procedures are coded in a new language called TPL (Test Procedure Language). The paper analyzes current testing practices, describes the structure and design of test procedures, and introduces the Fortran Test Procedure Language.

## CHAPTER 7

### WORKING

Capturing real-time video using Web-Camera: We will need a sensor to detect the user's hand movements for the system to work. As a sensor, the computer's webcam is used. The webcam records real-time video at a fixed frame rate and resolution determined by the camera's hardware. If necessary, the system's frame rate and resolution can be changed.

Converting the video captured into HSV format: The video has also been converted into HSV (hue, saturation, meaning, also called HSB), an alternative representation of the RGB color model created by computer graphics researchers to better reflect the perception of colored characteristics by human vision.



Fig 3.5: HSV Image

Each image frame is processed separately: Following the capture of the video, it goes through a brief pre- processing stage before being processed one frame at a time.

Conversion of each frame to a greyscale image: A grayscale image has lower computational complexity than a colored image. It also aids in faster color calibration without the use of external noise. All the necessary operations were carried out after the image was converted to grayscale.

Calibrate the color ranges: The device enters the calibration mode after the above steps, which assigns each color according to the HSV rule to its color hue, saturation or value values. Every color already has its predetermined values. For accurate color detection, the user can adjust the ranges. In the diagram below you can clearly see the variety of values used to detect every color cap.

Calculate the image's centroid by locating the image's region. To guide the mouse pointer, the user must first choose a point whose coordinates can be sent to the cursor. The device can monitor cursor movement using these coordinates. As the object travels around the frame, these coordinates change over time.

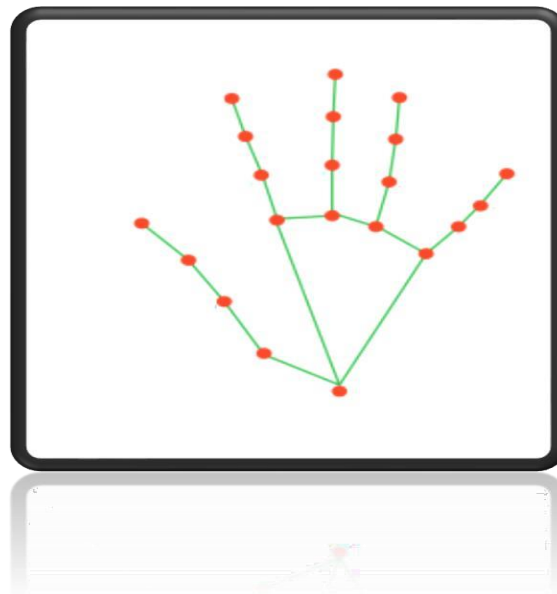


Fig 3.6: Co-ordinates



Tracking the mouse pointer. After determining the coordinates, the mouse driver is accessed, and the coordinates are sent to the cursor. The cursor positions itself in the required position using these coordinates. As a result, the mouse moves proportionally across the screen as the user moves his hands across the camera's field of view.

The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project.

Simulating the mouse's actions. To create the control actions in simulation mode, the user must make hand gestures. The computation time is reduced due to the use of color pointers.

## CHAPTER 8

### TESTING AND RESULT ANALYSIS

#### 8.1 OVERVIEW

To achieve accuracy, and consistency of the Virtual Mouse color recognition, testing phases have been conducted on various scenarios. The purpose of the testing phase is to ensure that the final deliverable is able to perform flawlessly in terms of accuracy, consistency, and performance. To achieve that, the program has to be able to recognize the colors input provided by the users with minimal adjustment, providing that the colors are thoroughly calibrated at first-hand. Furthermore, the program requires a quire to be able to execute the mouse functions efficiently and accurately as well.

#### 8.2 RECOGNIZATION CALLS

**Gesture Recognition:** Following are the results of various hand gestures that result into mouse operations:

- **Neutral Gesture:** Palm: Used to halt/stop execution of current gesture.



Fig 8.1: Neutral Gesture

- **Move Cursor**

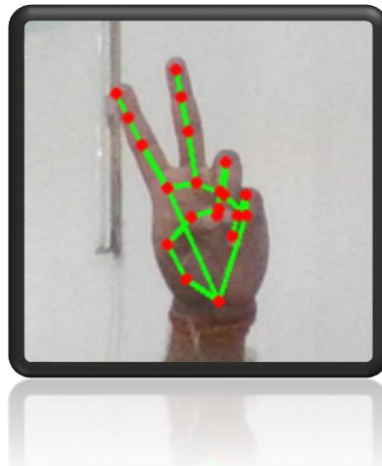


Fig 8.2: Move Curser Gesture

Cursor is assigned to the midpoint of index and middle fingertips. This gesture moves the cursor to the desired location. The speed of the cursor movement is proportional to the speed of hand.

- **Left Click**



Fig 8.3: Left Click Gesture

- **Double Click**

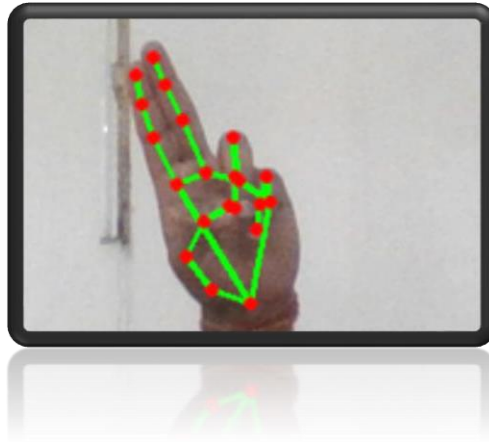


Fig 8.4: Double Click Gesture

- **Scrolling**



Fig 8.5: Scrolling Gesture

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.

The below two are **pinch gestures** like scrolling gestures:

- **Volume Control**

Dynamic Gestures for Volume control. The rate of increase/decrease of volume is proportional to the distance moved by pinch gesture from start point.



Fig 8.6: Volume Control gesture

- **Brightness Control**



Fig 8.7: Brightness Control gesture

Dynamic Gestures for Brightness control. The rate of increase/decrease of brightness is proportional to the distance moved by the pinch gesture from the start point.

- **Drag and Drop**

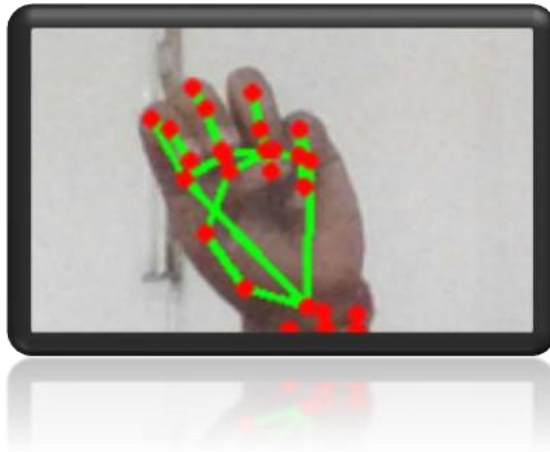


Fig 8.8: Drag and Drop Gesture

Gesture for drag and drop functionality. Can be used to move/transfer files from one directory to another.

- **Multiple Item Selection**

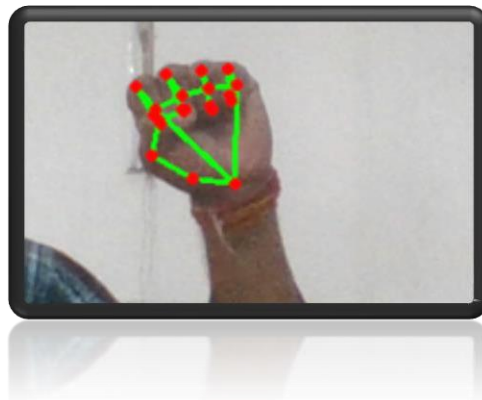


Fig 8.9: Multiple Item

## **CHAPTER 9**

### **IMPLEMENTATION**

#### **9.1 PYTHON**

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefits to all.

The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For Virtual Mouse, libraries used are speech recognition to recognize voice, Pytsx for text to speech, selenium for web automation etc. Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

Pytube is a Python library specifically designed for working with YouTube videos. It provides a convenient and easy-to-use API for fetching video metadata, downloading videos, and handling various video formats and quality options. Here are some key features and functionalities of Pytube:

## 9.2 IMPLEMENTED CODE

```
class Controller:

    tx_old = 0

    ty_old = 0

    trial = True

    flag = False

    grabflag = False

    pinchmajorflag = False

    pinchminorflag = False

    pinchstartxcoord = None

    pinchstartycoord = None

    pinchdirectionflag = None

    prevpinchlv = 0

    pinchlv = 0

    framecount = 0

    prev_hand = None

    pinch_threshold = 0.3

    def getpinchylv(hand_result):

        dist = round((Controller.pinchstartycoord -
            hand_result.landmark[8].y)*10,1)

        return dist

    def getpinchxlv(hand_result):
```



```

dist = round((hand_result.landmark[8].x -
Controller.pinchstartxcoord)*10,1)

return dist

def changesystembrightness():

currentBrightnessLv = sbcontrol.get_brightness()/100.0

currentBrightnessLv += Controller.pinchlv/50.0

if currentBrightnessLv > 1.0:

currentBrightnessLv = 1.0

elif currentBrightnessLv < 0.0:

currentBrightnessLv = 0.0

sbcontrol.fade_brightness(int(100*currentBrightnessLv) , start =

sbcontrol.get_brightness())

def changesystemvolume():

devices = AudioUtilities.GetSpeakers()

interface = devices.Activate(IAudioEndpointVolume._iid_,

CLSCTX_ALL, None)

volume = cast(interface, POINTER(IAudioEndpointVolume))

currentVolumeLv = volume.GetMasterVolumeLevelScalar()

currentVolumeLv += Controller.pinchlv/50.0

if currentVolumeLv > 1.0:

currentVolumeLv = 1.0

elif currentVolumeLv < 0.0:

currentVolumeLv = 0.0

```

```

volume.SetMasterVolumeLevelScalar(currentVolumeLv, None)

def scrollVertical():

    pyautogui.scroll(120 if Controller.pinchlv>0.0 else -120)

def scrollHorizontal():

    pyautogui.keyDown('shift')

    pyautogui.keyDown('ctrl')

    pyautogui.scroll(-120 if Controller.pinchlv>0.0 else 120)

    pyautogui.keyUp('ctrl')

    pyautogui.keyUp('shift')

# Locate Hand to get Cursor Position

# Stabilize cursor by Dampening

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 pinch_control_init(hand_result):

Controller.pinchstartxcoord = hand_result.landmark[8].x

Controller.pinchstartycoord = hand_result.landmark[8].y

Controller.pinchlv = 0

Controller.prevpinchlv = 0

Controller.framecount = 0

# Hold final position for 5 frames to change status

def pinch_control(hand_result, controlHorizontal,
controlVertical):

if Controller.framecount == 5:

Controller.framecount = 0

Controller.pinchlv = Controller.prevpinchlv

```

```

if Controller.pinchdirectionflag == True:

    controlHorizontal() #x

elif Controller.pinchdirectionflag == False:

    controlVertical() #y

lvx = Controller.getpinchxlv(hand_result)

lvy = Controller.getpinchylv(hand_result)

if abs(lvy) > abs(lvx) and abs(lvy) > Controller.pinch_threshold:

    Controller.pinchdirectionflag = False

    if abs(Controller.prevpinchlv - lvy) < Controller.pinch_threshold:

        Controller.framecount += 1

    else:

        Controller.prevpinchlv = lvy

        Controller.framecount = 0

elif abs(lvx) > Controller.pinch_threshold:

    Controller.pinchdirectionflag = True

    if abs(Controller.prevpinchlv - lvx) < Controller.pinch_threshold:

        Controller.framecount += 1

    else:

        Controller.prevpinchlv = lvx

        Controller.framecount = 0

def handle_controls(gesture, hand_result):

    x,y = None,None

```

```

if gesture != Gest.PALM :

x,y = Controller.get_position(hand_result)

# flag reset

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

# implementation

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)

```

The first variable is called "Controller" and it is a class that represents the controller of the game. Next, there are two variables for tracking how many times we have been hit with our hand (tx\_old) and how many times we have touched our opponent's screen (told).

These values start at 0 because they represent when the program starts. The next variable is called "trial" which tracks whether you are currently showing a hands game or not. If trial equals True then grab flag will also equal True, otherwise grab flag will equal False.

Next, there are three flags: flag, grab flag, and pinchmajorflag which track if your character has grabbed their opponent's screen yet during this round of play; these flags start off as False because they represent when the program starts.

There are also two more flags: pinchminorflag and pinchstartxcoord which track where your character started to touch their opponent's screen on their left side; these values start out as None because they represent when the program starts. Lastly, there is one last flag called "pinchdirectionflag".

## **CHAPTER 10**

### **CONCLUSION**

In conclusion, it should come as no surprise that the physical mouse will eventually be replaced by a virtual, non-physical one in Human-Computer Interactions (HCI), where every mouse movement can be performed anywhere, anytime, and without regard to the environment. This research created a program that can recognize color movements and combinations and transform them into actual mouse functions in order to replace the standard physical mouse without sacrificing precision and efficiency. A few strategies had to be used because accuracy and efficiency were crucial factors in making the program as helpful as a real mouse. In order to minimize and stabilize the sensitivity of cursor movements, as minor movement could result in undesirable cursor movements, the coordinates of the colors responsible for managing cursor movements are first averaged based on a collection of coordinates. In addition, several color combinations were developed with the addition of distance computations between the two hues that made up the combination because a difference in distance can result in a difference in how the mouse behaves. This implementation's goal is to make it easier to control the program with minimal fuss. As a result, accurate mouse function triggering can be achieved with little trial and error.

Overall, modern technology has made significant progress in improving society's quality of life and productivity, not the other way around. Societies shouldn't coexist with older technology while resisting the changes brought about by newer ones. Instead, it is advised that individuals accept changes to lead more productive, efficient lives.

A Web camera is running on the mouse cursor. This will also lead to new levels of human-computer interaction(HCI), which does not require physical contact



with the device. This machine can perform all mouse tasks centered on color recognition. This device is capable of being useful for interacting with contactless input modes. For people who don't use a touchpad, it's helpful. The architecture of the device proposed would dramatically change people's interactions with computers. Everyone is compatible with the Webcam, the microphone, and the mouse. It would eliminate the need for a mouse completely. It can also be used in gaming or any other independent application. Free movement, left-click, right-click, drag/select, scroll-up, and scroll-down are all operations that can be performed using only gestures in this Multi-Functional system. The majority of the applications necessitate additional hardware, which can be quite costly. The goal was to develop this technology as cheaply as possible while also using a standardized operating system. Various application programs can be written specifically for this technology in order to create a wide range of applications with minimal resources.

Most of the applications require additional hardware which is often expensive. The motive of this work is to create this technology as cheaply as possible and to create it under a standardized operating system as well. Though our system can be used as an alternative to a physical keyboard and mouse, it still may perform less accurately in low light conditions. This is a concern for further research. Moreover, the work can be extended to a wide variety of environments and can be tested using sophisticated existing models. .

## **CHAPTER 11**

### **FUTURE WORK**

The mouse forms an integral part of the computer system. Our system architecture can facilitate the use of computers for paralyzed people. We have developed a virtual system where people can communicate with the computer without using a physical mouse. This could lead to a new age of Human-Computer Interaction in which physical contact with the computer would not be necessary at all.

The use of object detection and image processing in OpenCV for the implementation of our work has proved to be practically successful and the task of the mouse is achieved with good precision. This system can be beneficial to certain people who have no control over their limbs. The present application though seems to be feasible and more user friendly.

An attempt to make the input modes less constraints dependent for the user's hand gestures has been preferred. Another important aspect for the related development could be the design of an independent gesture vocabulary framework. The color detection algorithm can cause detection problems if another colored rubber is in working domain of webcam. There are several features and improvements needed for the program to be more user-friendly, accurate, and flexible in various environments. The following describes the improvements and the features required: Smart Recognition Algorithm Due to the current recognition process being limited to a 25cm radius, an adaptive zoom-in/out functions are required to improve the covered distance, where it can automatically adjust the focus rate based on the distance between the users and the webcam. Better Performance The response time heavily relies on the hardware of the machine; this includes the processing speed of the processor, the size of the available RAM, and the available features of the webcam.

## **CHAPTER 12**

### **ADVANTAGES**

- i. The main advantage of using hand gestures is to interact with the computer OS a non-contact human-computerinput modality.
- ii. Reduce hardware costs by eliminating the use of the mouse.
- iii. Convenient for users not comfortable with the touchpad.
- iv. The framework may be useful for controlling different types of games and other applications dependent on thecontrol through user-defined gestures.
- v. We are developing a system to control the mouse cursor using a real-time camera.
- vi. This system is based on computer vision algorithms and can do all mouse tasks.
- vii. However, it is difficult to get stable results because of the variety of lighting and skin colors of human races.
- viii. This system could be useful in presentations and to reduce workspace.
- ix. Features such as enlarging and shrinking windows, closing windows, etc. by using the palm and multiple fingers.

## **REFERENCES**

### **12.1 RESEARCH PAPERS**

- [1] Angel, Neethu's," Real Time Static & Dynamic Hand Gesture Recognition", International Journal of Scientific& Engineering Research Volume 4, Issue3, March-2013.
- [2] Chen-Chaung Hsieh and Dung-Hua Liou," A Real Time Hand Gesture Recognition System Using Motion HistoryImage" asps, 2010.
- [3] Holon Park, "A Method for Controlling the Mouse Movement using a Real Time Camera", Brown University,Providence, RI, USA, Department of computer science, 2008.
- [4] Eidem, E. Yardimci, Y. Atalay, V. Cetin, A. E. "Computer vision-based mouse", Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASS). IEEE International Conference

### **12.2 ONLINE SITES AND CHANNELS**

- i. Code With Harry
- ii. Stack Overflow
- iii. JavatPoint

