A PROJECT REPORT ON

VIRTUAL MOUSE USING HAND GESTURES

A dissertation submitted in partial fulfillment of the Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

SUBMITTED BY

G. KARTHIK 20JQ1A0508

Under the Esteemed Guidance of

Ms.CH. SUREKHA, M.Tech

Asst. Professor

Department of Computer Science & Engineering

KAKINADA INSTITUTE OF TECHNOLOGY AND SCIENCE

(Affiliated to JNT University, Kakinada, Approved by A.I.C.T.E., New Delhi, Accreditedby NAAC with 'A' Grade)

TIRUPATHI(V), DIVILI -533433 KAKINADA, A.P

2020-2024

KAKINADA INSTITUTE OF TECHNOLOGY AND SCIENCE

(AFFILIATED TO J.N.T. UNIVERSITY, APPROVED BY A.I.C.T.E., NEW DELHI,

Accredited by NAAC with A Grade)

KAKINADA, TIRUPATI (V)-533433.KAKINADA Dist. A.P.

CERTIFICATE

This is to certify that the project work entitled " **VIRTUAL MOUSE USING HAND GESTURES"** is Done by **"G. KARTHIK (20JQ1A0508)"** student of Department of Computer Science &Engineering is a record of Bonafede work carried out by her under my guidance and supervision during 2023-2024.This project is done as a partial fulfillment of obtaining Bachelor of Technology Degree to be awarded by Jawaharlal Nehru Technological University, Kakinada. The matter embodied in this project report has not been submitted to any other university for the award of any other degree.

Signature of Guide

Ms. CH. SUREKHA, M.Tech

Asst. Professor of CSE

Signature of HOD

Mr.B.VEERENDRA, M.Tech,(Ph.D)

Assoc. Professor of CSE

Signature of the External Examiner

ACKNOWLEDGEMENT

I take this moment to thank numerous people who lent their constant support encouragement for the successful completion of this project.

It's my privilege and delight to state my profound unfathomed of respect and gratitude to my estimable and mentor **Ms. CH. SUREKHA** Madam, Assist.Professor, Computer Science and Engineering, KITS, Divili, for her constructive channelizing and meticulous guidance throughout the course of B. Tech and also in completion of the dissertation work.

My heartfelt gratitude to **Mr. B. VEERENDRA** sir, Head of the Department of Computer Science & Engineering for encouragement and assistance to me, which contribute to the successful completion of this project.

I feel elated to thank **Dr. E. SARVARAMESWARUDU** sir, principal of Kakinada Institute Of Technology and Science for his cooperation and help in completionof my project and throughout my course.

I thank my honorable Chairman **Sri B. SRINIVASA RAO** correspondent,Kakinada Institute of Technology and Science for providing me with good faculty support throughout the course.

I express my sincere thanks to all the Teaching, Non-Teaching staff of Computer Science & Engineering department for their encouragement towards me for the successful completion of the project.

My heartful and sincere gratitude to my lovable friends and my beloved parents for their tremendous motivation and moral support.

BY

G. KARTHIK 20JQ1A0508

DECLARATION

I here by that the project entitled "**VIRTUAL MOUSE USING HAND GESTURES**" has been undertaken by me and this work has been submitted to the **KAKINADA INSTITUTE OF TECHNOLOGY & SCIENCE, TIRUPATI**(V) Affiliated to the **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science Engineering. I further declare that this project work has not been submitted in partial fulfillment of the award of any degree of this or any other educational institutions.

PROJECT ASSOCIATES:

G. KARTHIK 20JQ1A0508

INDEX

Chapter no.	Title	Page no.
	ACKNOWLEDGEMENT	
	DECLARATION	
	ABSTRACT	
1.	**INTRODUCTION**	1
	1.1 Introduction	2
	1.2 Existing System	2
	1.3 proposed system	2
	1.4 Scope and Purpose	2
	1.5 System overview	3
2.	**LITERATURE SURVEY**	4
	2.1 Virtual mouse history	4
	2.2 Virtual hand recognition	5
	2.3 Types of controls	8
3.	**REQIREMENT ANALYSIS**	10
	3.1 Introduction	10
	3.2 Modular description & Functional requirements	10
	3.3 Software Requirements	11
	3.4 Hardware Requirements	12
4.	**SYSTEM DESIGN**	13
	4.1 System Architecture	13
	4.2 Class Diagram	14
	4.3 Use case Diagram	15
	4.4 Activity Diagram	16
	4.5 State chart Diagram	17
	4.6 Sequence Diagram	18
	4.7 Collaboration Diagram	19
	4.8 Component Diagram	20

	4.9 Deployment Diagram	21
5.	**IMPLIMENTATION ISSUES**	22
	5.1 Python Framework	22
	5.2 Features of python	24
	5.3 Web scraping applications	28
	5.4 History of python	29
	5.5 Versions	31
	5.6 Python IDE's	33
	5.7 Python Architecture	39
	5.8 Fragments	41
	5.9 python UI Layouts	44
	5.10 Open computer vision	47
	5.11 K-NN Algorithm	50
	5.12 Proposed Algorithm	52
	5.13 Transitional Algorithm	53
6.	**TESTING**	55
	6.1 Types of testing	56
	6.2 Test Cases	58
7.	**SAMPLE CODE**	60
8.	**OUTPUT SCREENS**	67
9.	**CONCLUSION & FUTURESCOPE**	74
10.	**BIBLIOGRAPHY**	75

LIST OF TABLES

S.NO	Table No.	Name of Table	Page No.
1	5.1	Versions of Python	31
2	6.1	Test Cases	58

LIST OF FIGURES

S.NO.	Figure No.	Name of the figures	PageNo.
1	4.1	System Architecture	13
2	4.2	Class Diagram	14
3	4.3	Use Case Diagram	15
4	4.4	Activity Diagram	16
5	4.5	State Chart Diagram	17
6	4.6	Sequence Diagram	18
7	4.7	Collaboration Diagram	19
8	4.8	Component Diagram	20
9	4.9	Deployment Diagram	21
10	5.1	Icon of Python	22
11	5.2	Features of Python	24
12	5.3	Python IDEs	33
13	5.4	Python Architecture	39
14	5.5	Pack layout manager	44
15	5.6	Grid layout manager	45
16	5.7	K-NN Algorithm Working	50
17	5.8	Proposed Algorithm	52
18	5.9	Transitional Algorithm	53
19	6.1	Levels of Testing	56

ABSTRACT

The main goal of the project is to move the mouse pointer on the screen without using any hardware, such as a mouse, and instead by utilizing finger motions, i.e. the gesture recognition process. Different technologies have been explored in the development of virtual mouse in recent years. We are working with to make our devices more interact and work with minimal physical contact. In the project, We suggest interactive computer system without using hardware mouse. This device has potential to benefit everyone ,Who have difficulty using hardware mouse. This project proposes a way to control the position of the cursor with the hands without using any electronic devices or hardware gadgets. While the operations like clicking, scrolling and dragging by using different hand gestures. We can simply use web-cam as an input. Hand gesture are the effortless manner of communicating. The Desired and required output will be displayed on monitor.

1. INTRODUCTION

1.1 INTRODUCTION

This project is to develop a virtual mouse using hand gesture recognition. This hand gestures are the most effortless and normal way of communication. The aim is to perform various operation of the cursor. Instead of using more expensive sensors, A simple web camera can identify the gestures and perform the action. The basic goal of VIRTUAL MOUSE USING HAND GESTURES is to improve the interaction between users and computers. Recent days we are using hardware mouse as an navigating device to a specified user for their desired operations. The virtual mouse is used to reduce the space for using the physical mouse, and it can be used in many situations where we cannot able to use the physical mouse.By using hand gestures we can implement without any physical connection between the user and system.Instead of using more expensive sensors, a simple web camera can identify the gestures and perform the action. A virtual mouse using hand gesture is a computer system that allows users to control the cursor on their computer screen using hand gestures, instead of a physical mouse. This system uses a camera or other sensor to track the user's hand movements, and then translates those movements into cursor movements on the screen. By using hand gestures, users can interact with their computers in a more intuitive and natural way, without the need for a physical device.

Virtual mice using hand gestures can be particularly useful for people who have difficulty using a physical mouse, such as those with motor impairments or disabilities. They can also be used in situations where a physical mouse is not practical or convenient, such as in presentations or when giving demonstrations.

The technology behind virtual mice using hand gestures involves computer vision and machine learning algorithms. Computer vision algorithms are used to track the user's hand movements in real-time, while machine learning algorithms are used to recognize specific hand gestures and translate them into cursor movements or other actions on the computer.

Overall, virtual mice using hand gestures are a promising technology that has the potential to improve accessibility and ease of use for computer users. As the technology advances, it may also enable new forms of interaction and control, such as controlling a computer using hand gestures from a distance.

1.2 EXISTING SYSTEM

The existing system consist of a mouse that can be either wireless and wired to control the cursor. Mouse is a small, movable device. An important function of the mouse is to move the cursor from one place to another place, open and closean application, select the folders and files. It Perform basic mouse operations like minimize,drag, scroll up, scroll down, left-click, right-click using physical hardware device. Mouse is the basic input and output to computers and the use of both these devices require the useof hands.

The basic input and output to computers and the use of both these devices require the use of hands.

1.3 PROPOSED SYSTEM

The system works by Identifying the color of the hand and Decides The position of the cursor. We can use hand gestures to monitoring the system. Virtual mouse control system consists of the simple mouse operating using the fingure tips for which are captured by web cam. It is also tend to use to increase and decrease brightness of the system. In this proposed system we can also implement to the increase and decrease of the volume of thesystem by using special hand gestures. The proposed system can able to work for any skintone of any colour as well as we can accurately use it at minimal lighting condition.

1.4 SCOPE AND PURPOSE

Scope

Since use of hand gestures is increasing in our day-to-day life, we can add many things to it in future. For hand gesture we can following things:

- Enhancement of our system to control the whole computer system.
- Development of Gesture based passwords

Purpose

Technologies which will completely remove mouse dependency can be developed with the help of hand gesture based control system.

1.5 SYSTEM OVERVIEW

A virtual mouse using hand gesture is a computer system that allows users to interact with their computers by using hand gestures instead of a physical mouse. The system uses a camera or sensor to capture the user's hand movements, which are then processed to determine the intended action, such as moving the cursor or clicking.

Sensor: The system requires a sensor, such as a camera or depth sensor, to capture the user's hand movements.

Hand tracking: The system uses computer vision algorithms to track the user's hand movements in real-time. This involves identifying the hand in the captured image or video stream, tracking its movement over time, and estimating its position and orientation.

Gesture recognition: The system uses machine learning algorithms to classify the user's hand gestures into specific actions, such as moving the cursor, clicking, or scrolling. The algorithms can be trained on a dataset of hand gestures to learn to recognize common gestures.

Cursor control: Once the user's hand movements have been recognized as a cursor movement, the system translates the hand movement into a corresponding cursor movement on the screen.

Clicking and scrolling: The system also recognizes hand gestures that correspond to clicking or scrolling actions. For example, a finger tap or a hand gesture can be used to simulate a mouse click, and a two-finger swipe gesture can be used to simulate a scroll.

User interface: The virtual mouse system typically includes a graphical user interface (GUI) that displays the cursor and other user interface elements. The user can interact with the GUI using the virtual mouse and perform tasks such as opening applications, selecting menu items, and typing text.

Overall, a virtual mouse using hand gestures can provide an intuitive and natural way to interact with a computer, especially for users who have difficulty using a physical mouse or other input devices. The system can also enable new forms of interaction and control, such as controlling a computer using hand gestures from a distance.

2.LITERATURE SURVEY

2.1 VIRTUAL MOUSE HISTORY

Here, we categorize seven of the recent research papers on Vision base Hand Gesture Recognition. In this section an overview of different architectural approaches used to build Hand gesture applications is given, with emphasis on research direction, technology and results from theoretical proofs or simulations.

The important objective of this paper is to use two of the most important modes of interaction – head and hand to control any Computer Vision algorithms based application running on a computer. Video input stream hand is segmented. The corresponding gesture is recognizing based on the shape and pattern of movement of the hand. Hidden Markov model is used for the head gesture. Pre-processing common to hand and head gesture recognition 1st Capture a frame from camera.2nd: Hand and face are detected using Viola Jones algorithm. Classifiers are trained on the images of hand and face to detect them using an artificial neural network. Face detected can be used to figure out the area of the head. Method specific to Head gesture recognition: 1st: Optical flows of the all pixels, calculated with the gradient method, in the extracted head region are treated as values representing a head movement.2nd: The results of head movements are then used for recognition using finite state automata

The basic intention of static hand gesture recognition is to analyze the given hand gesture data represented by some features into some predefined finite number of gesture classes. The main interest of this effort is to explore the use of two feature extraction methods, especially, hand contour and complex moments to resolve the hand gesture recognition issue by identifying the main advantages and disadvantages of each method. Artificial neural network is built for the purpose of classification by using the back- propagation learning algorithm. The hand gesture image is passed through three stages, namely, pr-processing, feature extraction, and classification. In the sate of pr-processing, some operations are practiced to extract the hand gesture from its background and prepare the hand gesture image for the feature extraction stage. In the very initial method, the hand contour is used as a factor which treats the scaling and the translation issues. The complex moment's algorithm is however, used to outline the hand gesture and treat the rotation problem in addition to the scaling and translation. The results show that the hand contour

method has a performance of (71.30%) recognition, while complex moments have a better performance of (86.90%) recognition rate .

The gesture recognition system consists of five major steps. They are input image acquisition, pre-processing, feature extraction, classification of gesture and generation of suitable command for the control, sign language translation, in the rehabilitation device for people with upper extremity physical impairments. Vision based Gestural Controllable Human Computer Interaction system, to obtain robust and more accurate result, adopted various methods in the pre-processing stage and feature extraction stage. In the pre-processing stage skin color threshold method is used in background modeling for more accurate hand region segmentation. After this, in the feature extraction stages SIFT and MBC algorithm is used. Monogenic signal representation decomposes an original signal into three complementary components: orientation, amplitude, and phase. The monogenic variation in each local region and monogenic feature in each pixel are encoded, and then calculates the statistical features of the extracted local features. The proposed MBC scheme has significantly lower time and space complexity than other local feature extraction methods. The features extracted using SIFT are more prone to variation in scale and rotation.

2.2 VIRTUAL HAND RECOGNITION

Gesture Recognition system's is to create an interface that is natural for humans to operate or communicate with a computerized device. Objective of this work was to develop a control system for a robot freight ramp, based on gesture recognition. With that purpose, They decided to use a generic webcam for the image acquisition process, and they have defined a gesture vocabulary for the telerobotic control, using gesture recognition algorithm based on histograms and motion detection, which makes it suitable for real time control, easy to implement and efficient in unconstrained environments. This paper presents a fast, robust and accurate method for hand gestures recognition under unconstrained scenes.

Gesture recognition for gestures made from side angles are one of the most common problems encountered while using a single camera. Based on combination of Affine Transform and Discrete Fourier Transform (DFT), a hand gesture recognition system for multiple viewing angles using a single camera, is presented. Experimental result s 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 degree and +-45 degree respectively, which is considered good in the field of computer vision. The proposed system is made invariant to translation, rotation and scaling by the inclusion of the properties of DFT. The original shape of the sign isreconstructed by finding out Inverse Discrete Fourier Transform (IDFT). In this technique, affine transformation at particular angle makes the gestures input from various angles approximately similar to the gestures being made at 0 degree platform. Hence, the proposed method can be considered as an effective method for multi angle gesture detection. It is observed that the system is simple to use and cost effective also because only a single camera is used and from the experimental results, it can be said that the system performance is good enough and can be used .A vision based system to control various mouse activities such as left and right clicking using hand gestures to make the interaction more efficient and reliable is proposed. This paper delineates a vision based interface for regulating a computer mouse via 2D hand gestures. Hand gestures rely upon camera based color detection technique. This method mainly focuses on the use of a Web Camera to develop a virtual HCI device in a cost effective manner. Centroid of each input image is found. Hand movement also moves the centroid thus making it the principle of sensing for the alteration of cursor on computer screen. The hand image is treated here as the parent image. The left and right click functions of a mouse are implemented by folding the first and middle fingers of hand respectively and develop a baby image. So, by comparing the length of fingers in baby images with those in mother image gives an idea about the functionality performed by the hand gesture. When the length of finger crossovers the threshold length in baby image, it executes a clicking operation. Here, the efficiency of tracking the hand is improved by using red and blue colored caps on the fingers to make centroid looking more prominent. Post study, author believes that this technology has great future in HCI based systems. It can be widely used in the fields of robotics, biomedical instrumentation, computer gaming and many more.Gesture movement recognition system uses common USB camera and the method of skin color segmentation to get the image. The input image is converted from RGB space to YCbCr space with Otsu algorithm for threshold segmentation on the Cr channel of skin color clustering to get a better segmentation result. In the process of image preprocessing, filling small area algorithm is used to improve segmentation result by filling small outline

inside the gesture image with white, so as to solve the problem of the hole in the gesture image and improve the recognition accuracy of the system, which help realize the gesture recognition and mouse virtual control system. This system achieves using gestures instead of the mouse clicking, pointer movement and other virtual operations. Compared with the data glove and multifunction camera, the required equipment is simple and low cost, which is in favor of gesture recognition application and promotion. The development of human-computer interaction has been developed increasingly. The combination of gesture recognition and mouse control will make people's lives more convenient and intelligent; therefore this gesture movement recognition system has huge practicality and space for further development.

In this paper, the use of a camera and computer vision technology such as image segmentation and feature extraction, a technique is developed which can be used for computer control using hand gestures. The convex full and convexity defects algorithms perform in open CV platform using C++. From the obtained color image, the binary image is derived using image segmentation technique. Convexity defects are formed by the regions between the fingers. The number of convexity defects present in the hand gesture determines the count of the number of fingers present in the hand gesture. Point start describes Point of contour where the convexity defect begins.

Point End describes Point of contour where the convexity defect ends. Point Far describes Point within the defect which is farthest from the convex hull. Depth is distance between the convex This point as parameters for designing hand gestures for computer control. Further studies are going on in this topic to develop the applications using these parameters. In this paper, hand gesture recognition (HGR) algorithm capture the image of a single-handed gesture of hearing an impaired person using a simple webcam for an eccentric approach. An image capturing 12megapixel iball c12 webcam captured RGB color format and resized 160*120 to reduce the computational time. Image Preprocessing consists of hand segmentation by mapping In YCbCr color space into YCbCr color plane various possible ways of segmentation using different color spaces and models. The Region of Interest Extraction (ROI) uses the Sobel algorithm to find edges where the gradient of input binary image is maximum. A Gesture from 1 to 5 recognized correctly and zero, one, six, nine are count 1. The value of radius R varies with max, this method is invariant to size, distance from webcam and orientation of user's hand. In the experiments, we assume a

stationary background in order to get good segmentation results. The system is completely autonomous and easy to use as users.

In this paper, The Hand gesture recognition system focus on thresholding approach and skin color model along with an effective template matching using principal component analysis. Sensor base and Vision base are hand gestures recognition techniques. Sensors are attached to hand which record to get the position of the hand and collect data for gestures. Vision- based techniques use colored markers to get a position of data and various image processing algorithms. Image Acquisition, Hand Segmentation, Feature Extraction are the methodology for hand gesture system. Images of 4 gestures used with 5 different poses per gesture from 4 subjects making 20 images per gesture. The Skin color of different person can vary and background image can also contain the skin pixels so after skin color model Otsu thresholding is applied to remove the background are steps of hand segmentation. PCA method used for feature extraction and it reduces the dimensionality of the image while preserving information. The system is tested for 4 gestures with 5 different poses per gesture from 4 subjects making 20images per gesture and shows 91 .25% average accuracy and average recognition time of 0.098251 sec.

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, and drag, and soon. 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.

2.3 TYPES OF CONTROLS:

A) **Head Control**: A special sensor (or built-in webcam) can track head movement to move the mouse pointer around on the screen. In the absence of a mouse button, the software's dwell delay feature is usually used. Clicking can also be accomplished with a well-placed switch.

B) **Eye Control**: The cost of modern eye gaze systems is decreasing. These enable users

to move the pointer on the screen solely by moving their eyes. Instead of mouse buttons, a dwell delay feature, blinks, or a switch are used. The Tobii PCEye Go is a peripheral eye tracker that lets you use your eyes to control your computer as if you were using a mouse.

C) **Touch Screens**: Touch screens, which were once seen as a niche technology used primarily in special education schools, have now become mainstream. Following the success of smartphones and tablets, touch-enabled Windows laptops and all-in-one desktops are becoming more common. Although this is a welcome new technology, the widespread use of touch screens has resulted in a new set of touch accessibility issues. However, each of the methods below has its own set of disadvantages. The use of the head or eyes to control the cursor regularly can be hazardous to one's health. This can lead to a number of problems with health. When using a touch screen, the user must maintain their focus on the screen at all times, which can cause drowsiness. By comparing the following techniques, we hope to create a new project that will not harm the user's health

3.REQUIREMENTS ANALYSIS

3.1 INTRODUCTION

A software requirement specification(SRS)is a complete description of the behaviour of the software to be developed. It includes a set of use cases that describe all of the users will have with the software . In addition to use cases, the SRS contains functional requirements which define the internal working of the software : that is , the calculations, technical details, data manipulation and processing, and other specific functionality that shows how the use cases are to be satisfied. It also contains non functional requirements, which impose constraints on the design or implementation (such as performance requirements, quality standards or design constraints).

The SRS phase consists of two basic activities

1) **problem/requirement analysis**:

The process is order and more nebulous of the two deals with understanding the problem, the goals and constraint.

2) **Requirement Specification** :

Here the focus is on specifying what has been found giving analysis such as representation, specifications are addressed during this activity.

The requirement phase terminates with the production of the validate SRS document producing the SRS document is the basic goal of this phase.

Role of SRS:

The purpose of the software requirement specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium thought which the client and user needs are the accurately specified. It forms the basis of software development good SRS should satisfy all the parties involved in the system**.**

3.2 MODULAR DESCRIPTION &FUNCTIONAL REQUIREMENTS

The list of modules in corporate with"VIRTUAL MOUSE USING HAND GESTURE"are:

- Open cv
- Media pipe
- Comtypes
- Pycaw
- InTenum
- Screen_brightness_control

Open CV: Open cv(open source computer vision library) is an open source computer visi -on and machine learning software library. Open cv was built to provide a common infrast -ructure for computer vision applications and to accelerate the use of machine perception In the commercial products. Being a BSD-licensed producted. Open cv makes it easy for -businesses to utilize and modify the code.

Media pipe: Media Pipe is a Framework for building machine learning pipelines for processing time-series data like video, audio, etc. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices like Raspberry Piand Jetson Nano.

Com types: Ctypes is a foreign function library for Python. It provides C compatible datatypes, and allows calling functions in DLLs or shared libraries. It can be used to wrap these libraries in pure Python.

Pycaw: Python Core Audio Windows Library, working for both Python2 and Python3.

InTenum: Enumeration or Enum in C is a special kind of data type defined by the user. It consists of constant integrals or integers that are given names by a user. Enum is a class in python for creating enumerations, which are a set of symbolic names (members) bound to unique, constant values.

Screen_brightness_control: A Python tool for controlling the brightness of your monitor. Supports Windows and most flavours of Linux.

3.3 SOFTWARE REQUIREMENTS:

Developer Sides:

Operating system	: Windows 11
Application System	: PC
Programming Language	: Python
Development IDE	: PYTHON IDLE3.9
Gradle	: VScode Through Anaconda
Python SDK	: 4GB/equivalent above
Python JDK	: 4GB/equivalent above

Client Side:

Operating System	: Any
Web Browser	: Any

Server Side:

Operating System	: Any

Application Server : Any

3.4 HARDWARE REQUIREMENTS:

Developer Side:

Processor : Intel (R)core(TM)i3/equivalent

RAM :8 GB/equivalent

Disk Space : 512 SSD/equivalent

Client Side:

Processor : Intelp3/ equivalent

RAM :8GB/ equivalent

Disk Space :512 SSD/ equivalent

Processor :Server Environment Capable Hardware

RAM :8 GB/equivalent

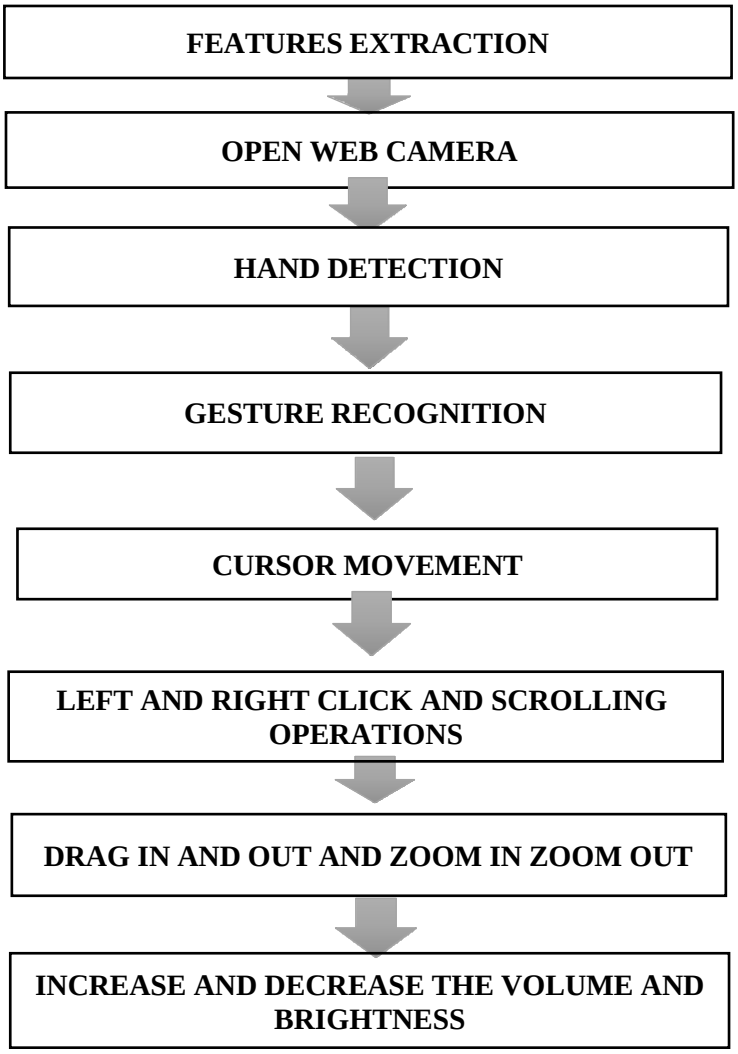
Disk Space :512 SSD/ equivalent

4.SYSTEM DESIGN

Systems design is the process of defining the architecture, product design, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as theapplication of systems theory to product development.

4.1 SYSTEM ARCHITECTURE

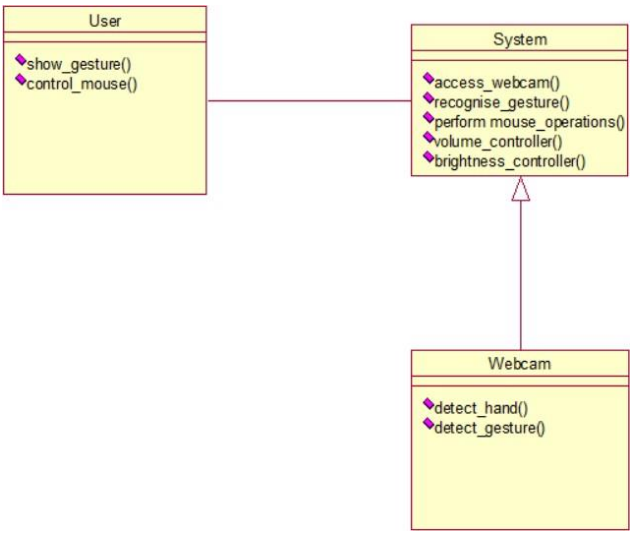
A system architecture is the conceptual model that defines the structure, behavior, and moreviews of a system. An architecture description is a formal description and representation of asystem, organized in a way that supports reasoning about the structures and behaviors of the system. There have been efforts to formalize languages to describe system architecture,collectively these are called architecture description languages(ADLs).



4.1 System Architecture

4.2 CLASS DIAGRAM:

Class Diagram consists of the classes and the object and the interaction between them. It mainly deals with interaction between classes in the system, their behaviour and properties of the system. Apart from classes this also provides inheritance relationship in the project. A class diagram consists of basically two parts: first one is the member variable and class variables and the second part consists of the total number of methods available in the class.

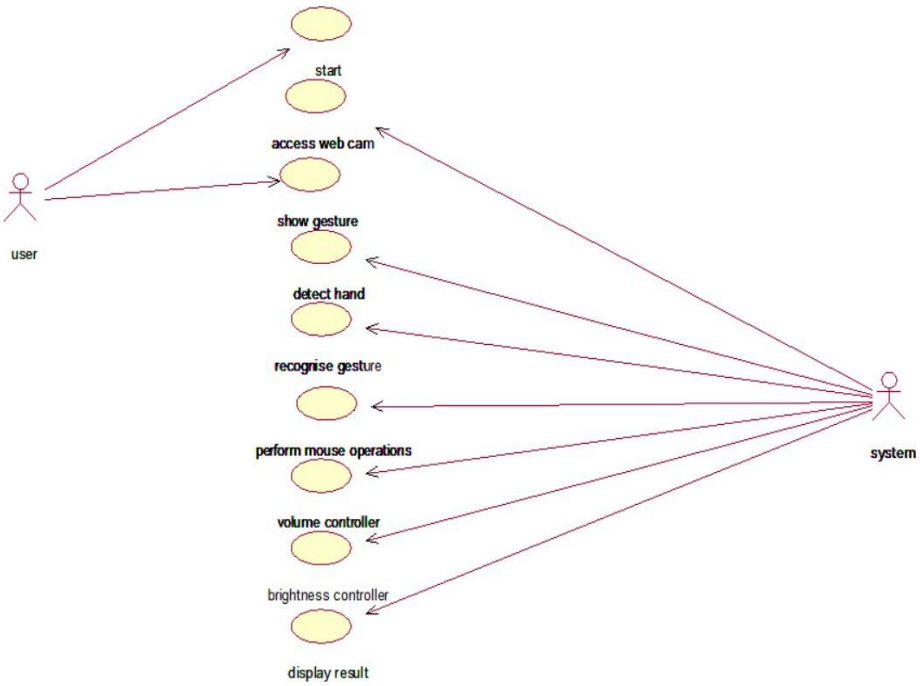


4.2 Class diagram for virtual mouse using hand gesture

4.3 USE CASE DIAGRAM :

Use case diagram consists of use cases and actors and show the interaction between them. The key points are

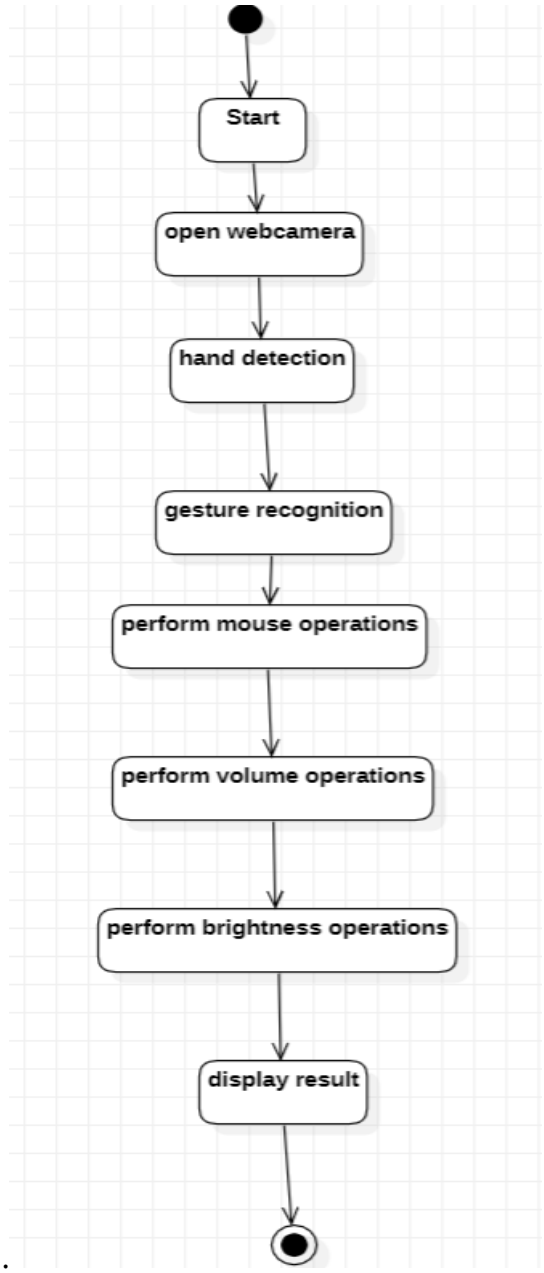
- The main purpose the system is to show the interaction between the usecase and the actor
- They represent the system requirement from user's perspective.
- The use cases are the functions that are to be performed in the module.
- An actor could be the end-user of the system or an external system



4.3 Usecase diagram for virtual mouse using hand gesture

4.4 ACTIVITY DIAGRAM:

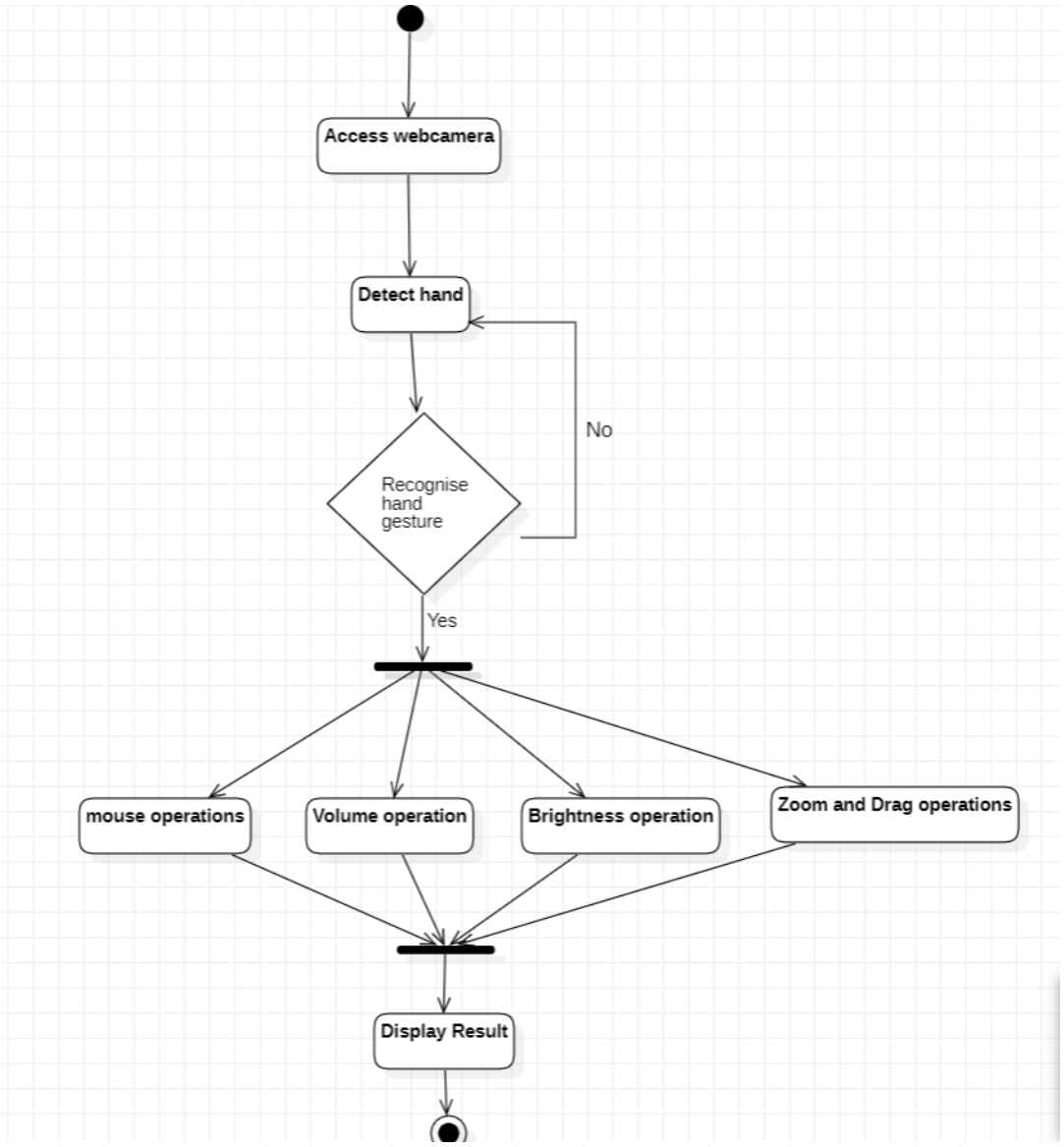
Activity diagrams are graphical representations of workflows off stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Languages, activity diagrams are intended to model both computational and Organizational processes.



4.4 Activity diagram for virtual mouse using hand gesture

4.5 STATE CHART DIAGRAM:

State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination



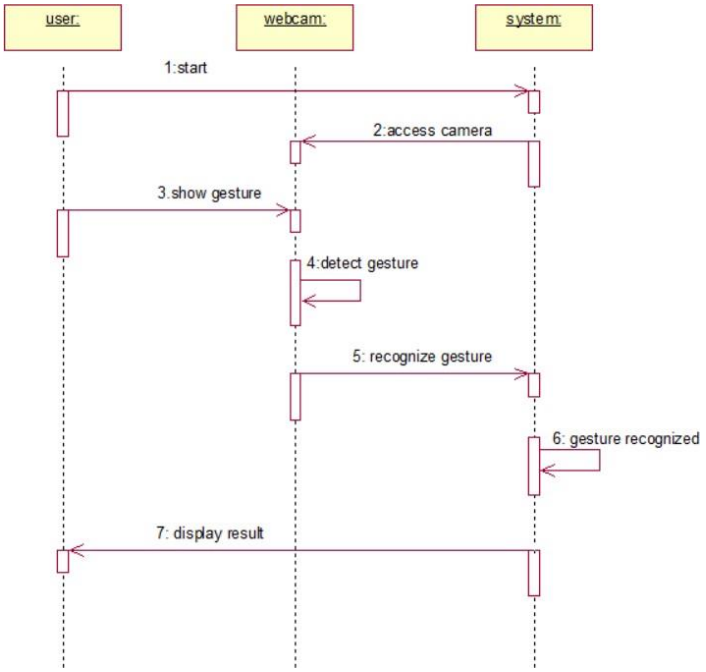
4.5 State chart diagram for virtual mouse using hand gestures

4.6 SEQUENCE DIAGRAM :

The purpose of sequence diagram is to show the flow of functionality through a use case. In other words, we call it a mapping process in terms of data transfers from the actor through thecorresponding objects.

The key points are:

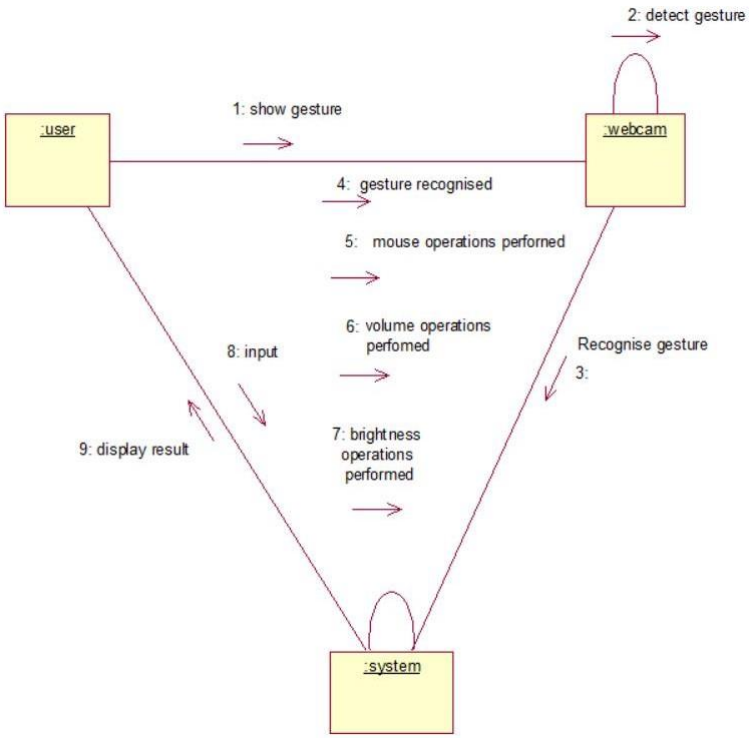
- The main purpose is to represent the logical flow of data with respect to a process.
- A sequence diagram displays the objects and not the classes.



4.6 Sequence diagram for virtual mouse using hand gesture

4.7 COLLABORATION DIAGRAM:

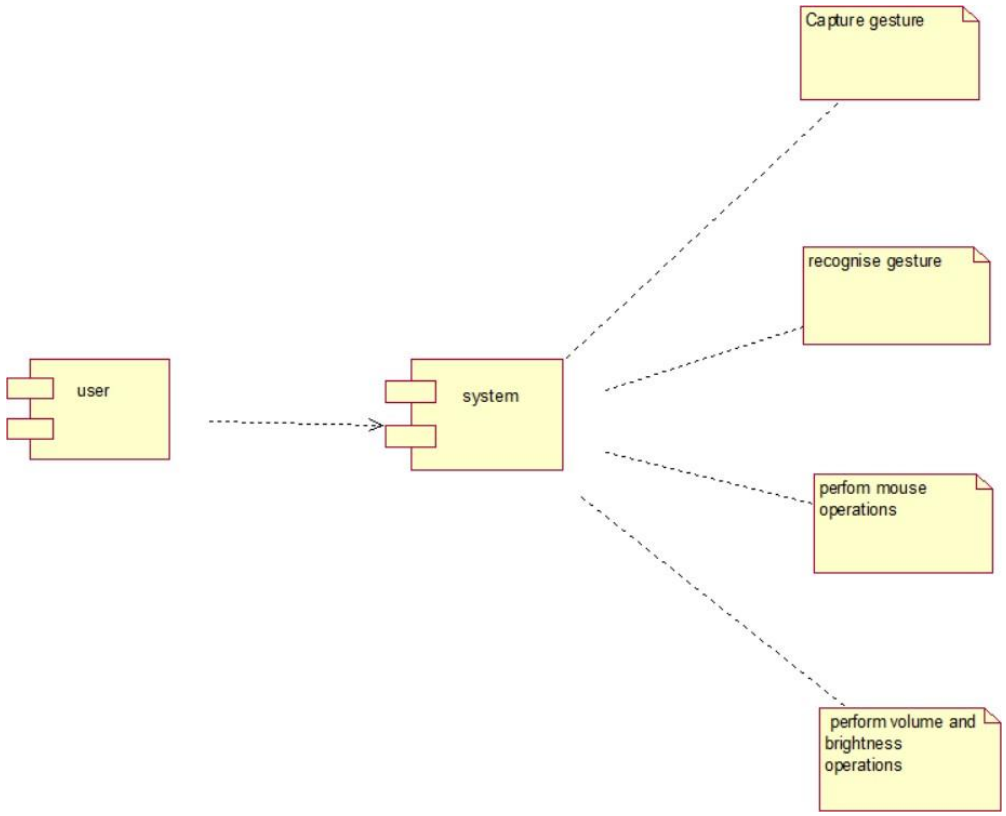
It consisting of a set of objects and their relationship. That emphasizes the time ordering of messages. It emphasizes the structural organization of the objects that's end and receive messages.



4.7 Collaboration diagram for virtual mouse using hand gestures

4.8 COMPONENT DIAGRAM:

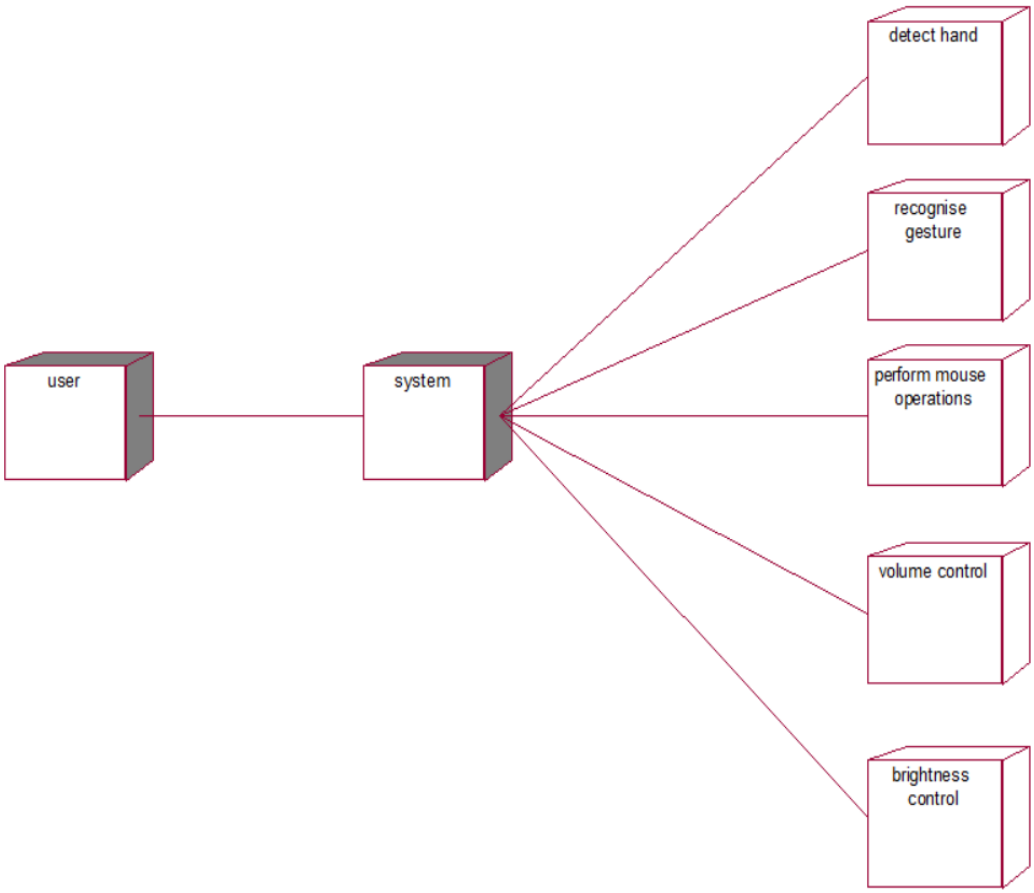
A component diagram is a type of UML diagram that shows the structural organization of software components and their relationships to each other. It provides a high-level view of the system and helps to identify the components, their interfaces, and the interactions among them.



4.8 component diagram for virtual mouse using hand gestures

4.9 DEPLOYMENT DIAGRAM :

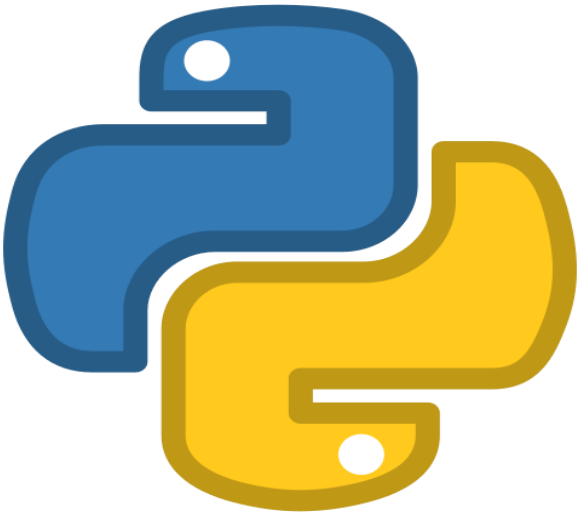
A deployment diagrams in the Unified Modelling language modules the physical deployment of artifacts on nodes.



4.9 Deployment diagram for virtual mouse using hand gesture

5.IMPLEMENTATION ISSUES

WHAT IS PYTHON?



2.1 Icon of Python diagram

5.1 PYTHON FRAMEWORK:

A framework is a collection of modules or packages which helps in writing web applications. Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics ,used for general-purpose programming. Python frameworks are used to reuse the code and to structure projects. The frameworks provide easiness while developing an application (or software) to developers. They offer automatic implementation of redundant tasks, reduce development time, and focus significantly on application logic rather than a common element.

Types of Python Frameworks:

There are primarily three types of frameworks as follows.

1.Full-Stack framework :

These frameworks have all web development requirements such as form generators, form validation, and template layouts. These are the core features of a typical full-stack Framework .It consists of some frameworks.

Django:

- Used for developing rich web application.
- It focuses on the Don’t Repeat Yourself Principle.

Giotto:

- It is based on Model View Controller
- It provides some essential characteristics such as automatic URL routin Eand Jinja2 for HTML templates with an API and accessible interface.

Pyramid:

- It aims to reduce the complexity
- It has flexible authentication and authorization.

2. Micro framework :

Micro frameworks require lots of code and additional requirements manually. This kind of framework doesn't provide the data abstraction layer's facility, form validation, and specific tools and libraries.

Bottle:

- It creates a single source of file for every application.It offers the request dispatching route

Dash:

- Used to develop analytical web applications
- There is extensive plug-in support

- It supports a high degree of customization

Falcon:

- Used for building web APIs
- It has a highly-optimized code base and also extensible

1. Asynchronous Framework

Asynchronous frameworks are gaining popularity that allows for handling a large set of concurrency connections. It uses the library to run the process concurrently.

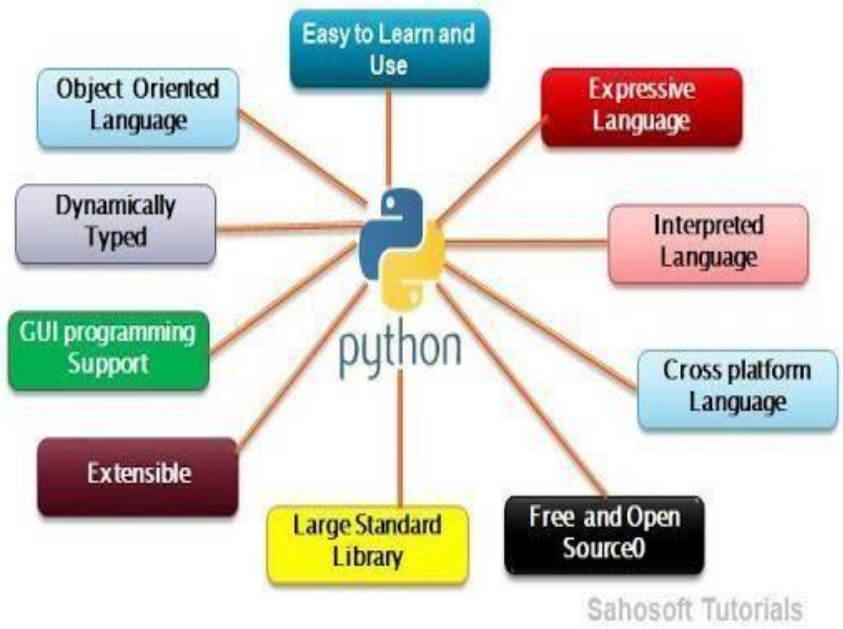
AIQHTTP:

- It is a pluggable routing
- It provides the facility to build the view effectively

Growler:

- Support for a multitude of open-source packages
- Use decorator for writing clean, reusable code
- It uses the decorators to reduce the complexity of code, and we can write the reusable code

5.2 FEATURES OF PYTHON:



5.2 Features of Python

1.Easy to code:

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Java script, Java, etc.

2.Object-Oriented Language:

Python supports object-oriented language and concepts of classes, objects encapsulation, etc.

3. Dynamically Typed Language:

Python is a dynamically-typed language. That means the type for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

4.GUI Programming Support:

Graphical User Interface is used for the developing Desktop application. PyQT5 is the library which is used for developing the web application.

5.Extensible feature:

Python is a Extensible language. We can write us some Python code into C or C++ language and also we can compile that code in C/C++ language.

6.Large Standard Library:

Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

1. **Free and Open Source:**

Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword.

2. **Cross-platform Language:** Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a cross-

platform language. It enables programmers to develop the software for several competing platforms by writing a program only once.

3. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time, like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called Byte code.

4. Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print ("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

PYTHON APPLICATIONS:

- Python supports cross-platform operating systems which makes building applications with it all the more convenient. Some of the globally known applications such as YouTube, BitTorrent, Drop Box, etc.

Web applications :

- Python can be used to make web-applications at a rapid rate

- A number of libraries that can help integrate protocols such as HTTPS, FTP, SSL etc. and even help in the processing of JSON, XML, E-Mail and so much more.
- It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautiful Soup, Feed parser, etc.
- Python provides many useful frameworks
- Django and Pyramid framework(Use for heavy applications)
- Flask and Bottle (Micro-framework)
- Prone and Django CMS (Advance Content management)

Game Development:

- Python is also used in the development of interactive games.
- There are libraries such as PySoy which is a 3D game engine supporting Python 3,
- Games such as Civilization-IV, Disney's Toontown Online, Vega Strike etc. have been built using Python.

Machine Learning and Artificial Intelligence

Machine Learning and Artificial Intelligence are the talks of the town as they yield the most promising careers for the future. you can design your own code which yields a better solution, which still is much easier when we compare it to other languages. Support for these domains with the libraries that exist already such as Pandas, Scikit-Learn, NumPy and so many more.

Data Science and Data Visualization

- Data is money if you know how to extract relevant information which can help you take calculated risks and increase profits.
- You can even visualize the data libraries such as Matplotlib, Seaborn, which are helpful in plotting graphs and much more

Desktop GUI

- We use Python to program desktop applications**.**
- It provides the Tkinter library that can be used to develop user interfaces.
- There are some other useful toolkits such as the wxWidgets,, Kivy, PYQT that can be used to create applications on several platforms.

5.3 WEB SCRAPING APPLICATIONS

Python is a savior when it comes to pull a large amount of data from websites which can then be helpful in various real-world processes such as price comparison, job listings, research and
development and much more.

Business Applications

- Business Applications are different than our normal applications covering domains such as e-commerce, ERP and many more.
- They require applications which are scalable, extensible and easily readable and Python provides us with all these features. Platforms such as Tryton is available to develop such business applications.

Audio and Video Applications :

We use Python to develop applications that can multi-task and also output media. Video and audio applications such as TimPlayer, Cplay have been developed using Python libraries

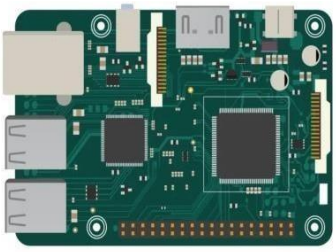
CAD Applications :

Computer-Aided Designing is quite challenging to make as many things have to be taken care of. Python makes this simple too and the most well-known application for CAD is Fandango.

Embedded Applications

Python is based on C which means that it can be used to create Embedded C software for embedded applications. This helps us to perform higher-level applications on smaller devices which can compute Python.

The most well-known embedded application could be the Raspberry Pi



5.4 HISTORY OF PYTHON :

- Python laid its foundation in the late 1980s. The implementation of Python was started in December 1989 by **Guido Van Rossum** at CWI in Netherland. In February 1991, **Guido Van Rossum** published the code (labeled version 0.9.0) to alt. sources

- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.
- Python 2.0 added new features such as list comprehensions, garbage collection systems.
- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.
- ABC programming language is said to be the predecessor of Python language, which was capable of Exception Handling and interfacing with the Amoeba Operating System.
- The following programming languages influence Python:
- ABC language.
- Modula-3

Why the name of the python ?

There is a fact behind choosing the name Python. **Guido van Rossum** was reading the script of a popular BBC comedy series "**Monty Python's Flying Circus**". It was late on-air 1970s.

- Van Rossum wanted to select a name which unique, sort, and little-bit mysterious. So he decided to select naming Python after the **"Monty Python's Flying Circus"** for their newly created programming language.

Python is also versatile and widely used in every technical field, such as Machine Learning, Artificial Intelligence, Web Development, Mobile Application, Desktop Application, Scientific Calculation, etc.

Old Python logo, 1990s–2006

New Python logo, 2006 present

5.5 VERSIONS:

A list of Python versions with its released date is given below.

PYTHON VERSION	RELEASED DATE
PYTHON 1.0	January 1994
PYTHON 1.5	December 31,1997
PYTHON 1.6	September 5, 2000
PYTHON 2.0	October 16, 2000
PYTHON 2.1	April 17,2001
PYTHON 2.2	December 21,2001
PYTHON 2.3	July 29,2003
PYTHON 2.4	November 30,2004
PYTHON 2.5	September 19, 2006

PYTHON 2.6	October 1,2008
PYTHON 2.7	July 3,2010
PYTHON 3.0	December 3,2008
PYTHON 3.1	June 27,2009
PYTHON 3.2	February 20,2011
PYTHON 3.3	September 29,2012
PYTHON 3.4	March 16,2004
PYTHON 3.6	December 23, 2016
PYTHON 3.7	June 27,2018
PYTHON 3.8	October 14,2019
PYTHON 3.9	October 5,2020
PYTHON 3.10	October 4,2021
PYTHON 3.11	December 6,2022

5.1 Versions of Python

5.6 PYTHON IDE'S

IDE stands for Integrated Development Environment is defined as a coding tool that helps to automate the process of editing, compiling, testing, etc. in an SDLC and it provides ease tothe developer to run, write and debug the code.

It is specially designed for software development that consists of several tools which is used for developing and testing the software.

There are some Python IDEs which are as follows:

- PyCharm
- Spyder
- Pydev
- Atom
- Wing
- Jupter notebook
- Thonny
- Rodeo
- Microsoft visual studio
- Eric python

5.3 Python IDEs

PyCharm

PyCharm was developed by the Jet Brains, and it is a cross-platform Integrated Development Environment (IDE) specially designed for python. It is the most widely used IDE andavailable in both paid version and free open-source as well. It saves ample timeby takingcare of routine tasks.

It is a complete python IDE that is loaded with a rich set of features like auto code completion, quick project navigation, fast error checking and correction, remote developmentsupport, database accessibility, etc.

Features:

- Smart code navigation
- Errors Highlighting
- Powerful debugger
- Supports Python web development frameworks, i.e., Angular JS, Java script

Spyder

Spyder is an open-source that has high recognition in the IDE market and most suitable for data science. The full name of Spyder is Scientific Python Development Environment. It supports all the significant platforms Linux, Windows, and MacOS X.

It provides a set of features like localized code editor, document viewer, variable explorer, integrated console, etc. and supports no. of scientific modules like NumPy, SciPy, etc.

Features:

- Proper syntax highlighting and auto code completion
- Integrates strongly with IPython console
- Performs well in multi-language editor and auto code completion mode

PyDev

PyDev is defined as one of the commonly used Python IDE, which is an external plugin for Eclipse. It is a natural choice of the Python developers that are coming from the Javabackground and very popular in the market as Python interpreter.

Aleksandar Totic is famous for his contribution to Mosaic browser and worked on Pydev project during 2003-2004.

Pydev has a feature which includes Django integration, automatic code completion, smartindents and block indents, etc.

Features:

- Strong Parameters like refactoring, debugging, code analysis, and code coveragefunction.
- It supports virtual environments, Mypy, and black formatter.
- Also supports PyLint integration, remote debugger, Unit test integration, etc.

Atom

Features:

- A plugin named "Markdown Preview Plus" provides built-in support for editing andvisualizing Markdown files.
- Visualize the results on Atom without open any other window.

Wing

It is defined as a cross-platform IDE that is packed with necessary features and with decent development support. Its personal edition is free of cost. The pro version comes with a 30 days trial for the developers to try it out.

It has several features that include auto-completion, syntax highlighting, indents, and debugging.

Features:

- Customizable and can have extensions as well.
- Supports remote development, test-driven development along with the unit test.

Jupyter Notebook

Jupyter is one of the most used IPython notebook editors that is used across the Data Scienceindustry. It is a web application that is based on the server-client structure and allows you to create and manipulate notebook documents. It makes the best use of the fact that python is aninterpreted language.

Features:

- Supports markdowns
- Easy creation and editing of codes
- Ideal for beginners in data science

Thonny

Thonny is another IDE which is best suited for learning and teaching programming. It is a software developed at the University of Tartu and supports code completion and highlight syntax errors.

Features:

- Simple debugger
- Supports highlighting errors and auto code completion.

Rodeo

Rodeo is defined as one of the best IDE for python that is most widely used for data science projects like taking data and information from different resources.

It supports cross-platform functionality and provides auto-completion of code.

Features:

- Allows the functions for comparing data, interact, plot, and inspect data.
- Auto code completion, syntax highlighter, visual file navigator, etc.

Microsoft Visual Studio

Microsoft Visual Studio is an open-source code editor which was best suited for developmentand debugging of latest web and cloud projects. It has its own marketplace for extensions.

Features:

- Supports Python Coding in Visual studio
- Available in both paid and free version

Eric Python

The Eric Python is an editor which is developed in Python itself and can be used for bothprofessional and non-professional work

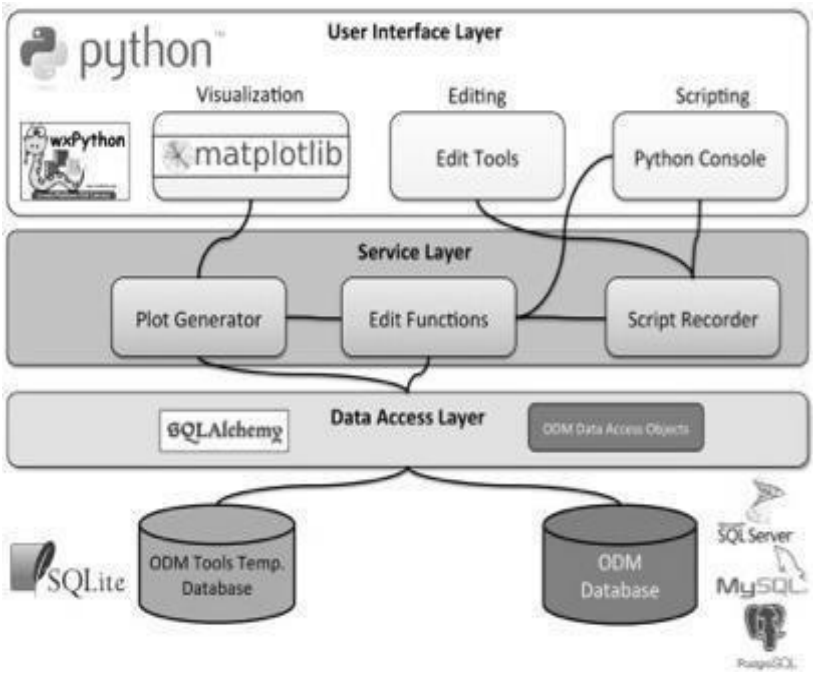
Features

- Offers configurable window layout, editors, source code folding
- Advanced project management capability, version control.

5.7 PYTHON ARCHITECTURE:

Python is an object-oriented programming language like Java. Python is called an interpreted language. Python uses code modules that are interchangeable instead of a single long list of instructions that was standard for functional programming languages. The standard implementation of python is called “cpython”. It is the default and widely used implementation of Python.

Python doesn’t convert its code into machine code, something that hardware can understand. It actually converts it into something called byte code. So within python, compilation happens, but it’s just not into a machine language. It is into byte code (.pyc or .pyo) and this byte code can’t be understood by the CPU. So we need an interpreter called the python virtual machine to execute the byte codes.



5.4 Python Architecture

The Python Programming language has been around for a long time. Guido van Rossum started development work on the first version in 1989, and it has since grown to become one of the more popular languages used in a wide range of applications from graphical interfaces to finance and data analysis.

This write-up looks at the nuts and bolts of the Python interpreter. It targets CPython, the

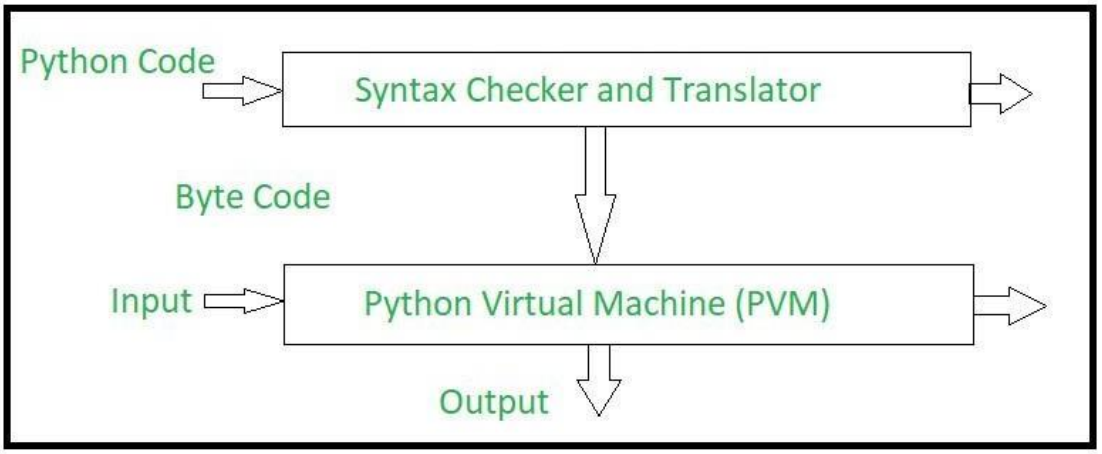
most popular, and reference implementation of Python at the point of this write-up. Python and CPython are used interchangeably in this text, but any mention of Python refers to CPython, the version of Python implemented in C. Other implementations include PyPy - Python implemented in a restricted subset of Python, Jython - Python implemented on the Java Virtual Machine, etc.

I regard the execution of a Python program as split into two or three main phases, as listed below. The relevant stages depend on how the interpreter is invoked, and this write-up covers them in different measures:

Initialization: This step covers the set up of the various data structures needed by the Python process and is only relevant when a program is executed non-interactively through the command prompt.

Compiling: This involves activities such as building syntax trees from source code, creating the abstract syntax tree, building the symbol tables, generating code objects etc.

Interpreting: This involves the execution of the generated code object's bytecode within some context.



The methods used in generating parse trees and syntax trees from source code are languageagnostic, so we do not spend much time on these. On the other hand, building symbol tables and code objects from the Abstract Syntax tree is the more exciting part of the compilation phase. This step is more Python-centric, and we pay particular attention to it. Topics we will cover include generating symbol tables, Python objects, frame objects, code objects, function objects etc. We will also look at how code objects are interpreted

and the data structures that support this process.

This material is for anyone interested in gaining insight into how the CPython interpreter functions. The assumption is that the reader is already familiar with Python and understands the fundamentals of the language. As part of this exposition, we go through a considerable amount of C code, so a reader with a rudimentary understanding of C will find it easier to follow. All that is needed to get through this material is a healthy desire to learn about the CPython virtual machine.

This work is an expanded version of personal notes taken while investigating the inner working of the Python interpreter.

At the end of this write-up, a reader should understand the processes and data structures that are crucial to the execution of a Python program. We start next with an overview of the execution of a script passed as a command-line argument to the interpreter.

5.8 FRAGMENTS:

A code to study the fragmentation of filaments (FragMent) FragMent is a python/C module which collates a number techniques used to study fragmentation in filaments. It also performs model selection using a frequentist and Bayesian approach to find the best descriptor of a filament's fragmentation. The code is open-source and can be downloaded here.

The accompanying paper can be found here. It details the sensitivities of each method and explains in more detail a number of the procedures one should use when analysing fragmentation.

While the code was designed to investigate filament fragmentation the functions are general and may be used for any set of 2D points to study more general cases of fragmentation.

Dependencies FragMent requires 4 common libraries to be installed:

Numpy , Scipy, ctypes, emcee.

To allow the import of the FragMent module from any directory use the export command to modified the PYTHONPATH variable.

This is done by adding the line export PYTHONPATH=$PYTHONPATH:"Path to FragMent's download location" to the .bashrc file in the home directory. To compile the C portion of the module use the command

For Linux:

gcc -shared -Wl,-soname,FragMent_C -o FragMent_C.so -fPICFragMent_C.c -g

For Mac:

cc -shared -Wl,-install_name,FragMent_C -o FragMent_C.so -Fpic FragMent_C.c -g Fragments uses concepts from version control to replace many uses of templating languages. Instead of a templating language, it provides diff-based templating; instead of revision control, it provides "fragmentation control".

Fragments is a violation manager; it is engine.

What is diff-based templating?

Generating HTML with templating languages is difficult because templating languages often have two semi-incompatible purposes. The first purpose is managing common HTML elements & structure: headers, sidebars, and footers; across multiple templates. This is sometimes called page "inheritance". The second purpose is to perform idiosyncratic display logic on data coming from another source. When these two purposes can be separated, templates can be much simpler. Fragments manages this first purpose, common HTML elements and structure, with diff and merge algorithms. The actual display logic is left to your application, or to a templating language whose templates are themselves managed by Fragments.

What is fragmentation control?

The machinery to manage common and different code fragments across multiple versions of _a single file_ already exists in modern version control systems. Fragments adapts these tools to manage common and different versions of _several different files_. Each file is in effect its own "branch", and whenever you modify a file ("branch") you can apply ("merge") that change into whichever other files ("branches") you choose. In this sense Fragments is a different kind of "source control"--rather than controlling versions/revisions over time, it controls fragments across many files that all exist simultaneously. Hence the term "fragmentation_control". As I am a linguist, I have to point out that the distinction between

[Synchronic](http://en.wikipedia.org/wiki/Synchronic_analysis)

[Diachronic](http://en.wikipedia.org/wiki/Diachronics) Linguistics gave me this idea in the first_place.

How does it work?

The merge algorithm is a version of [Precise CodevilleMerge] modified to support cherrypicking. Precise Codeville Merge was chosen because it supports [accidental clean merges] and [convergence](http://revctrl.org/Convergence).

That is, if two files are independently modified in the same way, they merge together cleanly. This makes adding new files easy; use Fragment's `fork` command to create a new file based on other files (or just `cp` one of your files), change it as desired, and commit it.

Subsequent changes to any un-modified, common sections, in that file or in its siblings, will be applicable across the rest of the repository.

Like version control, you run Fragments on the command line each time you make a change to your HTML, not before each page render.

What is it good for?

Fragments was designed with the task of simplifying large collections of HTML or HTML templates. It could replace simpler CMS-managed websites with pure static HTML. It could also handle several different translations of an HTML website, ensuring that the same HTML structure was wrapped around each translation of the content. But Fragments is also not .

HTML specific. If it's got newlines, Fragments can manage it. That means XML, CSS, JSON, YAML, or source code from any programming language where newlines are common (sorry, Perl). cFragments is even smart enough to know not to merge totally different files together. You could use it to manage a large set of configuration files for different servers and deployment configurations, for example. Or you could use it to manage bug fixes to that mess of duplicated source files on that legacy project you wish you didn't have to maintain.In short, Fragments can be used anyplace where you have thought to yourself "this group of files really is violating DRY".

Integration with version control:

Fragments has no history; It only stores the previous committed state of a file.

Storing history is up to your version control system.

But Fragments stores its repository configuration in such a way to allow your version control system to manage it painlessly and obviously.

Configuration is stored in a `_fragments` directory.

This directory name is not preceded by a `.`, and all the files in it are stored as plain text. You want to make sure to run `fragments init` from your repository root, which is usually at least one directory level above your actual content.

Otherwise your `_fragments` directory may get accidentally deployed to production or interfere with template loaders.

The `rename` and `forget` commands in Fragments are written to not interfere with a version control's rename and remove commands, as these commands sometimes need to be used in tandem.

Invisibility

Fragments is invisible to people who don't know it's being used.

If you (or someone else) make more than one change to a file, Fragments' `apply` command allows you to perform chunk-based interactive application of changes, similar to `git commit --patch` or `hg record`.

So, you can give a single HTML file to your web designer or junior programmer, let him or her modify it as desired. Later, you can selectively apply some of those changes across all other HTML files, while leaving other changes only in the modified file.

5.9 PYTHON UI LAYOUTS

Different Layout Managers in Python's TKInter GUI package

1. Pack Layout Manager
2. Grid Layout Manager
3. Place Layout Manager

1.Pack Layout Manager:

pack is one of the oldest and most used layout managers in Python's TKInter package. It is very easy to style and place the widgets in an app using this pack() manage r. When you use this pack() method on a widget, you don't need to explicitly specify the position of that widget, pack automatically places the widget in window based upon the space available in the window. You can use pack when your layout only consists of a group of items all aligned horizontally or vertically, mostly in case of a navigation menu or something like that. This pack() has 3 options to use they are: fill, expand, and side. I will create a simple example to demonstrate this pack manager.

When you use this pack() method on a widget, you don't need to explicitly specify the position of that widget, pack automatically places the widget in window based upon the space available in the window.

I will use fill option to align three labels vertically.

```
# fill option
label1 = Label(root, text="Label 1", bg="#E74C3C", fg="white").pack(fill=X, padx=10)
label2 = Label(root, text="Label 2", bg="#2ECC71", fg="black").pack(fill=X, padx=10)
label3 = Label(root, text="Label 3", bg="#F1C40F", fg="white").pack(fill=X, padx=10)
```

Use side option to align them horizontally.

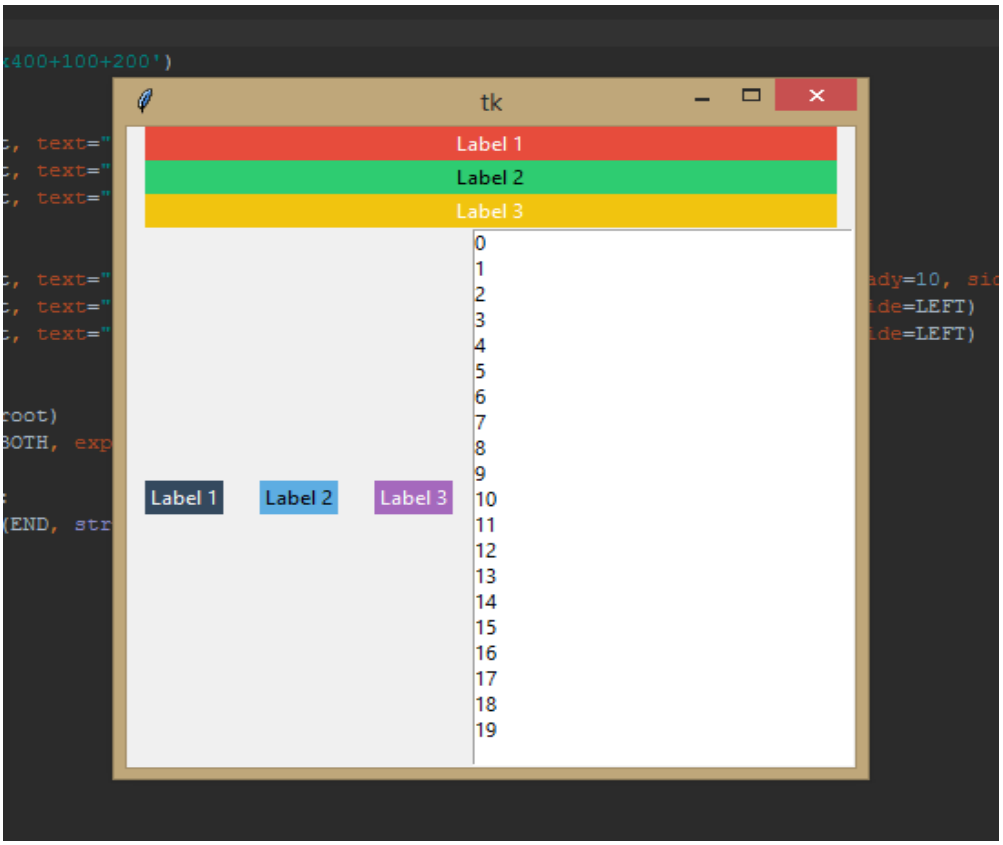
```
# side option
label4 = Label(root, text="Label 1", bg="#34495E", fg="white").pack(fill=X, padx=10, pady=10, side=LEFT)
label5 = Label(root, text="Label 2", bg="#5DADE2", fg="black").pack(fill=X, padx=10, side=LEFT)
label6 = Label(root, text="Label 3", bg="#A569BD", fg="white").pack(fill=X, padx=10, side=LEFT)
```

Expand operation to take up the full height until the content ends. We use listbox widget to fillup space.

```
# expand optionlistbox=Listbox(root)listbox.pack(fill=BOTH, expand=1) for I range(20):listbox.insert(END, str(i))
```

Pack Manager Example:

```
__author_____ = 'Avinash'from tkinter import *root = Tk()
```



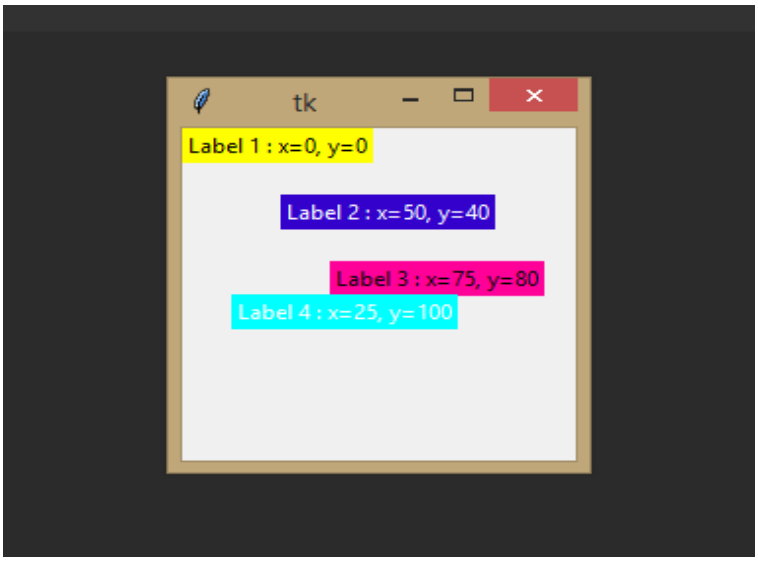
5.5 Pack Layout Manager

Grid Layout Manager:

Grid is one of the most flexible layout manager out of the three GUI layout managers in python. It was introduced as an alternative to pack. Grid allows you to position the elements inrows and columns which give you more flexibility to your widgets.

I will create a simple 2×2 table like structure using grid's rows and column way.

__author_____='Avinash'from tkinter import *Label(text="Label 1", width=10).grid(row=0,column=0)
Label(text="Label 2", width=10).grid(row=0,column=1)Label(text="Label3"
: x=75, y=80", bg="#FF0099", fg="black").place(x=75, y=80) Label(root, text="Label 4 :x=25, y=100", bg="#00FFFF", fg="white").place(x=25, y=100)mainloop()



5.6 Grid Layout Manager

5.10 OPEN COMPUTER VISION

Computer vision

Computer vison can be defined as a discipline that explains how to reconstruct, interrupt And understand the 3D scene from its 2D images, in terms of the properties of the structure present in the scene it deals with modeling and replicating human vision computer software and hardware.

Computer Vision overlaps significantly with the following fields –

Image Processing – It focuses on image manipulation.

Pattern Recognition – It explains various techniques to classify patterns.

Photogrammetry – It is concerned with obtaining accurate measurements from image.

Applications of Computer Vision:

Here we have listed down some of major domains where Computer Vision is heavily used.

Robotics Application

- Localization -Determine robot location automatically
- Navigation
- Obstacles avoidance
- Assembly (peg-in-hole, welding, painting)
- Manipulation (e.g. PUMA robot manipulator)
- Human Robot Interaction (HRI)-Intelligent robotics to interact with and serve people.

Medicine Application

- Classification and detection (e.g. lesion or cells classifications and tumor detection)
- 2D/3D segmentation
- 3D human organ reconstruction (MRI or ultrasound)
- Vision-guided robotics surgery

Industrial Automation Application

- Industrial inspection (defect detection)
- Assembly
- Barcode and package label reading
- Object sorting
- Document understanding (e.g. OCR)

Security Applications

- Biometrics (iris, finger print, face recognition)
- Surveillance-Detecting certain suspicious activities or behaviors

Transportation Application

- Autonomous vehicles
- Safety

Features of OpenCV Library

Using OpenCV library, you can-

- Read and Write Images
- Capture and save videos
- Process images
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or Images
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV Library Modules

Following are the main library modules of the OpenCV Library

Core Functionality

This module covers the basic data structures such as scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array

Mat, which is used to store the images. In the Java Library of OpenCV, this module is included as package with the name org. opencv.core.

Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module included as a package with the name org.opencv.imgproc.

Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the library of OpenCV, this module is included as a package with the name org.opencv.video.

Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the java library of OpenCV, this module is included as a package with the name org.opencv.videoio.

Calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name org.opencv.calib3d.

Feature2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name org.opencv.feature2d.

Objdetect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.objdetect.

Highgu This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, this module is included in two different packages namely, org.opencV.

5.11 K-NN ALGORITHM

The KNN algorithm is a popular classification algorithm used in machine learning. In the context of a virtual mouse using hand gestures, KNN can be used to classify different hand gestures and map them to specific mouse actions such as left-click, right-click, and scroll.The working process of the KNN algorithm for virtual mouse using hand gesture can be explained as follows:

Data Collection: Collect hand gesture data using a camera or any other device capable of detecting hand gestures. Each gesture should be associated with a specific mouse action, such as left-click, right-click, or scroll.

Feature Extraction: Extract meaningful features from the collected data. These features could include hand position, hand shape, finger position, and motion trajectory.

Data Preprocessing: Normalize and preprocess the extracted features to eliminate any noise or outliers that may affect the accuracy of the KNN model.

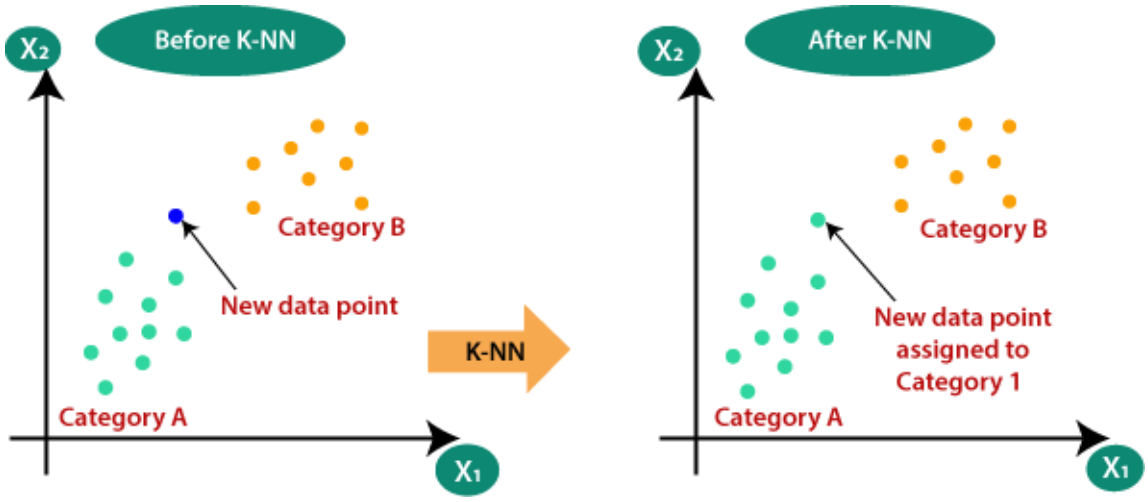
Training: Use the preprocessed data to train the KNN model. During training, the KNN algorithm will use the labeled data to identify patterns in the features that correspond to specific mouse actions.

Testing: Test the KNN model by providing it with new, unlabeled hand gesture data. The KNN algorithm will use the patterns it identified during training to classify the new data and determine the corresponding mouse action.

Mapping: Map the classified hand gestures to specific mouse actions, such as left-click, right-click, or scroll, and use them to control the virtual mouse.

Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

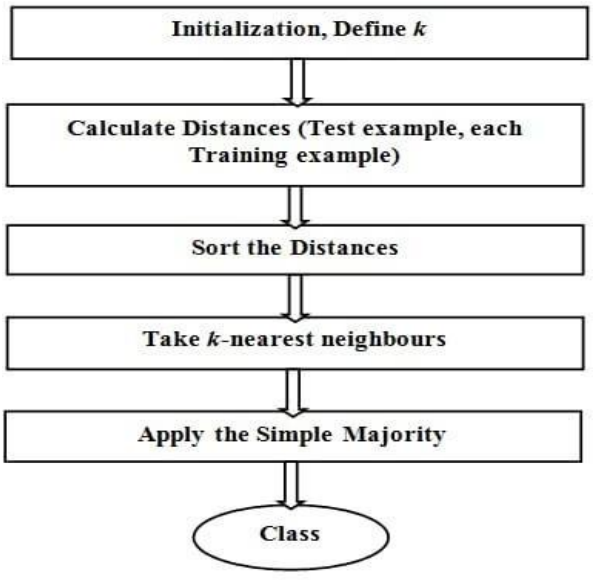
Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our Model is ready.



5.7 K-NN Working

5.12 PROPOSED ALGORITHM:

The goal of the algorithm is to detect gestures with real time processing speed ,minimize interference , and reduce the ebility to capture unintentional gestures

Working process of proposed algorithm:

Here is a proposed algorithm for a virtual mouse using hand gestures:

Step 1: Detect Hand Gestures

The first step is to detect hand gestures using a camera or sensor. There are different methods to detect hand gestures, but the most common approach is to use computer vision algorithms such as the Haar Cascade Classifier, which can detect the position and orientation of the hand.

Step 2: Process Image

Once the hand gesture is detected, the next step is to process the image and extract relevant features such as the position of the hand, the direction of the movement, and the distance between the hand and the camera.

Step 3: Convert to Mouse Movement

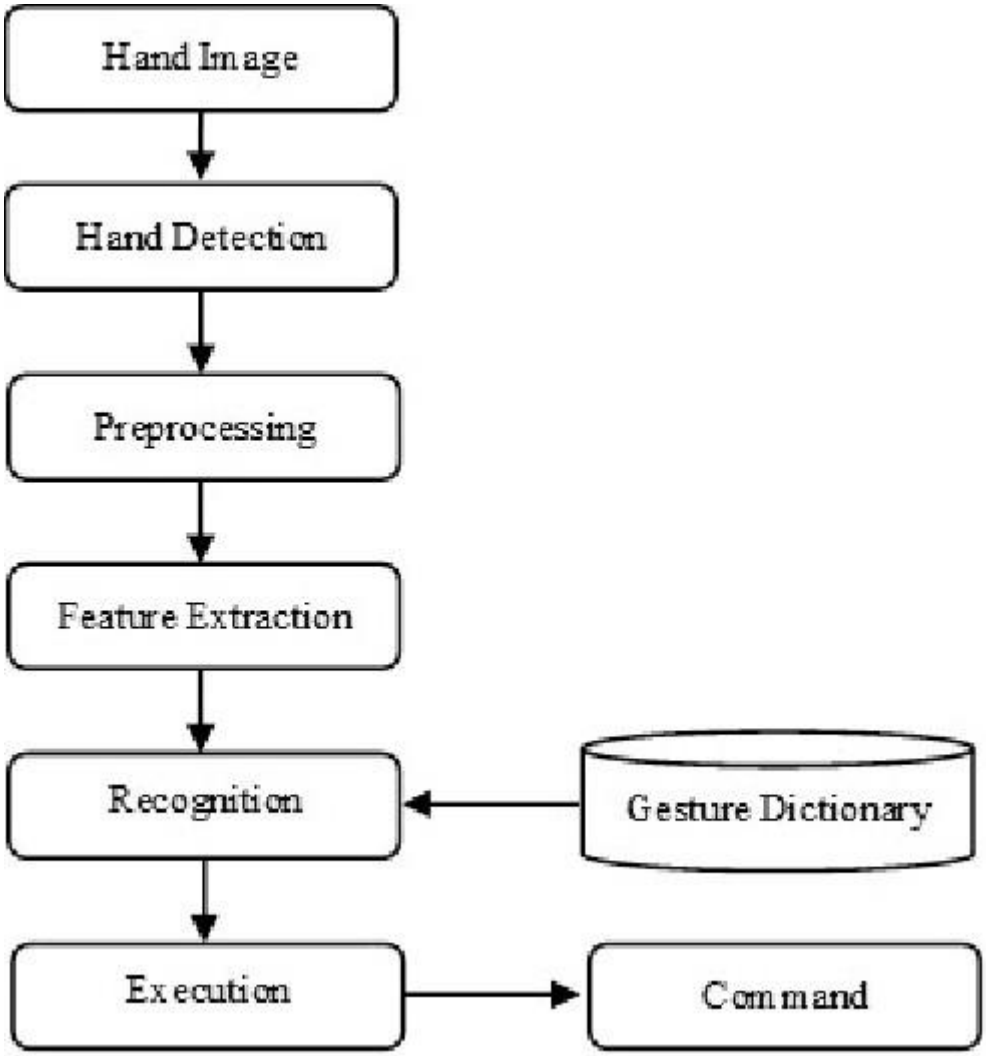
Using the processed image, we can convert the hand gestures into mouse movement. For example, if the user moves their hand to the left, the cursor on the screen should move to the left. The direction and speed of the mouse movement can be adjusted based on the distance between the hand and the camera and the speed of the gesture.

Step 4: Click and Drag

In addition to moving the cursor, the virtual mouse should also be able to perform left and right clicks, as well as drag and drop operations. For example, to perform a left-click, the user can make a fist gesture, and to perform a right-click, they can use a specific finger gesture.

Step 5: Calibration

Before using the virtual mouse, the system should be calibrated to the user's hand size and gesture movements. This calibration process can involve asking the user to perform specific gestures and adjusting the system's settings accordingly.



5.8 Proposed Algorithm

5.13 TRANSITIONAL ALGORITHM:

A transitional algorithm is a set of steps that describes how to transition from one state or condition to another. This type of algorithm is commonly used in computer programming, where it is necessary to move data from one format or structure to another**.**

Working process:

A transitional algorithm is a set of instructions or steps that describe how to transform or transition data from one state or form to another. The algorithm is designed to be reliable, efficient, and accurate, and it can be used for a variety of purposes such as data conversion, data migration, and data integration.

The working process of a transitional algorithm involves several key steps, including:

Input Data: The algorithm takes input data in a specific format or structure, which can be in the form of a file, database, or other data source.

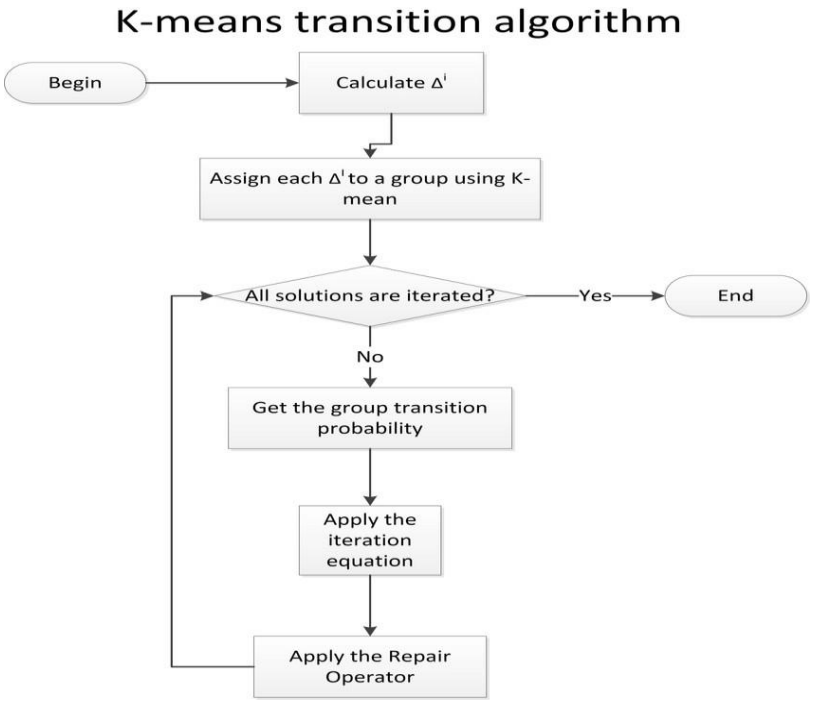
Data Analysis: The algorithm analyzes the input data to understand its current state, such as its format, structure, and content.

Transformation Rules: The algorithm defines transformation rules that dictate how the input data will be transformed to the desired output state. The rules can be defined based on the data analysis or specified beforehand.

Data Transformation: The algorithm applies the transformation rules to the input data to create the desired output. This may involve converting data formats, changing data structures, or performing data manipulation.

Data Validation: The algorithm validates the transformed data to ensure it meets the desired output state's requirements. This may involve data quality checks, data completeness checks, or other validation criteria.

Output Data: The algorithm outputs the transformed data in the desired format orstructure, which can be in the form of a file, database, or other data destination.



5.9 Transitional algorithm

6.TESTING

Testing is a process, which reveals errors in the program. It is the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected.In order to make sure that the system does most have errors, the different levels of testing strategies that are applied at differing phases of software development are:

Level of testing:

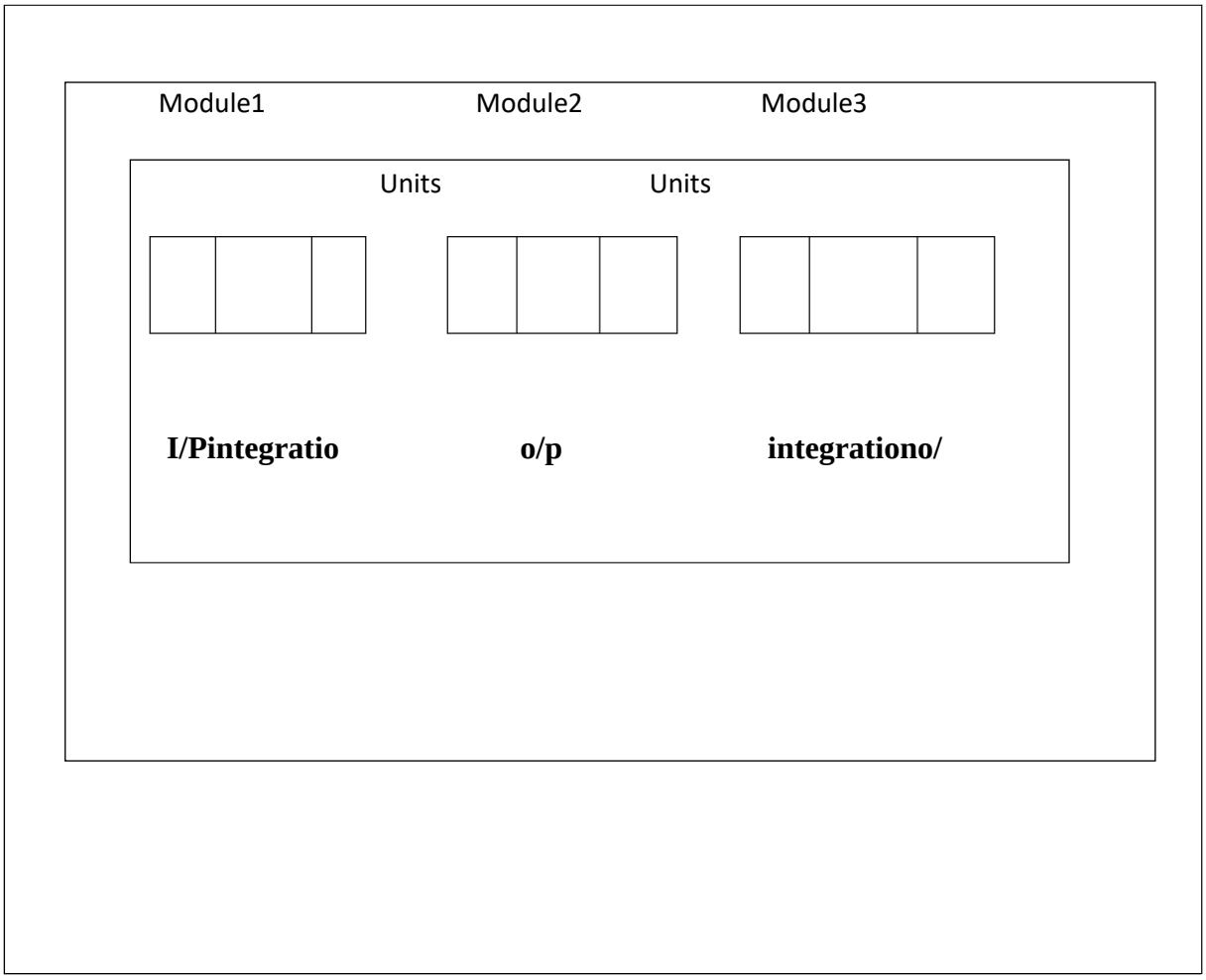


Fig.6.1. Level of testing

6.1 TYPES OF TESTING:

Unit Testing

Unit Testing is done individual modules as they are completed as they and become executable

It is confined only to the designer's requirements.

Each module can be tested using the following two strategies:

Black Box Testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been used to find errors in the following categories:

a) Incorrect or missing functions
b) Interface errors
c) Errors in data structure or external database access
d) Performances errors
e) Initialization and termination errors.

In this testing only that output is checked for correctness.

The logical of the data is not checked.

White Box testing

In this the cases are generated on the logical of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

It has been uses to generate the test cases in the following cases:

a) Guarantee that all independent paths have executed
b) Execute all logical decisions on their true and false sides.
c) Execute all loops at their boundaries and within their operational bounds.
d) Execute internal data structures to ensure their validity.

Integrating Testing

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

Acceptance testing

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

Validation

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled . In case of erroneous input corresponding error messages are displayed.

Compilation Test:

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected to very high transaction volumes.

Execution Test:

This program was successfully loaded and executed. Because of good programming there was no execution error.

Output Test:

The successfully output screens are placed in the output screens section above.

6.2 TESTCASES

SL.NO	TEST CASE NAME	TEST PROCEDURE	EXPECTED OUTPUT	RESULT
1	Using in Bright and normal lighting environment camera capturing hand	Making the system work in Bright light	Invoking the camera and check whether we are able to capture the Hand and its gesture in the normal light.	Pass
2	Detecting Both Left and Right hands	By placing Both hands facing the Camera	Camera detecting Both The Hands	Pass
3	Cursor movement	By using index and middle fingures with specified “V” gesture	Movement of cursor Occurs	Pass
4	Left and Right click operations	Pressing down with the index finger will perform a left click operation and pressing down with the middle finger will perform a right click operation	performing Left and Right click Operations	Pass
5	Scrolling operations	By using Left hand thumb and index fingure moves up and down with specified “SUPER” symbol gesture	Performing scrolling operations	Pass
6	Dragging operations	By using right fist to performing drag and drop operations	Drag operations performed	Pass

7	Zoom in and Zoom out operations	By using left hand thumb and index fingures moves left to left it perform Zoom in , right to right it perform Zoom out	Zoom in and zoom out operations performed	Pass
8	Volume control	By using Right hand thumb and index fingures moves vertically up and down	Performing volume operations	Pass
9	Brightness control	By using Right hand thumb and index fingures moves horizontally right to left and right to right.	Performing brightness operations	Pass

6.1 Test Cases

7.SAMPLE CODE

7.1 SOURCE 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

# Gesture Encodings
class Gest(IntEnum):
    # Binary Encoded
    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

    # Multi-handedness Labels
    class HLabel(IntEnum):
        MINOR = 0
        MAJOR = 1
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)

    # Function to find Gesture Encoding using current finger_state.
    # Finger_state: 1 if finger is open, else 0
    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(dist1/0.01,1)

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


      # Handling Fluctations due to noise
      def get_gesture(self):
          if self.hand_result == None:
              return Gest.PALM

          current_gesture = Gest.PALM
```

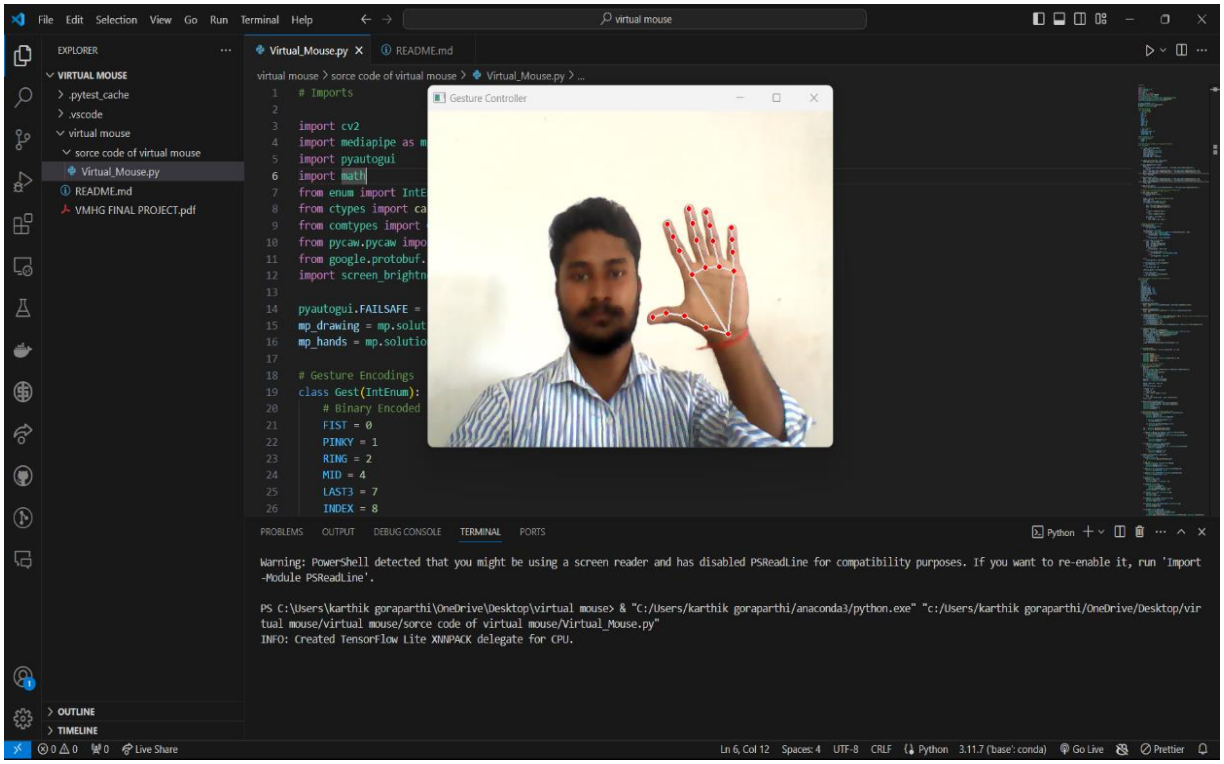
```
            if self.finger in [Gest.LAST3,Gest.LAST4] and self.get_dist([8,4]) <
0.05:
                if self.hand_label == HLabel.MINOR :
                    current_gesture = Gest.PINCH_MINOR
                else:
                    current_gesture = Gest.PINCH_MAJOR

            elif Gest.FIRST2 == self.finger :
                point = [[8,12],[5,9]]
                dist1 = self.get_dist(point[0])
                dist2 = self.get_dist(point[1])
                ratio = dist1/dist2
                if ratio > 1.7:
                    current_gesture = Gest.V_GEST
                else:
                    if self.get_dz([8,12]) < 0.1:
                current_gesture =  Gest.TWO_FINGER_CLOSED
                    else:
                        current_gesture =  Gest.MID

            else:
                if current_gesture == self.prev_gesture:
                self.frame_count += 1
                else:
                self.frame_count = 0
                self.prev_gesture = current_gesture
                if self.frame_count > 4 :
                self.ori_gesture = current_gesture
# Executes commands according to detected gestures
                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
    currentBrightnessLv = 1.0
    elif currentBrightnessLv < 0.0:
    currentBrightnessLv = 0.0
def changesystemvolume():
    devices = AudioUtilities.GetSpeakers()
    interfacedevices.Activate(IAudioEndpointVolume._iid_,CLSCTX_ALL,None)
    volume = cast(interface, POINTER(IAudioEndpointVolume))
    currentVolumeLv = volume.GetMasterVolumeLevelScalar()
    currentVolumeLv += Controller.pinchlv/50.0
    if currentVolumeLv > 1.0:
```

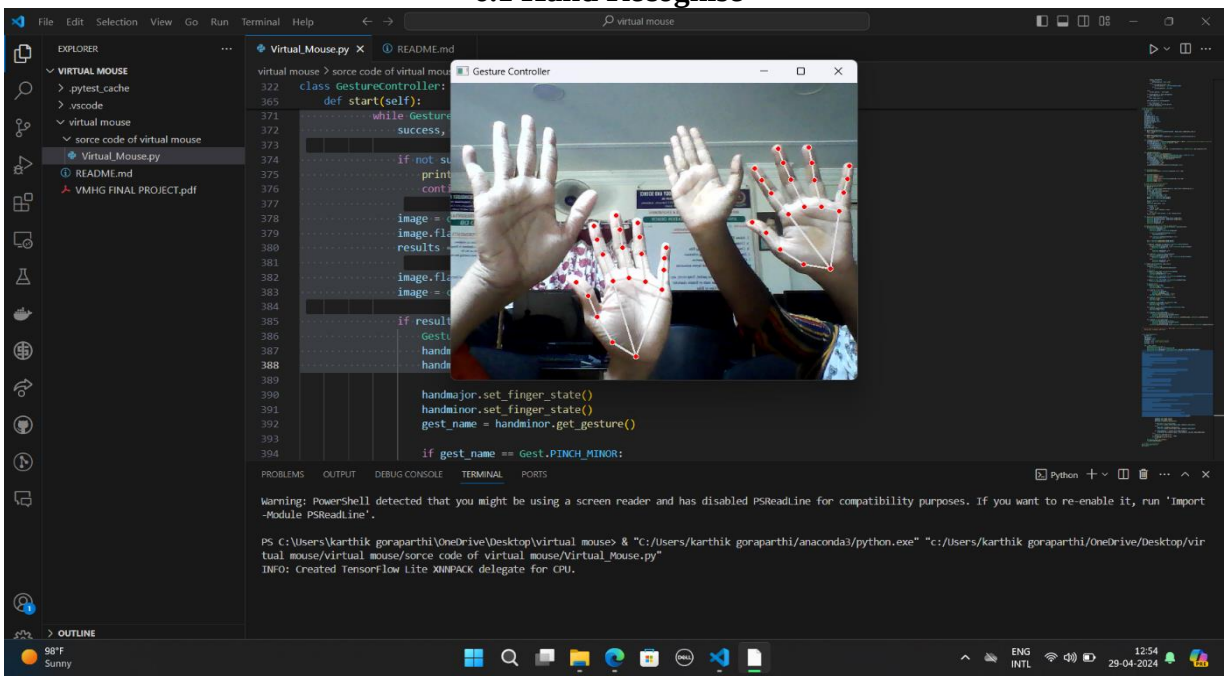
```
    elif currentVolumeLv < 0.0:
    volume.SetMasterVolumeLevelScalar(currentVolumeLv, None)
def scrollVertical():
    pyautogui.scroll(120 if Controller.pinchlv>0.0 else -120)


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
distsq = delta_x**2 + delta_y**2

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
```

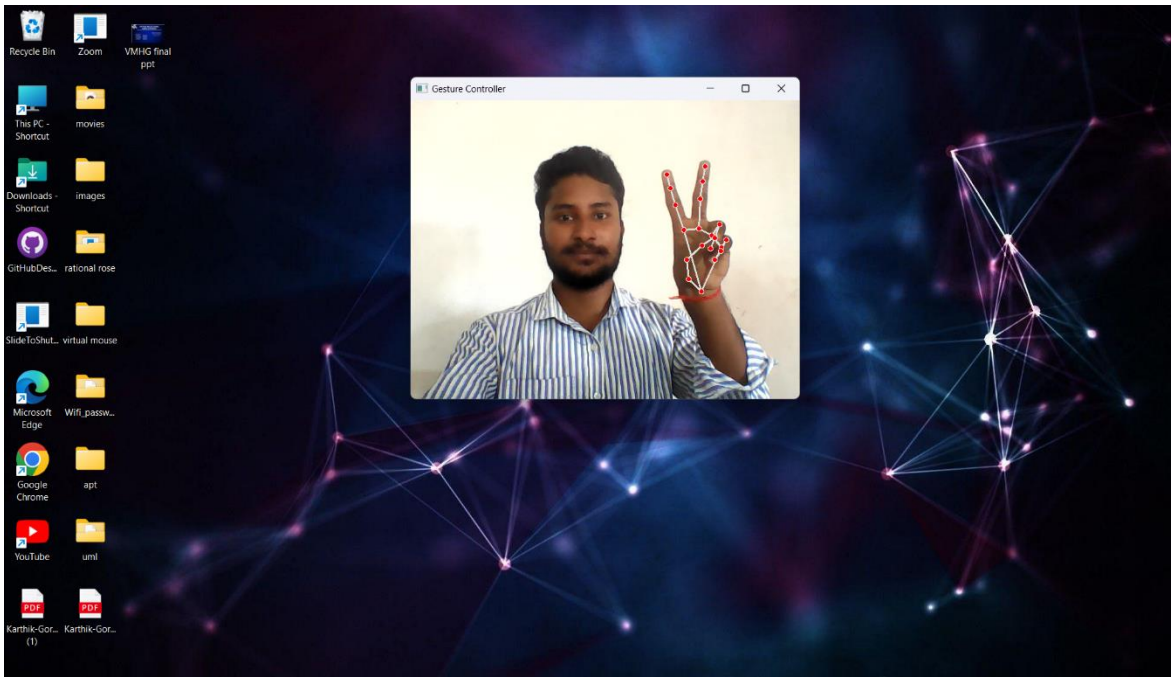
8.OUTPUT SCREENS



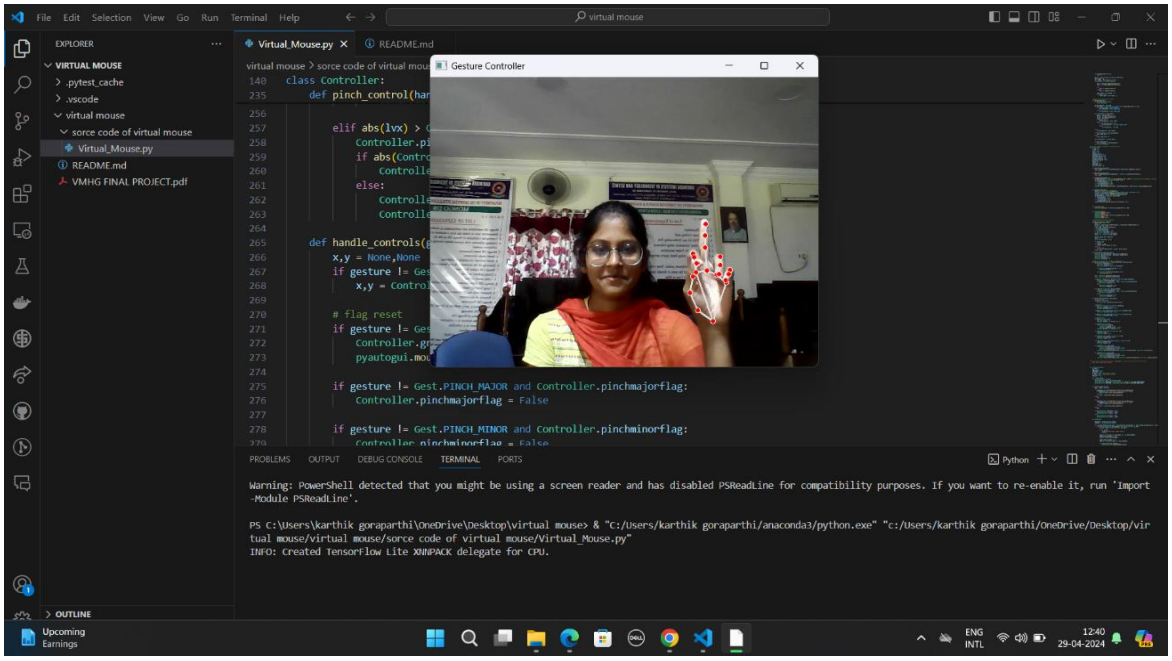
6.1 Hand Recognise



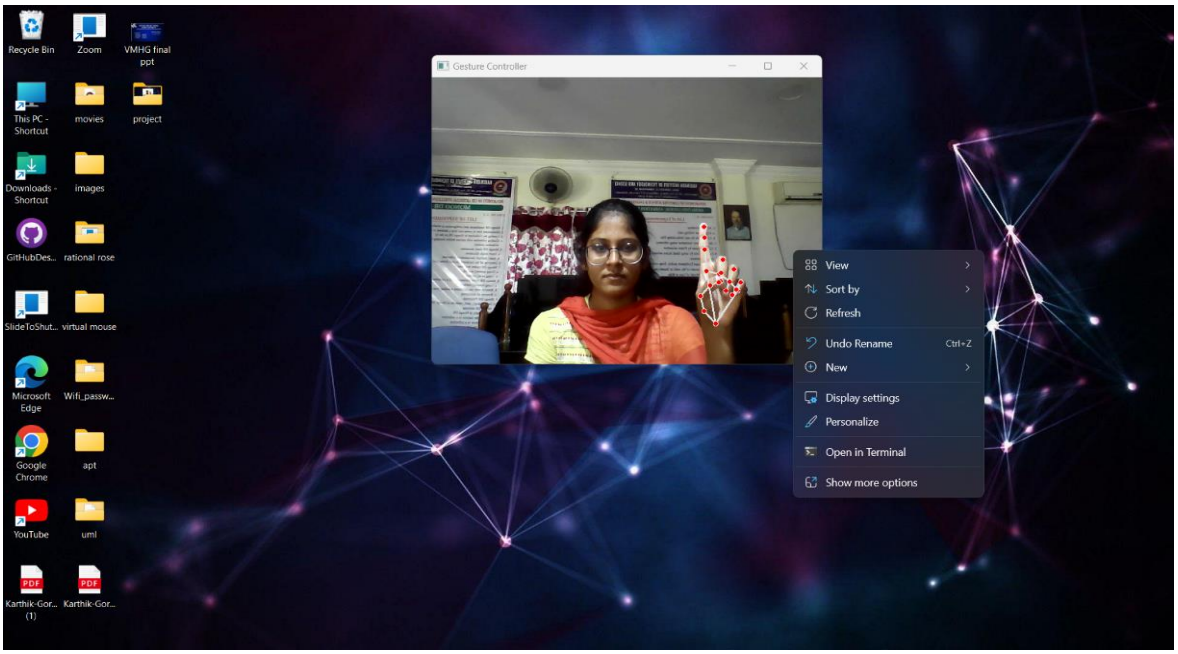
6.2 Detect hand and Two Hands



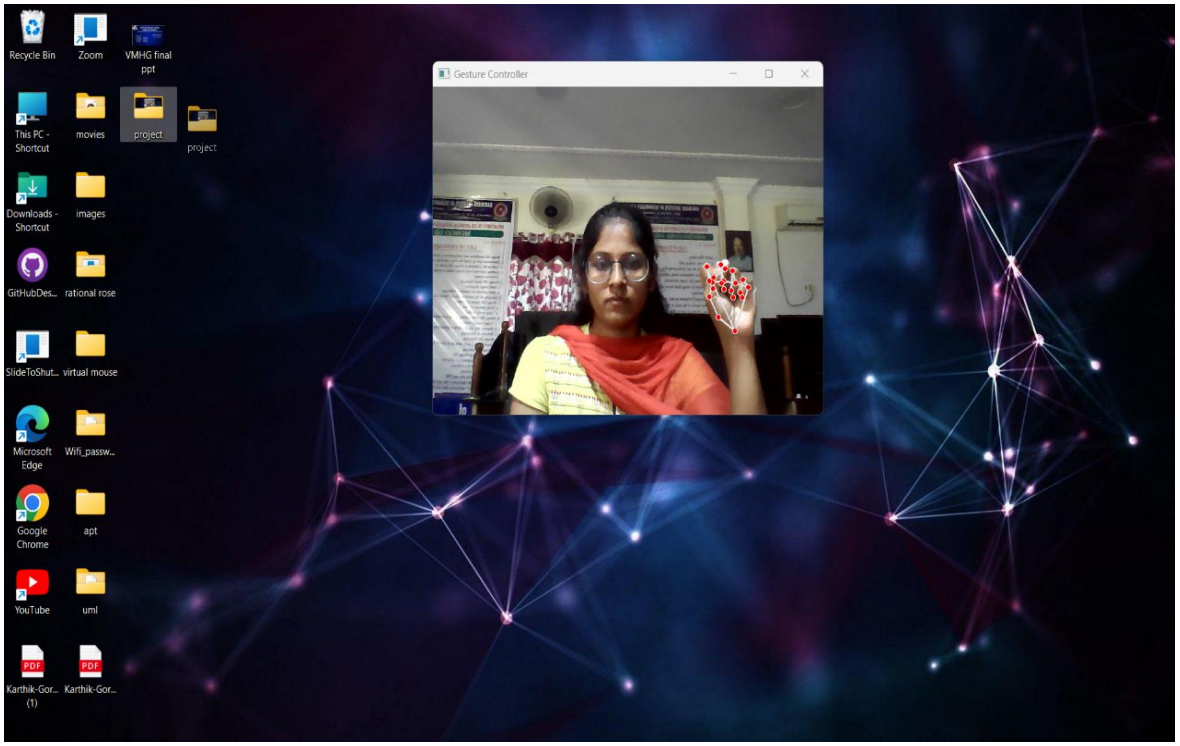
6.3 Cursor Movement



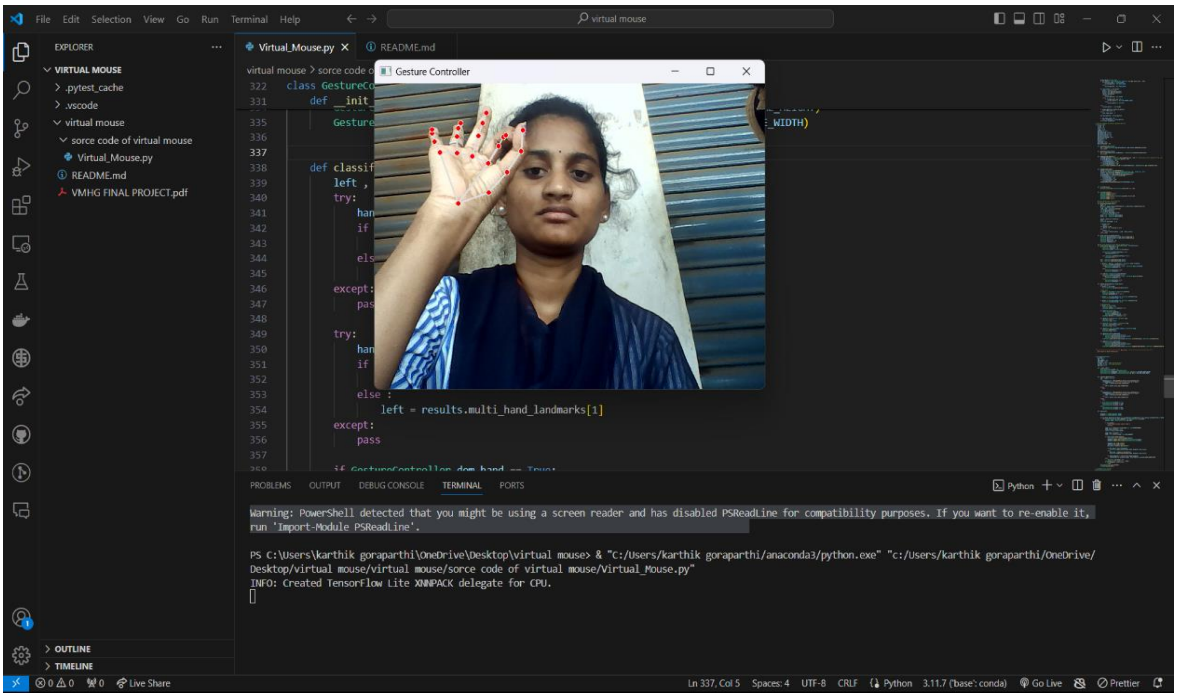
6.4 Left Click



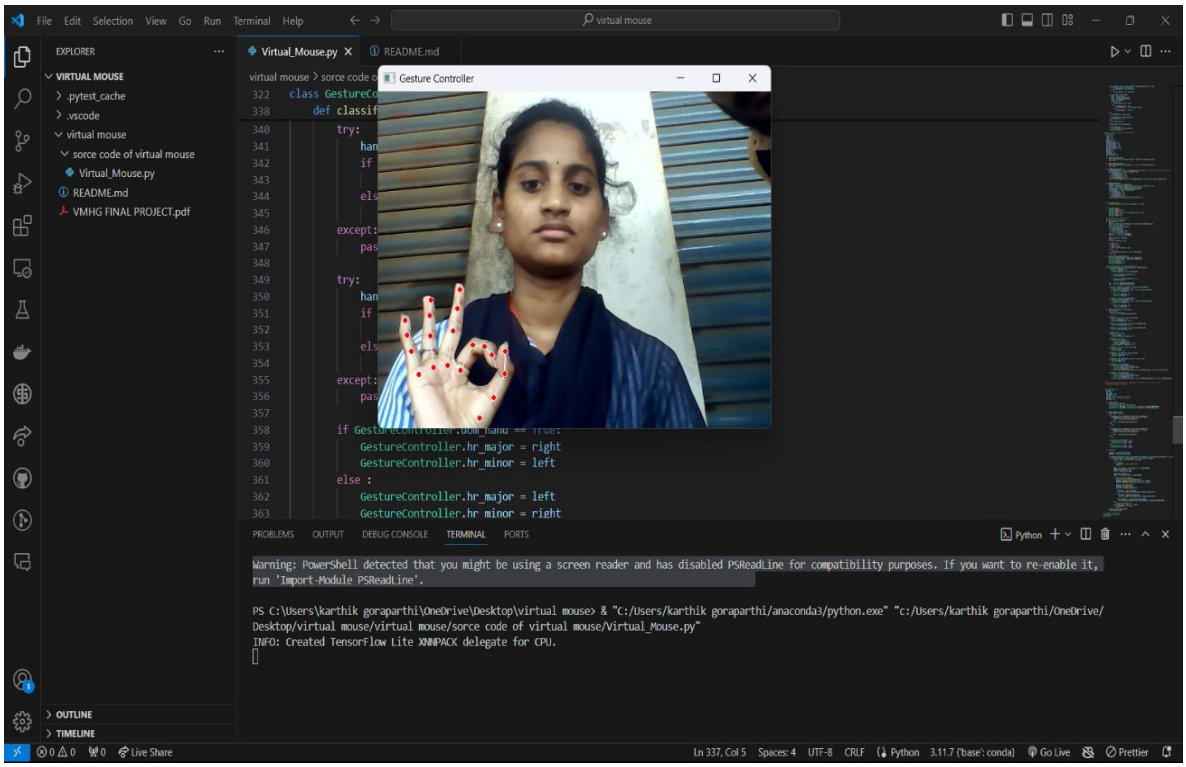
6.5 Right Click



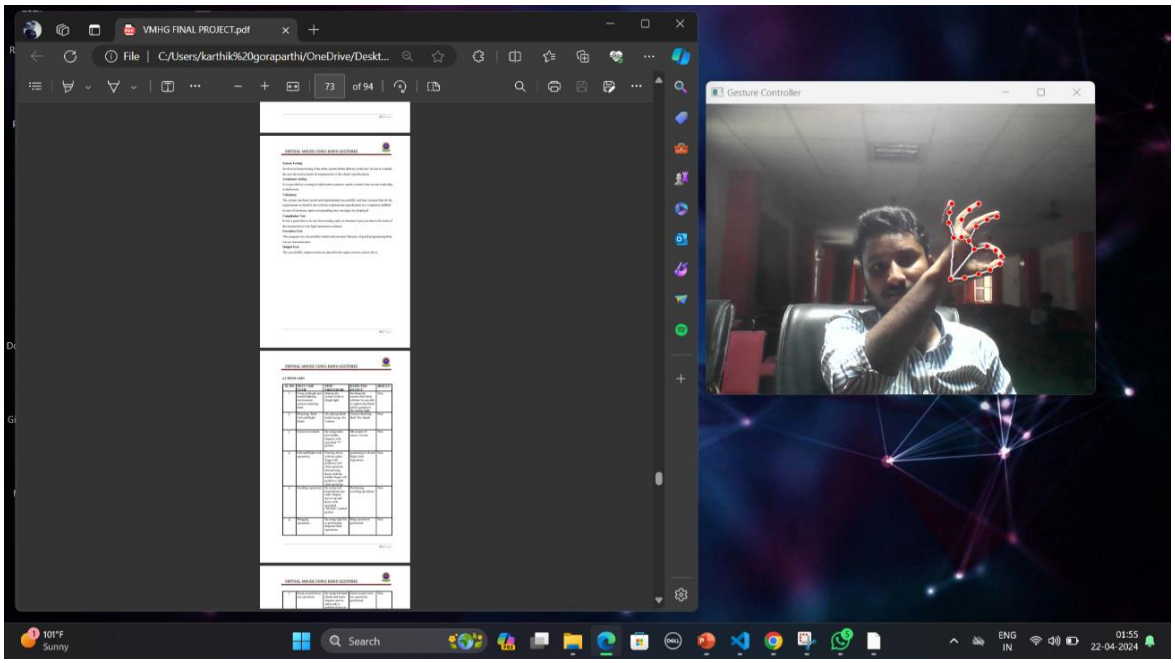
6.6 Drag And Drop



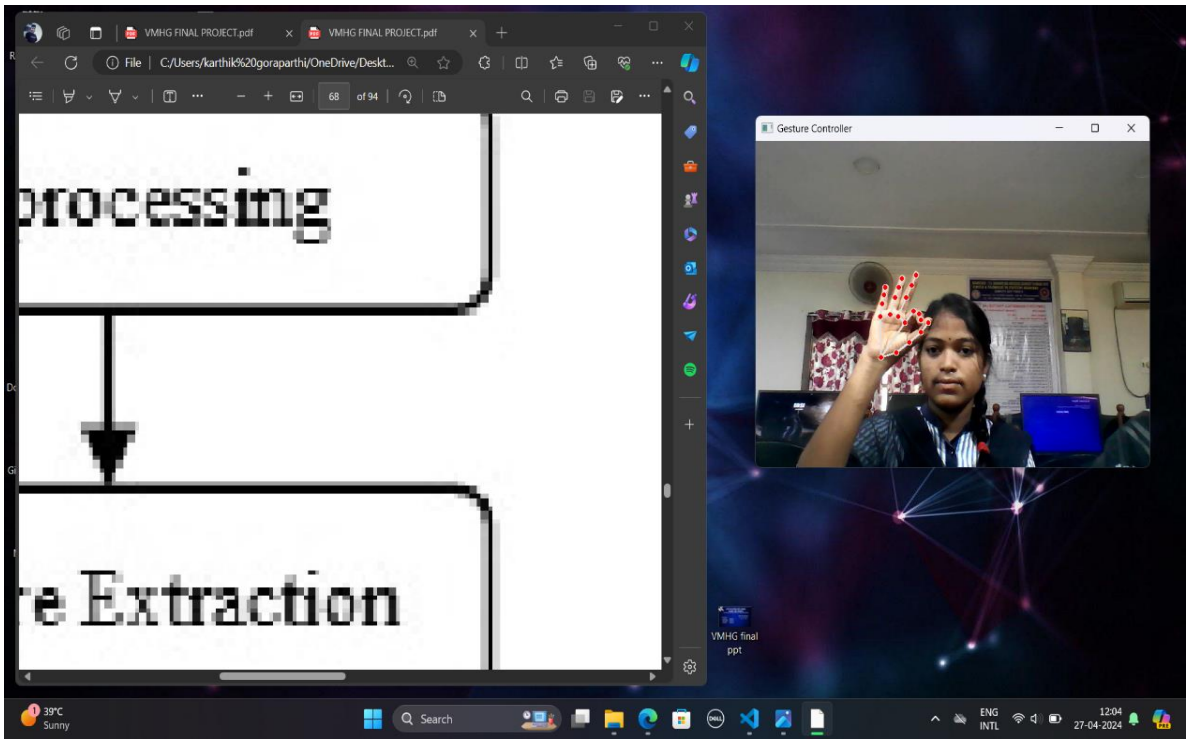
6.7 Scroll up



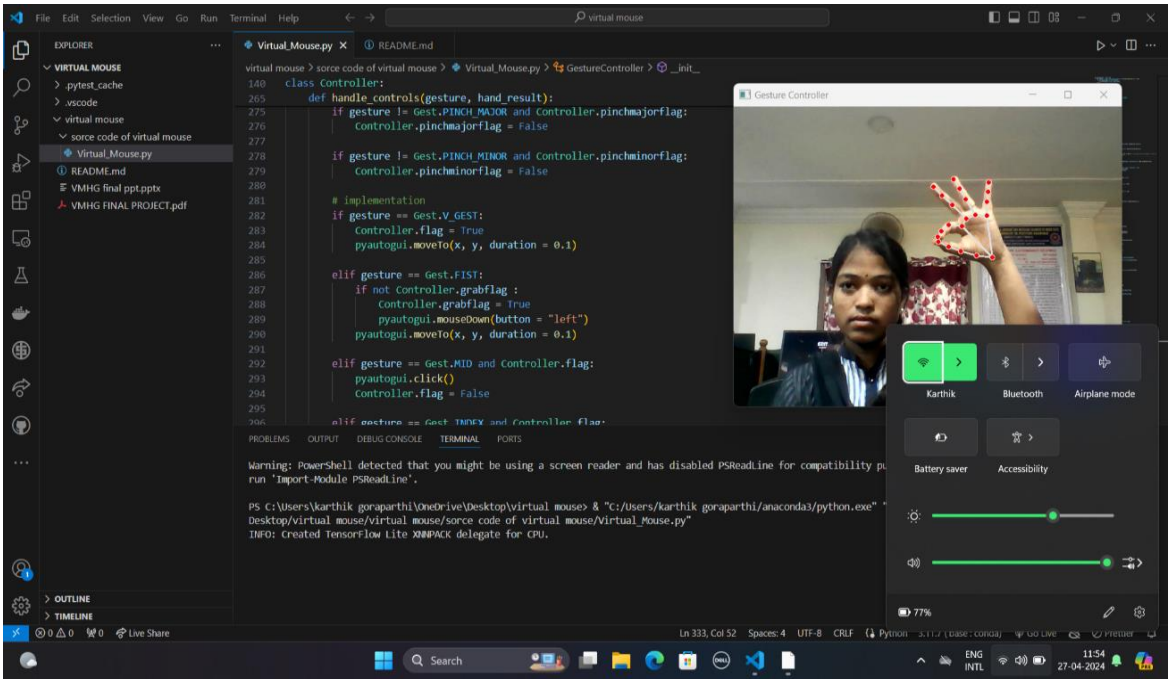
6.8 Scroll Down



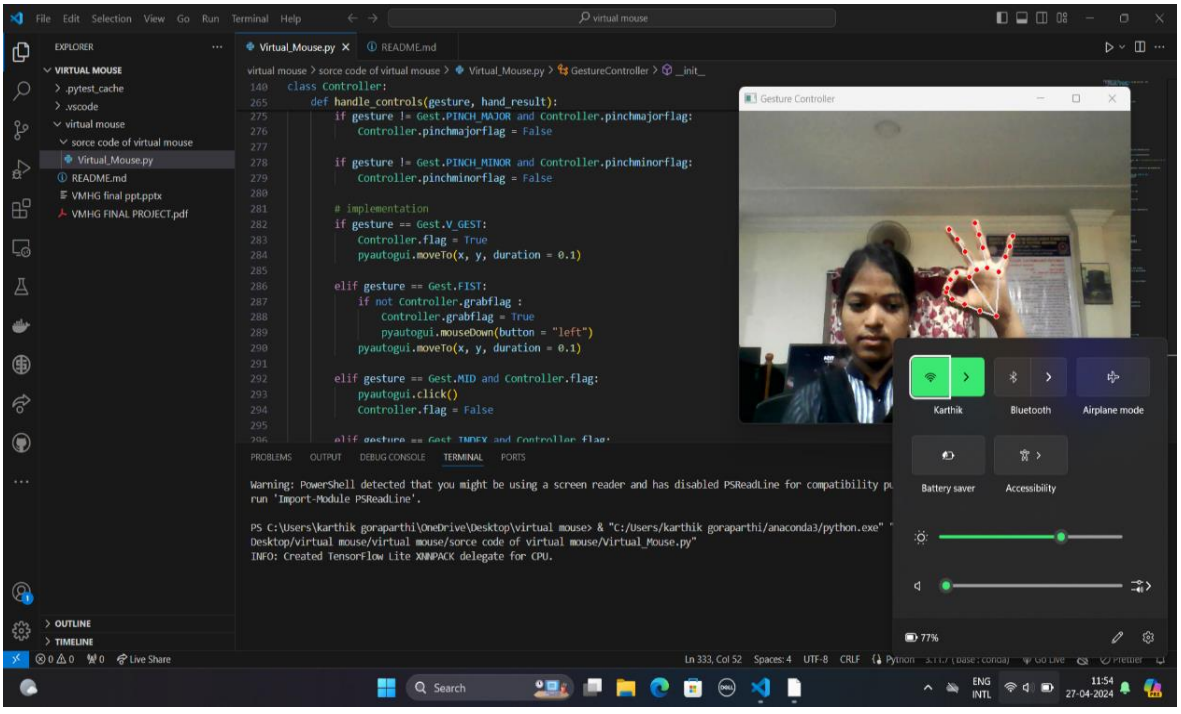
6.9 Zoom In



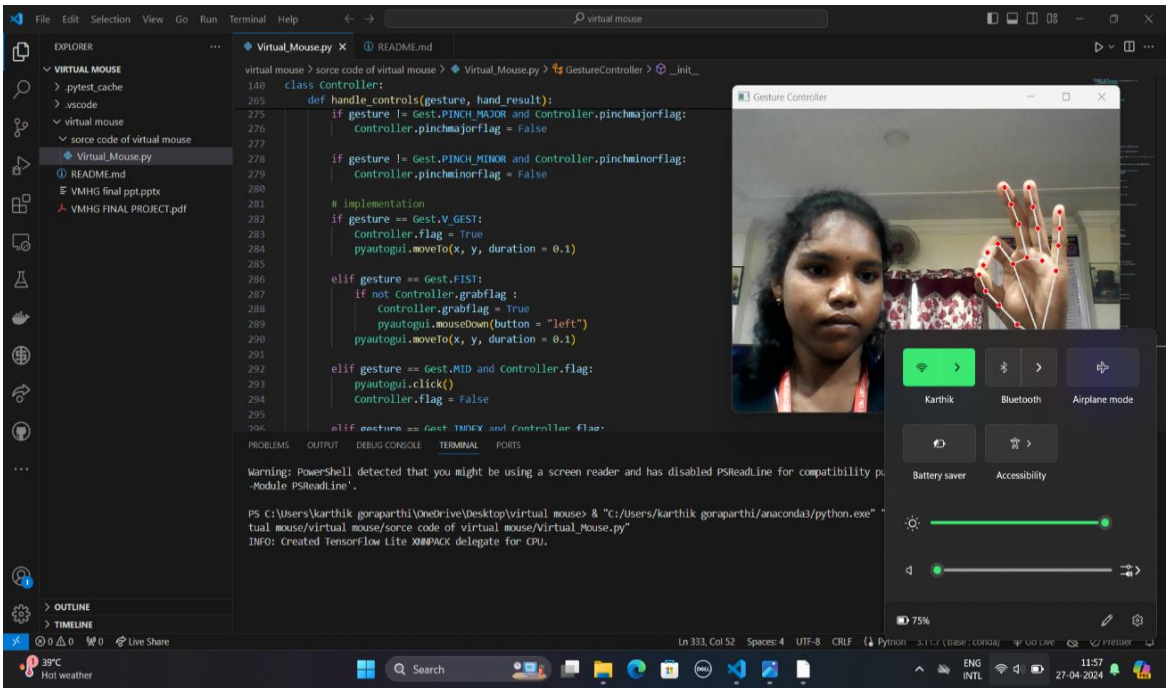
6.10 Zoom out



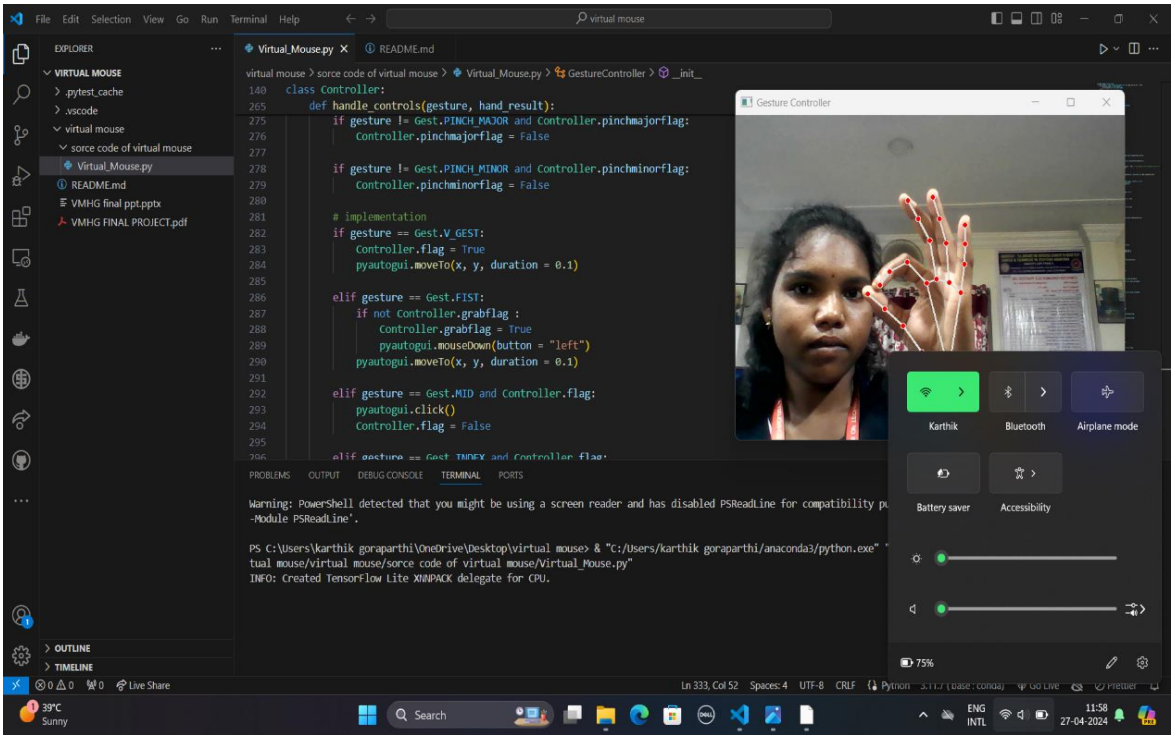
6.11 Volume increasing



6.12 Volume Decreasing



6.13 Brightness increasing



6.14 Brightness Decreasing

9.CONCLUSION & FUTURE SCOPE

CONCLUSION:

Due to accuracy and efficiency plays an important role in making the program as useful as an actual physical mouse, a few techniques had to be implemented. After implanting such type of application there is big replacement of physical mouse i.e., there is no need of any physical mouse. Each & every movement of physical mouse is done with this motion tracking mouse (virtual mouse).

There are several features and improvements needed in order for the program to be more user friendly, accurate, and flexible in various environments. The following describes the improvements and the features required:

FUTURE SCOPE:

a) Smart Movement: Due to the current recognition process are limited within 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.

b) Better Accuracy & Performance: The response time are heavily relying 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 webcam. Therefore, the program may have better performance when it's running on a decent machine with a webcam that performs better in different types of lightings.

c) Mobile Application: In future this web application also able to use on Android devices, where touchscreen concept is replaced by hand gestures.

D) **Enhancement:** To Control whole System using the Hand.

E) **Gesture Based Passwords:** To perform the security based passwaords with the specified gesture .

10. BIBLIOGRAPHY

FOR PYTHON INSTALLATION

https://www.python.org/downloads/release/python-3104/

FOR PYTHON HISTORY
https://www.javatpoint.com/python-tutorial

FOR PYTHON ARCHITECTURE
https://www.geeksforgeeks.org/internal-working-of-python/

FOR VISUAL STUDIO CODE INSTALLATION
https://visualstudio.microsoft.com/downloads/

FOR OPENCV INSTALLATION

www.opencv.org

FOR FRAGMENTS
https://pypi.org/project/fragments/

FOR UML DIAGRAMS
https://staruml.io/