TOUCHLESS MOUSE INTERACTIONS BASED ON HAND GESTURE RECOGNITION

A DESIGN PROJECT REPORT

submitted by

DEVADHARSHINI S

DHIVYA S

ELAKKIYA S

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

TOUCHLESS MOUSE INTERACTIONS BASED ON HAND GESTURE RECOGNITION

A DESIGN PROJECT REPORT

submitted by

DEVADHARSHINI S (811722104028)

DHIVYA S (811722104034)

ELAKKIYA S (811722104038)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **"TOUCHLESS MOUSE INTERACTIONS BASED ON HAND GESTURE RECOGNITION"** is a Bonafide work of **DEVADHARSHINI S (811722104028), DHIVYA S (811722104034), ELAKKIYA S (811722104038)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. A Delphin Carolina Rani, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR
Department of CSE
K Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

SIGNATURE
Mrs.M Mathumathi, M.E.,(Ph.D.),
SUPERVISOR
Assistant Professor
Department of CSE
K Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

Submitted for the viva-voice examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"TOUCHLESS MOUSE INTERACTIONS BASED ON HAND GESTURE RECOGNITION"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of Bachelor of Engineering. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor of Engineering.

Signature

____

DEVADHARSHINI S

____

DHIVYA S

____

ELAKKIYA S

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtedness to our institution **"K RAMAKRISHNAN COLLEGE OF TECHNOLOGY"**, for providing us with the opportunity to do this project.

We are glad to credit and praise our honorable and respected chairman sir **Dr. K. RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering adequate duration to complete it.

We would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project with full satisfaction.

We heartily thank **Dr. A. DELPHIN CAROLINA RANI, M.E., Ph.D.,** Head of the Department, **COMPUTER SCIENCE AND ENGINEERING** for providing her support to pursue this project.

We express our deep and sincere gratitude and thanks to our project guide **Mrs. M. Mathumathi, M.E.,(Ph.D.),** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry our this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course. We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

In the modern agricultural landscape, accurate soil analysis is crucial for optimizing crop yield and ensuring sustainable farming practices. Our project introduces an AI-based soil analysis system that utilizes advanced image processing and machine learning techniques to analyze soil from a simple image upload. Users can capture or upload an image of soil, and the system automatically enhances the image quality to improve feature extraction and analysis. This automation removes the need for expensive lab tests and makes soil assessment accessible to all, especially small-scale farmers. Once the image is processed, the system identifies the soil type—such as sandy, loamy, clay, or silt—using a trained machine learning model. Based on this classification, it recommends suitable crops that can be grown efficiently in the identified soil type, considering regional and climatic conditions. This recommendation system ensures that the users are equipped with knowledge that enhances both productivity and sustainability. Solution performs a virtual mineral analysis using visual and pattern- based indicators to estimate the presence of key nutrients and minerals in the soil, such as nitrogen, phosphorus, and potassium. It also suggests which essential minerals are lacking or present in excess and recommends necessary soil treatments. This feature supports better fertilizer use, reducing both costs and environmental impact.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

HCI	Human-Computer Interaction Interface
SVM	Support Vector Machine
ML	Machine Learning
SSD	Solid State Drive
CNN	Convolutional Neural Networks
LAN	Local Area Network
API	Application Programming Interface
GUI	Graphical User Interface
AST	Abstract Syntax Tree
REPL	Read-Eval-Print Loop
GIMP	GNU Image Manipulation Program
KNN	K-Nearest Neighbour
UML	Unified Modeling Language
DFD	Data Flow Diagram

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Predictive analytics tools are powered by several different models and algorithms that can be applied to wide range of use cases. Determining what predictive modeling techniques are best for your company is key to getting the most and leveraging data to make insightful decisions in the statistical context, Machine Learning is defined as an application of artificial intelligence where available information is used through algorithms to process or assist the processing of statistical data. While Machine Learning involves concepts of automation, it requires human guidance. Machine Learning involves a high level of generalization in order to get a system that performs well on yet unseen data instances.

Machine learning is a relatively new discipline within Computer Science that provides a collection of data analysis techniques. Some of these techniques are based on well-established statistical methods (e.g. logistic regression and principal component analysis) while many others are not.

Most statistical techniques follow the paradigm of determining a particular probabilistic model that best describes observed data among a class of related models. Similarly, most machine learning techniques are designed to find models that best fit data (i.e. they solve certain optimization problems), except that these machine learning models are no longer restricted to probabilistic ones.

The important advantage of machine learning techniques over statistical ones is that the latter require underlying probabilistic models while the former do not. Even though some machine learning techniques use probabilistic models, the classical statistical techniques are most often too stringent for the oncoming Big Data era, because data sources are increasingly complex and multi-faceted. Prescribing probabilistic models relating variables from disparate data sources that are plausible and amenable to statistical analysis might be extremely difficult if not impossible. Machine learning might be able to provide a broader class of more flexible alternative analysis methods better suited to modern sources of data. It is imperative for statistical agencies to explore the possible use of machine learning techniques to determine whether their future needs might be better met with such techniques than with traditional ones.

1.2 CLASSES OF MACHINE LEARNING

Examples of supervised learning

Logistic regression, when used for prediction purposes, is an example of supervised machine learning. In logistic regression, the values of a binary response variable (with values 0 or 1, say) as well as a number of predictor variables (covariates) are observed for a number of observation units. These are called training data in machine learning terminology. The main hypotheses are that the response variable follows a Bernoulli distribution (a class of probabilistic models), and the link between the response and predictor variables is the relation that the logarithm of the posterior odds of the response is a linear function of the predictors.

The response variables of the units are assumed to be independent of each other, and the method of maximum likelihood is applied to their joint probability distribution to find the optimal values for the coefficients (these parameterize the aforementioned joint distribution) in this linear function. The particular model with these optimal coefficient values is called the ―fitted model,‖ and can be used to ―predict‖ the value of the response variable for a new unit (or, ―classify‖ the new unit as 0 or 1) for which only the predictor values are known. Support Vector Machines (SVM) are an example of a non-statistical supervised machine learning technique; it has the same goal as the logistic regression classifier just described: Given training data, find the best-fitting SVM model, and then use the fitted SVM model to classify.

1.3 MACHINE LEARNING

Machine learning (**ML**) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

Virtual Mouse:

Virtual-Mouse is an application, which can replace your Computer Mouse with normal keystrokes on your keyboard! It allows you to assign any key or combination of keys to move around your mouse cursor and to execute various mouse occurrences like Double Clicking etc. Virtual-Mouse can be used both alone and together with your present mouse. Virtual-Mouse also can help you to eliminate arm and wrist pain caused by your mouse. The Network Administrator or a computer supplier you can find many advantages in using the Virtual-Mouse instead of a computer mouse. This contactless solution is particularly useful in hygienic or assistive environments, such as medical facilities or for users with physical disabilities. The virtual mouse offers a more interactive, accessible, and futuristic way to navigate a computer system. Although it requires a well-lit environment and may experience slight performance issues on low-end devices.

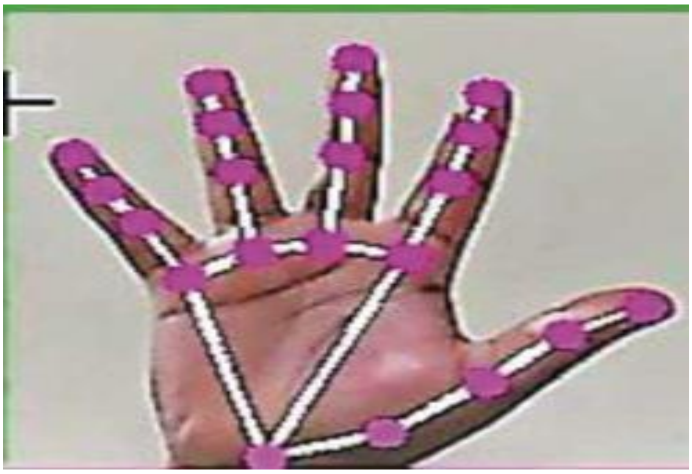

Fig.1.3 Hand Landmarks

Hand Gestures:

A gesture is a movement of the hand, arms, or other body part that is intended to indicate or emphasize something, often when speaking. In other words, gestures are body movements that express something. For example, a wave of the hand is a common gesture used to say hello to someone.

What are examples of gestures?

Gestures and Movement

- Frequent and even wild hand gestures.
- Finger pointing.
- Arms waving in the air.
- Raking fingers through their hair.
- Invasion of personal space in order to send a message of hostility.

Image capture:

Image Capture is an application program from Apple that enables users to upload pictures from digital cameras or scanners which are either connected directly to the computer or the network. It provides no organizational tools like a photo but is useful for collating pictures from a variety of sources with no need for drivers. The help of a web camera that captures the hand gestures and hand tip and then processes these frames to perform the particular mouse function such as left click, right click, and scrolling function.

The COVID-19 situation, it is not safe to use the devices by touching them because it may result in a possible situation of spread of the virus by touching the devices. Image capture may involve photochemical or digital or analog electronic technology. Traditional photography is an example of the former; digital photography is an example of the latter. Image capture usually occurs in some type of camera, which optically focuses the image onto the capture device or medium. In astronomical photography, the entire telescope serves as the camera, while in medical radiography, the image is captured initially by a phosphor screen which converts x-radiation to light, then it is recaptured by a piece of photographic film in contact with the screen.

1.4 PROBLEM DESCRIPTION :

The proposed AI virtual mouse system can be used to overcome problems in the real world such as situations where there is no space to use a physical mouse and also for the persons who have problems in their hands and are not able to control a physical mouse. Also, amidst of the COVID-19 situation, it is not safe to use the devices by touching them because it may result in a possible situation of spread of the virus by touching the devices, so the proposed AI virtual mouse can be used to overcome these problems since hand gesture and hand Tip detection is used to control the PC mouse functions by using a webcam or a built-in camera.

1.5 OBJECTIVE:

Main objective of the proposed AI virtual mouse system is to develop an alternative to the regular and traditional mouse system to perform and control the mouse functions, and this can be achieved with the help of a web camera that captures the hand gestures and hand tip and then processes these frames to perform the particular mouse function such as left click, right click, and scrolling function.

CHAPTER 2

LITERATURE SURVEY

TITLE : Cursor Manipulation with Hand Recognition Using Computer Vision **.**

AUTHOR : Mandar Salvi, Shravan Kegade, Aniket Shinde, Bhanu Tekwani

YEAR 2021

This paper aims to make a software program which will Track/Monitor your hand movement in front of the screen through a webcam and will move the cursor of the computing system with respect to your hand movement and can do certain fixed tasks like Right Click, Left Click, Scroll, Drag, Switch Between Programs, Go back, Forward, etc. This program will work in background and use convolutional Neural Networks Model (SSD) to convolve each and every video frame coming from input and at the end will classify the image into classes after further processing of the predicted class it will do necessary operations on Mouse/ Track pad driver to perform desired operations. This establishes the basis for the research, a program that can be deployed on any computer with a webcam to instantly create a gamepad out of thin air. The Hand Gesture Recognition System for Games comes to the rescue. Most laptop computers and many desktops come equipped with a webcam, so naturally, that would be the starting point.

ALGORITHM: convolutional Neural Networks (CNN)

MERIT: Hand gestures are a natural and intuitive way for users to interact with computers. This approach mimics real-world movements, making it user-friendly and easy to learn.

DEMERIT: The overall cost and complexity of the system.

TITLE **:** Hand Gesture Recognition System for Games

AUTHOR : Nhat Vu Le; Majed Qarmout; Yu Zhang; Haoren Zhou;

YEAR 2021

Video games are among the most popular forms of entertainment in the modern world. However, many gamers with physical disabilities are impeded by traditional controllers. While accessories such as foot pedals and enlarged buttons exist, many accessible gaming setups can end up costing hundreds of dollars. The solution to this problem must be obtainable by gamers of all demographics, and ideally, incorporate items most people already own. The Hand Gesture Recognition System for Games comes to the rescue. Most laptop computers and many desktops come equipped with a webcam, so naturally, that would be the starting point. Users would be able to perform various hand gestures, with each being mapped to a set of button combinations on a virtual gamepad. As gesture detection would have to be done in real-time, fast Computer Vision libraries such as OpenCV are needed to process images. This establishes the basis for the research, a program that can be deployed on any computer with a webcam to instantly create a gamepad out of thin air. The final program features an intuitive user interface with customizable game profiles that can be saved to or loaded from storage.

ALGORITHM: R-CNN deep learning algorithm

MERIT: R-CNN employs a deep neural network to extract rich hierarchical features from input images, enabling the model to learn complex representations of hand gestures and improve recognition accuracy.

DEMERIT: In this is balancing accuracy and speed is crucial for a seamless gaming experience.

**TITLE : ** Hand Gesture Control for Human–Computer Interaction with Deep Learning

AUTHOR : S. N. David Chua, K. Y. Richard Chin, S. F. Lim & Pushpdant Jain

**YEAR : ** 2022

The use of gesture control has numerous advantages compared to the use of physical hardware. However, it has yet to gain popularity as most gesture control systems require extra sensors or depth cameras to detect or capture the movement of gestures before a meaningful signal can be triggered for corresponding course of action. This research proposes a method for a hand gesture control system with the use of an object detection algorithm, YOLOv3, combined with handcrafted rules to achieve dynamic gesture control on the computer. This project utilizes a single RGB camera for hand gesture recognition and localization. The dataset of all gestures used for training and its corresponding commands, are custom designed by the authors due to the lack of standard gestures specifically for human–computer interaction. The SuperFox game is developed; the game is controlled by commands associated with hand and fingertip movements. The system was tested with 10 participants to evaluate effectiveness of the game controls. Algorithms to integrate gesture commands with virtual mouse and keyboard input through the Pynput library in Python, were developed to handle commands such as mouse control, media control, and others. The mAP result of the YOLOv3 model obtained 96.68% accuracy based on testing result.

ALGORITHM: Deep Learning, YOLOv3

MERIT: YOLOv3 is known for its real-time object detection capabilities.

DEMERIT: This can be a limitation in certain human-computer interaction scenarios.

**TITLE : ** Hand and Fingertip Detection for Game-Based Hand Rehabilitation

AUTHOR : Nattarika Potigutsai; Ohm Sornil

YEAR 2021

Hand rehabilitation is the process of recovering hand movements to return to normal. For rehabilitation to be effective, it is necessary for patients to practice repetitive movements. During the process, the patients usually feel bored, lack motivation and enjoyable time. There is a growing interest in building games for rehabilitation to make it more interesting and motivating for patients. However, the major obstacle is that the devices are expensive, and the patients must go to use them in hospitals. This paper focuses on designing and implementing hand rehabilitation software by using hand and fingertip detection. The purpose of the project is to make the rehabilitation process more accessible, enjoyable, affordable, and can be made portable for patients to use at home. Deep learning techniques were utilized to perform hand and fingertip detection. The SuperFox game is developed; the game is controlled by commands associated with hand and fingertip movements. The system was tested with 10 participants to evaluate effectiveness of the game controls. The results showed that the system is feasible in controlling the game and able to be used to create motivation and enjoyable time for patients.

ALGORITHM: Deep learning, Yolov3

MERIT: It can provide real-time feedback to users, encouraging engagement and motivation.

DEMERIT: The cost and availability of such hardware can be a limiting factor for widespread adoption, especially in resource-constrained environments.

TITLE : A Unified Learning Approach for Hand Gesture Recognition and Fingertip Detection

AUTHOR : Alam, Mohammad Mahmudul Islam, Mohammad Tariqul Rahman.

YEAR 2021

In human-computer interaction or sign language interpretation, recognizing hand gestures and detecting fingertips become ubiquitous in computer vision research. In this paper, a unified approach of convolutional neural network for both hand gesture recognition and fingertip detection is introduced. The proposed algorithm uses a single network to predict the probabilities of finger class and positions of fingertips in one forward propagation of the network. Instead of directly regressing the positions of fingertips from the fully connected layer, the ensemble of the position of fingertips is regressed from the fully convolutional network. Subsequently, the ensemble average is taken to regress the final position of fingertips. Since the whole pipeline uses a single network, it is significantly fast in computation. The proposed method results in remarkably less pixel error as compared to that in the direct regression approach and it outperforms the existing fingertip detection approaches including the Heat map-based framework.

ALGORITHM: Convolutional neural network (CNN)

MERIT: Integrating both tasks into a single model may reduce the overall latency in processing, as the model can perform hand gesture recognition and fingertip detection in parallel. This is beneficial for real-time applications, especially in interactive environments.

DEMERIT: A unified model capable of simultaneously handling hand gesture recognition and fingertip detection can be complex.

TITLE : A Vision Base Application for Virtual Mouse Interface Using Hand Gesture

AUTHOR : Sankha Sarkar, Indrani Naskar, Sourav Sahoo

YEAR 2021

This paper proposes a way of controlling the position of the mouse cursor with the help of a fingertip without using any electronic device. We can be performing the operations like clicking and dragging objects easily with help of different hand gestures. The proposed system will require only a webcam as an input device. Python and OpenCV will be required to implement this software. The output of the webcam will be displayed on the system's screen so that it can be further calibrated by the user. The python dependencies that will be used for implementing this system are NumPy, Autopsy and Media pipe. The developing technologies within the twenty-one century, the areas of virtual reality devices that we are using in our daily lifestyle, these devices are become more compact with wireless technologies like Bluetooth. This paper shows an AI virtual mouse system that take input of hand gestures and detection for fingertip movement to perform mouse operations in computer by using computer vision.

ALGORITHM: Machine learning,OpenCV,

MERIT: Controlling the mouse cursor with hand gestures provides a natural and intuitive mode of interaction. Users can perform actions like clicking and dragging objects using familiar hand movements, making the interface user-friendly.

DEMERIT: The system ability to recognize a diverse range of hand gestures may be limited.

**TITLE : **Gesture Control Virtual Mouse

AUTHOR : Prachi Agarwal, Abhay Varshney, Garvit Bhola, Harsh Gupta and Harshbeer Singh

YEAR 2021

In todays technological era, many technology are evolving day by day. One such promising concept is human- system interface. For example, in a stressed out mouse there is no provision to increase restriction. In wi-fi mouse, one must have Bluetooth hardware installed within the laptop and Bluetooth dongle attached. The proposed era will haven't any such boundaries and could alternatively depend on gesture recognition. In this mission, operations like clicking and dragging of objects may be achieved with different hand gestures. The proposed gadget will only require a webcam as an input tool. The software programs that will be required to put into effect the proposed machine are OpenCV and python. The output of the digital camera may be displayed at the machine's display screen so that it may be in addition calibrated by means of the user. The python dependencies that will be used for implementing this machine are NumPy, math, wx and mouse. In this paper, we present a singular approach for human computer interplay (hci) in which cursor motion is controlled using a real-time camera.

ALGORITHM: Artificial intelligence(AI),OpenCV

MERIT: Gesture control provides a natural and intuitive user interface, mimicking real-world hand movements. This can enhance user experience and make computer interaction more accessible to a broader audience.

DEMERIT: Privacy concerns may arise, and it's essential to implement robust security measures to protect user data from unauthorized access

**TITLE : ** Hand Gesture Recognition Based Virtual Mouse Events

AUTHOR : Manav Ranawat; Madhur Rajadhyaksha; Neha Lakhani; Radha Shankarmani

YEAR 2021

This paper proposes a virtual mouse application based on the tracking of different hand gestures. The system eliminates the dependency on any external hardware required to perform mouse actions. A built-in camera tracks the user's hands, predefined gestures are recognized and the corresponding mouse events are executed. This system has been implemented in Python using OpenCV and PyAutoGUI. Similarly, different gestures can be used to control the keyboard, such as a one-finger gesture to choose an alphabet and a four figure gesture to swipe left and right. With no wires or other devices, it will function as a virtual mouse and keyboard. Researchers have studied background conditions, effects of differences in luminance and skin color individually. However, the proposed system aims to take into account all the above factors to build an application most suitable in the real world. The project's only piece of hardware is a webcam, and the coding is done in Python using the Anaconda platform.

ALGORITHM: Image Processing, Convolutional Neural Network (CNN)

MERIT: The system is implemented using Python, OpenCV, and PyAutoGUI, which are widely used and have good compatibility with various operating systems. This facilitates integration with existing software and workflows.

DEMERIT: The performance of the system may be influenced by environmental factors such as lighting conditions, background clutter, or the presence of other objects in the camera's field of view.

TITLE **:** Gesture Recognition Based Virtual Mouse and Keyboard (2020)

AUTHOR **:** Omkar Shinde, Kiran Navale, Dipak Kunjir

YEAR 2021

Nowadays, computer vision has progressed to the point where a computer can recognize its user using a basic image processing algorithm. People are using this vision in many parts of daily life at this point of development, such as face recognition, color detection, automatic cars, and so on. Computer vision is employed in this research to create an optical mouse and keyboard that uses hand motions. The computer's camera will scan the image of various movements made by a person's hand, and the mouse or pointer will move in response to the movement of the gestures, including doing right and left clicks using distinct gestures. Similarly, different gestures can be used to control the keyboard, such as a one-finger gesture to choose an alphabet and a four figure gesture to swipe left and right. With no wires or other devices, it will function as a virtual mouse and keyboard. The project's only piece of hardware is a webcam, and the coding is done in Python using the Anaconda platform. The Convex hull defects are created first, and then an algorithm is created by mapping the mouse and keyboard functions to the flaws using the defect calculations. If you map ping a couple of them with the mouse and keyboard, the computer will recognize the gesture and respond appropriately.

ALGORITHM: Machine learning, Image Processing

MERIT: Users can control the mouse cursor and perform keyboard functions without physically touching any devices, promoting a hands-free computing experience.

DEMERIT: Lower-quality cameras may result in reduced accuracy and responsiveness, impacting overall system performance.

**TITLE : ** Virtual Mouse Control Using Finger Action

**AUTHOR : ** Maniya Chandresh, Patel Pratik & Boda Jagruti

YEAR 2021

In latest era, human–computer interaction has gained more popularity. The mouse is one of the best inventions in human–computer interaction technology. It is used to point the digital world content by using the physical module. To give this existing traditional way of technology an intelligence, we proposed a computer vision-based hand gesture recognition system to interact with the digital world. Our system controls mouse pointer using finger count and finger strip. Our approach mainly focuses on color segmentation, color tracking, extraction of finger contour, and finger counting algorithm. To add-upon this, a speech recognition-based method is used to perform some basic operations on our system. Therefore, the proposed system eliminates the device dependency to interact with the computer, and it would be considered as a good approach to a gesture-based interface for HCI in future.

ALGORITHM: vision-based hand gesture recognition, finger counting

MERIT: Finger actions are a natural and intuitive way to interact with the computer. Users can perform actions such as pointing, clicking, and scrolling using familiar hand gestures, enhancing the user experience.

DEMERIT: Continuous use of finger actions for extended periods may lead to user fatigue. The ergonomics of gesture-based interactions should be considered to ensure user comfort during prolonged use.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system consists of the generic mouse and track pad system of monitor controlling and the non-availability of a hand gesture system. The remote accessing of monitor screen using the hand gesture is unavailable. Even-though it is largely trying to implement the scope is simply restricted in the field of virtual mouse. The existing virtual mouse control system consists of the simple mouse operations using the hand recognition system, where we could perform the basic mouse operation like mouse pointer control, left click, right click, drag etc. The further use of the hand recognition is not been made use of. Even-though there are a number of systems which are used for hand recognition, the system they made used is the static hand recognition which is simply recognition of the shape made by hand and by defining an action for each shape made, which is limited to a number of defined actions and a large amount of confusion.

3.2 PROPOSED SYSTEM

This Virtual Mouse Hand Recognition application uses a simple color cap on the finger without the additional requirement of the hardware for the controlling of the cursor using simple gestures and hand control. This is done using vision based hand gesture recognition with inputs from a webcam. Using the current system even-though there are a number of quick access methods available for the hand and mouse gesture for the laptops, using our project we could make use of the laptop or web-cam and by recognizing the hand gesture we could control mouse and perform basic operations like mouse pointer controlling, select and deselect using left click, and a quick access feature for file transfer between the systems connected via network LAN cable. The project done is a ―Zero Cost‖ hand recognition system for laptops, which uses simple algorithms to determine the hand, hand movements and by assigning an action for each movement. But we have mainly concentrated on the mouse pointing and clicking actions along with an action for the file transfer between connected systems by hand action and the movements.

The system we are implementing which is been written in python code be much more responsive and is easily implemented since python is a simple language and is platform independent with a flexibility and is portable which is desirable in creating a program which is focused in such an aim for creating a Virtual Mouse and Hand Recognition system. The system be much more extendable by defining actions for the hand movement for doing a specific action. It could be further modified to any further extent by implementing such actions for the set of hand gestures, the scope is restricted by your imagination.

Advantages

- This technology can be used to help patients who don’t have control of their limbs.
- Easy to interact with computers.
- Operation speed is high.
- Proposed method consumption of RAM is less than existing method.

3.3 SYSTEM ARCHITECTURE

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. Gesture recognition algorithms, often based on OpenCV or MediaPipe, analyze the hand movements and map them to specific mouse actions such as cursor movement, clicking.

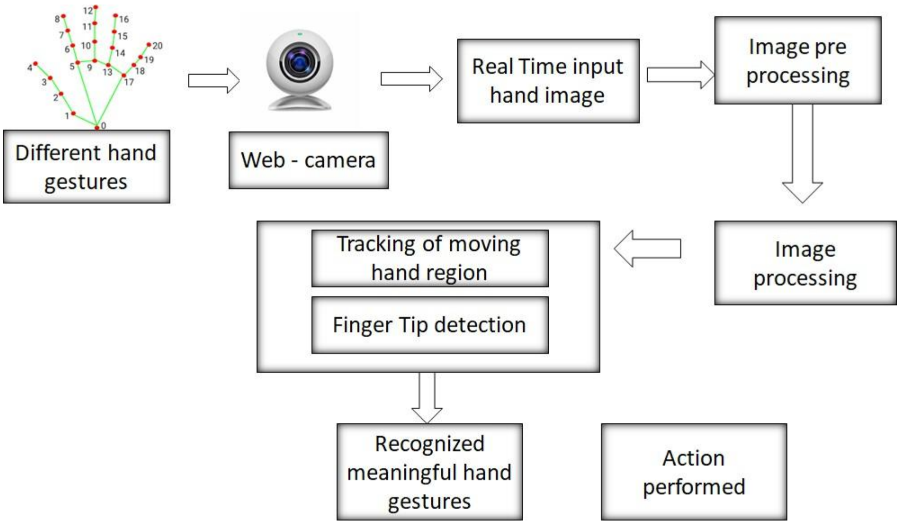

Fig.3.3 Architecture Diagram

The System begins by capturing frames from a webcam, utilizing computer vision techniques to interpret hand gestures for mouse control. The camera continuously captures real-time video frames, and each frame is processed to identify and track the user's hand. A sophisticated computer vision algorithm is employed to detect specific hand poses and movements. This involves color detection techniques, which can be implemented to recognize color caps or tapes worn by the user. Once the hand gestures are identified, the system maps them to corresponding mouse movements and operations. For instance, a pointing gesture translates into cursor movement on the screen, allowing users to navigate through the digital interface. Clicking gestures, recognized through specific hand poses or tapping motions, trigger mouse clicks, enabling users to interact with on-screen elements.

The system incorporates techniques for dragging and dropping actions. When a clicking gesture is followed by a continuous hand movement, the system interprets it as a drag-and-drop operation. This functionality facilitates the manipulation of digital objects and selection of text, enhancing user control. Furthermore, scrolling gestures are recognized by tracking specific hand motions mimicking the scrolling action of a mouse wheel. These gestures enable users to scroll through documents, web pages, or other content seamlessly. The algorithm considers not only the hand's position but also the speed and direction of its movement, ensuring accurate interpretation of user intent. The frames captured by the webcam, after undergoing the necessary processing, are displayed on the computer monitor, providing users with a real-time representation of the system's interpretation of their hand gestures.

The mapped gestures initiate corresponding mouse operations, allowing users to interact with the digital environment effortlessly. The output of the system is a dynamic and responsive virtual mouse control interface that relies solely on hand gestures, offering users an intuitive and natural means of navigating, clicking, dragging, and scrolling. This approach eliminates the need for traditional input devices and enhances the overall user experience in human-computer interaction.

3.4 GESTURE RECOGNITION

Gesture recognition play a pivotal role in the functionality of systems that utilize hand movements to control virtual mice. These advanced algorithms are designed to interpret and understand the nuances of hand gestures, enabling the system to recognize predefined gestures and translate them into specific mouse actions. The process typically involves multiple steps, including hand tracking, feature extraction, and classification.

Firstly, the Machine learning algorithm employs computer vision techniques, such as image processing and pattern recognition, to track the movement of the user's hand in real-time. This involves capturing frames from the camera feed and identifying the hand region within each frame. The algorithm then extracts relevant features from the hand, such as the positions of fingertips, palm orientation, and hand shape. Once the features are extracted, the algorithm uses a classification model to determine the corresponding gesture. This model is trained on a dataset that includes examples of various hand gestures mapped to specific mouse actions. Machine learning techniques, such as supervised learning, may be employed during the training phase to enable the algorithm to generalize and accurately recognize gestures not present in the training set.The recognition of predefined gestures allows the system to map these gestures to specific mouse actions, such as moving the cursor, clicking, dragging, or scrolling.

The algorithm's ability to discern the intent behind each gesture is crucial for providing a seamless and intuitive user experience. To enhance accuracy, algorithms may take into account factors like hand speed, direction, and the temporal sequence of gestures. In summary, gesture recognition algorithms leverage computer vision and machine learning techniques to analyze hand movements, extract relevant features, and classify predefined gestures. This process enables the virtual mouse control system to interpret user intentions accurately, providing an efficient and user-friendly interface for interacting with computers.

Pointing:

Pointing gestures are recognized when the system identifies the extended index finger or a distinct hand pose indicative of pointing. The computer vision algorithm analyzes the position of the fingertip in relation to the hand and tracks its movement. When the system detects a pointing gesture, it translates the movement into cursor displacement on the screen. The direction and speed of the hand movement determine the corresponding movement of the cursor, allowing users to point and navigate through digital content.

Clicking:

Clicking gestures involve recognizing a specific hand movement or pose that signifies a mouse click. This can include a tapping motion with a finger or a hand pose indicating a closed fist. When the system identifies a clicking gesture, it interprets this as a command to perform a primary click, simulating the action of pressing the left button on a physical mouse. This enables users to interact with on-screen elements, such as selecting icons or buttons.

Dragging:

Dragging gestures are identified when the system detects a continuous movement of the hand after a clicking gesture. Once the user initiates a click and maintains the hand movement, the system interprets this as a request to click and hold. As a result, users can drag and drop items, select text, or manipulate on-screen elements by sustaining the dragging gesture. Releasing the gesture completes the dragging action.

Scrolling:

Scrolling gestures are interpreted when the system identifies specific hand movements that mimic the scrolling action of a mouse wheel. For instance, a vertical or horizontal wave-like motion of the hand might be recognized as a scroll gesture. The algorithm translates this into the scrolling functionality, enabling users to navigate through documents, web pages, or other content by mimicking the scrolling motion.

To ensure accurate scrolling, the system continuously tracks the position and movement speed of the fingers. For example, a faster upward motion may result in a quicker scroll, while a slow movement can produce a smooth, gradual scroll. Horizontal scrolling can also be enabled using left or right hand swipes or by detecting lateral finger movements. Advanced implementations may use gesture duration, hand tilt, or finger spacing to fine-tune the scrolling speed and direction. The effectiveness of scrolling gestures depends on factors like lighting, background contrast, and webcam quality.

Developers often implement filters and smoothing techniques to prevent accidental scrolls caused by minor or shaky hand movements. Overall, scrolling gestures provide an intuitive and interactive way to browse content, making the virtual mouse more dynamic and user-friendly.

CHAPTER 4

SYSTEM DESCRIPTION

4.1 H/W SYSTEM CONFIGURATION:-

- Processor - Pentium – IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB

4.2 S/W SYSTEM CONFIGURATION:-

- Operating System : Windows 11
- Software : VS code

4.3 SOFTWARE ENVIRONMENT

Python Technology:

Python is an interpreter, high-level, general-purpose programming language. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python Programing Language:

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by Meta programming and met objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python packages with a wide range of functionality, including:

- Easy to Learn and Use
- Expressive Language
- Interpreted Language
- Cross-platform Language

- Free and Open Source
- Object-Oriented Language
- Extensible
- Large Standard Library
- GUI Programming Support
- Integrated

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

One of the key reasons behind its popularity is its clean and readable code structure, which allows developers to write fewer lines of code compared to other programming languages. Python is also platform-independent, meaning programs written in Python can run on various operating systems with minimal or no modification. In recent years, its role in emerging technologies like artificial intelligence, machine learning, and data science has made it even more significant in both academic and industrial domains.

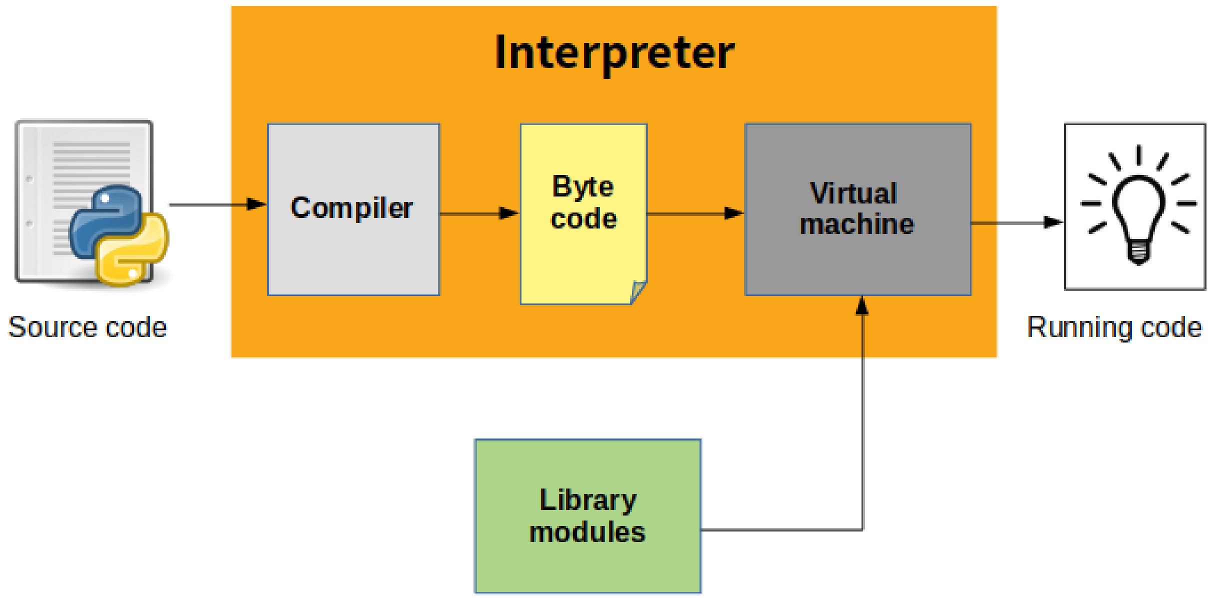

Fig.4.3 Processing diagram

The Python Platform:

The platform module in Python is used to access the underlying platform's data, such as, hardware, operating system, and interpreter version information. The platform module includes tools to see the platform's hardware, operating system, and interpreter version information where the program is running.

There are four functions for getting information about the current Python interpreter. python_version() and python_version_tuple() return different forms of the interpreter version with major, minor, and patch level components. python_compiler() reports on the compiler used to build the interpreter. And python_build() gives a version string for the build of the interpreter.

What does python technology do?

Python is quite popular among programmers, but the practice shows that business owners are also Python development believers and for good reason. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn. Business owners or CTOs appreciate the fact that there's a framework for pretty much anything – from web apps to machine learning.

Moreover, it is not just a language but more a technology platform that has come together through a gigantic collaboration from thousands of individual professional developers forming a huge and peculiar community of aficionados.

So what are the tangible benefits the language brings to those who decided to use it as a core technology? Below you will find just some of those reasons.

Productivity And Speed:

It is a widespread theory within development circles that developing Python applications is approximately up to 10 times faster than developing the same application in Java or C/C++. The impressive benefit in terms of time saving can be explained by the clean object-oriented design, enhanced process control capabilities, and strong integration and text processing capacities. Moreover, its own unit testing framework contributes substantially to its speed and productivity.

Python Is Popular For Web Apps:

Web development shows no signs of slowing down, so technologies for rapid and productive web development still prevail within the market. Along with JavaScript and Ruby, Python, with its most popular web framework Django, has great support for building web apps and is rather popular within the web development community.

Open-Source And Friendly Community:

As stated on the official website, it is developed under an OSI-approved open source license, making it freely usable and distributable. Additionally, the development is driven by the community, actively participating and organizing conference, hackathons, fostering friendliness and knowledge-sharing. Software developers love it for its straightforward syntax and reputation as one of the easiest programming languages to learn.

Python Is Quick To Learn

It is said that the language is relatively simple so you can get pretty quick results without actually wasting too much time on constant improvements and digging into the complex engineering insights of the technology. Even though Python programmers are really in high demand these days, its friendliness and attractiveness only help to increase number of those eager to master this programming language.

Broad Application

It is used for the broadest spectrum of activities and applications for nearly all possible industries. It ranges from simple automation tasks to gaming, web development, and even complex enterprise systems. These are the areas where this technology is still the king with no or little competence:

- Machine learning as it has a plethora of libraries implementing machine learning algorithms.
- Web development as it provides back end for a website or an app.
- Cloud computing as Python is also known to be among one of the most popular cloud-enabled languages even used by Google in numerous enterprise-level software apps.
- Scripting.
- Desktop GUI applications.

Python compiler

The Python compiler package is a tool for analyzing Python source code and generating Python bytecode. The compiler contains libraries to generate an abstract syntax tree from Python source code and to generate Python bytecode from the tree.

The compiler package is a Python source to bytecode translator written in Python. It uses the built-in parser and standard parser module to generate a concrete syntax tree. This tree is used to generate an abstract syntax tree (AST) and then Python bytecode and Scripting.

The full functionality of the package duplicates the built-in compiler provided with the Python interpreter. It is intended to match its behavior almost exactly. Why implement another compiler that does the same thing? The package is useful for a variety of purposes. It can be modified more easily than the built-in compiler. The AST it generates is useful for analyzing Python source code.

The Basic interface

The top-level of the package defines four functions. If you import compiler, you will get these functions and a collection of modules contained in the package.

Limitations

There are some problems with the error checking of the compiler package. The interpreter detects syntax errors in two distinct phases. One set of errors is detected by the interpreter’s parser, the other set by the compiler. The compiler package relies on the interpreter’s parser, so it get the first phases of error checking for free. It implements the second phase itself, and that implementation is incomplete. For example, the compiler package does not raise an error if a name appears more than once in an argument list: def f(x, x): ...A future version of the compiler should fix these problems.

Python Abstract Syntax

The compiler.ast module defines an abstract syntax for Python. In the abstract syntax tree, each node represents a syntactic construct. The root of the tree is Module object.

The abstract syntax offers a higher level interface to parsed Python source code. The parser module and the compiler written in C for the Python interpreter use a concrete syntax tree. The concrete syntax is tied closely to the grammar description used for the Python parser. Instead of a single node for a construct, there are often several levels of nested nodes that are introduced by Python's precedence rules.

The abstract syntax tree is created by the compiler.transformer module. The transformer relies on the built-in Python parser to generate a concrete syntax tree. It generates an abstract syntax tree from the concrete tree.

The transformer module was created by Greg Stein and Bill Tutt for an experimental Python-to-C compiler. The current version contains a number of modifications and improvements, but the basic form of the abstract syntax and of the transformer are due to Stein and Tutt.

Ast Nodes

The compiler.ast module is generated from a text file that describes each node type and its elements. Each node type is represented as a class that inherits from the abstract base class compiler.ast.Node and defines a set of named attributes for child nodes.

Development Environments:

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Other shells, including IDLE and IPython, add further abilities such as auto-completion, session state retention and syntax highlighting.

4.4 IMPLEMENTATIONS

Reference implementation

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several select C99 features. It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including

Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available.

Other implementations

PyPy is a fast, compliant interpreter of Python 2.7 and 3.5. Its just-in-time compiler brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.

Stackless Python is a significant fork of CPython that implements microthreads; it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a stackless version.

MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers. This includes Lego Mindstorms EV3.

Unsupported implementations

Other just-in-time Python compilers have been developed, but are now unsupported:

Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter five-fold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores, while ordinary implementations suffer from the global interpreter lock.

Psyco is a just-in-time specialising compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than standard Python code.

In 2005, Nokia released a Python interpreter for the Series 60 mobile phones named PyS60. It includes many of the modules from the CPython implementations and some additional modules to integrate with the Symbian operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia N900 also supports Python with GTK widget libraries.

Cross-compilers to other languages

There are several compilers to high-level object languages, with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

- Jython enables the use of the Java class library from a Python program.
- IronPython follows a similar approach in order to run Python programs on the .NET Common Language Runtime.
- The RPython language can be compiled to C, and is used to build the PyPy interpreter of Python.
- Pyjs compiles Python to JavaScript.
- Cython compiles Python to C and C++.
- Numba uses LLVM to compile Python to machine code.
- Pythran compiles Python to C++.
- Somewhat dated Pyrex (latest release in 2010) and Shed Skin (latest release in 2013) compile to C and C++ respectively.
- Google's Grumpy compiles Python to Go.
- MyHDL compiles Python to VHDL.
- Nuitka compiles Python into C++.

4.5 PERFORMANCE

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13.

Api Documentation Generators:

Python API documentation generators include:

- Sphinx
- Epydoc
- HeaderDoc
- Pydoc

Uses

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers.

Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

Python is used extensively in the information security industry, including in exploit development.Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language. GNU Debugger uses Python as a pretty printer to show complex structure

4.6 LIBRARIES

In computer programming, a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

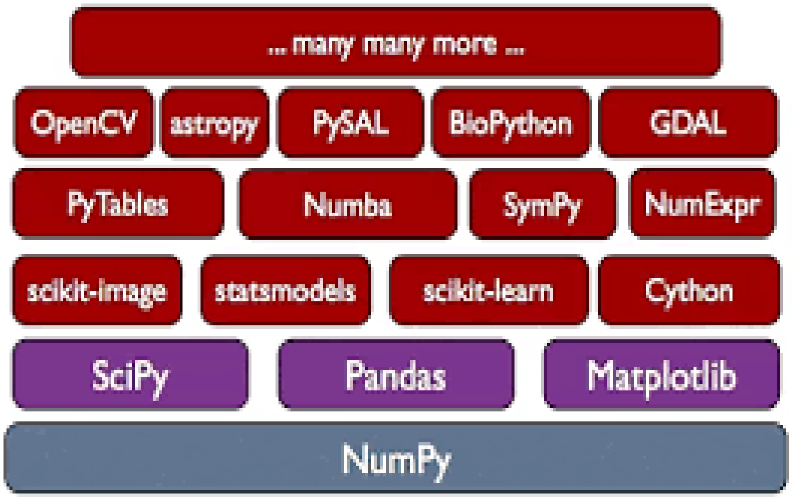

Fig.4.7 Libraries

Library features

- Data Frame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and sub setting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.

CHAPTER 5

MODULE IMPLEMENTATION

5.1 MODULE LIST

- Hand Tracking Module
- Gesture Recognition Module
- Mouse Control Module
- Gesture-Based Calculator Module
- Main Application / Integration Module

5.2 MODULE DESCRIPTION

5.2.1 Hand Tracking Module

- The Hand Tracking Module is the foundation of this project, responsible for detecting and tracking the user's hand in real-time using a webcam. This module utilizes the MediaPipe Hands solution by Google, which can detect up to 21 landmarks on each hand, including finger tips, knuckles, and the wrist. By processing each video frame from the webcam, it accurately identifies the position of the hand and extracts the coordinates of each landmark. These coordinates are crucial for determining gestures and interpreting finger movements.
- The module also includes functionality for drawing these landmarks and connections on the video feed for visualization. Robust hand tracking is essential for both mouse control and calculator operations, making this module the core of the entire system. It handles noise, varying lighting conditions, and quick hand movements to ensure high accuracy and responsiveness for determining gestures and interpreting finger movements and gesture recognition .

- These coordinates allow the system to analyze finger positions, distinguish between open and closed fingers, and recognize specific gestures. For instance, identifying whether the index finger is raised while the others are closed can trigger mouse movement, while more complex gestures can be mapped to calculator inputs. Additionally, the module provides real-time **visual feedback** by drawing landmarks and their connections directly on the live video feed, aiding both debugging and user interaction by making gesture recognition visible and intuitive.

5.2.2 Gesture Recognition Module.

- The Gesture Recognition Module interprets the hand landmark data received from the Hand Tracking Module and converts it into meaningful gestures. These gestures can represent mouse actions such as cursor movement, left or right clicks, as well as calculator-related inputs like numerical digits and operations.
- By analyzing the relative positions of fingers (e.g., whether a finger is open or closed), the module can determine which fingers are being held up and recognize combinations like pinches or swipes. For instance, if the thumb and index finger tips are close together, it may be interpreted as a click gesture.
- The module is designed to handle gesture ambiguity by setting distance thresholds and using temporal smoothing to avoid rapid false detections. It serves as the brain of the interaction system, enabling the system to interpret natural human gestures into digital commands accurately.So for this, we use data pre-processing task crucial step while creating a deep learning model. These gestures can represent mouse actions such as cursor movement, left or right clicks, as well as calculator-related inputs like numerical digits and operations.

5.2.3 Mouse control Module

- The Mouse Control Module translates the recognized hand gestures into virtual mouse actions using the pyautogui library. When the Gesture Recognition Module identifies that the index finger is extended, position is used to move the mouse cursor across the screen.
- This involves converting the finger's position from camera coordinates to screen coordinates using interpolation functions. If a "pinch" gesture is detected (e.g., thumb and index finger tips touching), the module performs a mouse click. Additional gestures can be mapped to perform right-clicks, double-clicks, or scrolling actions, depending on the application's complexity. This module essentially allows users to control their computer interface without touching any device, providing a hygienic and futuristic alternative to traditional input methods.
- It is particularly useful in scenarios where physical contact is undesirable, such as in medical or industrial environments to build model, compute performance metrics and choose best performing model.

5.2.4 Gesture-Based Calculator Module

- The Gesture-Based Calculator Module enables users to perform basic arithmetic operations using only hand gestures. This module leverages the ability to count fingers as numeric input—showing one finger indicates the number 1, two fingers represent 2, and so on.
- Specific gestures are used to identify mathematical operations: for instance, a "peace" sign (two fingers) might represent addition, while a closed fist might indicate evaluation.
- The calculator builds an expression string in memory as the user inputs digits and operations via gestures. Once a gesture signaling execution is

detected, the expression is evaluated using Python’s eval() function or a custom parser, and the result is displayed on screen.

- This module provides a novel way of interacting with a calculator, making it accessible and hands-free. It also demonstrates the system’s flexibility in interpreting different gesture contexts beyond just mouse control.based on the problem’s requirements.

5.2.5 Main Application /Integration Module

- The Main Application Module serves as the central coordinator that integrates all other modules into a single, seamless experience. It initializes the webcam feed, sets up the hand detection and tracking pipeline, and continuously processes frames in a loop.
- Within each loop iteration, the application passes the video frame to the Hand Tracking Module, processes the landmarks via the Gesture Recognition Module, and decides whether to invoke Mouse Control or Calculator functionality based on the current mode.
- This module also handles user interface elements such as displaying the live camera feed, drawing hand landmarks, and showing real-time feedback like cursor position or calculator expressions.
- Additionally, it can support mode switching using specific gestures—for example, holding up five fingers might switch from mouse mode to calculator mode. The Main Application Module ensures smooth communication between subsystems and provides the user with a cohesive, interactive experience that merges gesture control with practical functionality, drawing hand landmarks, and showing real-time feedback like cursor position or calculator expressions. It initializes the webcam feed, sets up the hand detection and tracking pipeline, and continuously processes frames in a loop, like cursor position or calculator expressions.

CHAPTER 6

SYSTEM DESIGN

6.1 UML DIAGRAM

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.

6.1.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

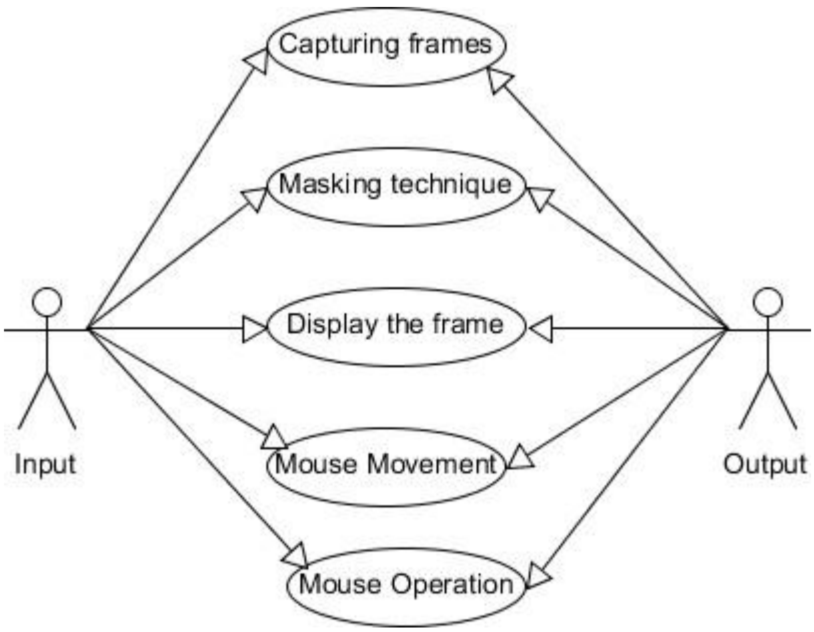

Fig.6.1.1 Use case Diagram

6.1.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among the classes. It explains which class contains information. It help developers understand the **blueprint of the system**, making it easier to design, communicate, and implement software efficiently.

Fig.6.1.2 Class Diagram

6.1.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

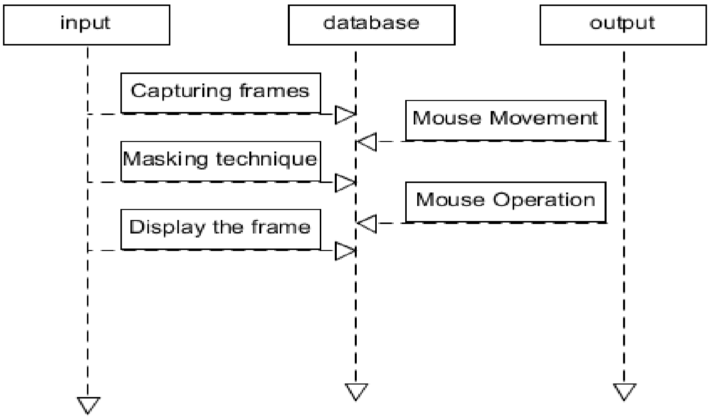

Fig.6.1.3 Sequence Diagram

6.1.4 DEPLOYMENT:

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components. They are also widely used to model the physical aspects of object-oriented software systems, such as executable files, libraries, and databases. By breaking down a system into manageable parts, component diagrams promote modularity, scalability, and maintainability in software architecture.

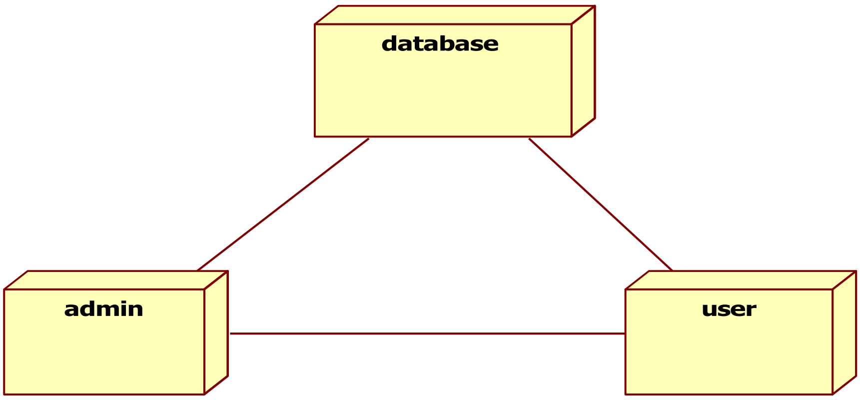

Fig.6.1.4 Deployment Diagram

6.1.5 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.
5. This tool is particularly helpful for both technical and non-technical stakeholders to understand system requirements, identify inefficiencies, and ensure a clear understanding before actual development begins.

DFD DIAGRAM:

LEVEL 0

Fig.6.1.5 DFD Diagram

6.2 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : Identified classes of valid input must be accepted.

Invalid Input : Identified classes of invalid input must be rejected.

Functions : Identified functions must be exercised.

Output : Identified classes of application outputs must exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot ―see‖ into it. The test provides inputs and responds to outputs without considering how the software works.

6.2.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.The test objectives can be followed by,

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

The main goal of integration testing is to verify that different modules interact correctly and data flows seamlessly between them, ensuring the system works as intended when integrated. This testing helps identify issues related to interface mismatches, data communication errors, and inconsistent behavior across modules that may not be apparent during unit testing.

Integration testing can be performed using different approaches such as **top-down**, **bottom-up**, or **big bang** integration. It acts as a bridge between unit testing and system testing, improving overall software quality by catching defects early in the integration process before the system is fully deployed.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

Touchless mouse interaction based on hand gesture recognition is a forward-looking innovation that redefines the way users interact with computing systems. By eliminating the need for physical contact, it not only enhances user convenience but also addresses concerns related to hygiene and device wear-and-tear—making it especially valuable in public spaces, medical facilities, and cleanroom environments. The integration of computer vision and machine learning technologies allows for the real-time interpretation of hand movements, enabling the system to recognize gestures with increasing accuracy and responsiveness.

7.2 FUTURE ENHANCEMENT

Future enhancements for the virtual mouse system include support for more complex gestures, such as multi-finger movements and dynamic actions like swiping, zooming, and rotating, which would greatly expand its functionality and user interaction capabilities. Additionally, incorporating user personalization would allow the system to learn and adapt to individual differences in hand shapes and gesture styles, making it more accurate and user-friendly over time. To improve performance and accessibility, hardware optimization could be achieved by integrating affordable sensors like depth cameras or infrared devices, reducing reliance on high-end webcams. Finally, focusing on applications in accessibility tools can make the virtual mouse an invaluable alternative input method for users with physical disabilities, enhancing their ability to interact with digital devices more easily and comfortably.

APPENDIX A

(CODING)

```
import cv2
import mediapipe as mp
import numpy as np
import pyautogui
import time
# Initialize MediaPipe Hands
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
hands = mp_hands.Hands(
static_image_mode=False,
max_num_hands=1,
min_detection_confidence=0.7,
min_tracking_confidence=0.7
)
screen_width, screen_height = pyautogui.size()
cap = cv2.VideoCapture(0)
# To avoid multiple rapid clicks
click_delay = 1  # seconds
last_click_time = 0
def run_virtual_mouse():
global last_click_time
while cap.isOpened():
ret, frame = cap.read()
if not ret:
break
```

```
frame = cv2.flip(frame, 1)  # Mirror effect
rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
result = hands.process(rgb_frame)
if result.multi_hand_landmarks:
for hand_landmarks in result.multi_hand_landmarks:
mp_drawing.draw_landmarks(frame,hand_landmarks,mp_hands.HAND_CON
NECTIONS)
landmarks = [(lm.x, lm.y) for lm in hand_landmarks.landmark]
index_finger_up = landmarks[8][1] < landmarks[6][1]
middle_finger_up = landmarks[12][1] < landmarks[10][1]
cursor_x = int(landmarks[8][0] * screen_width)
cursor_y = int(landmarks[8][1] * screen_height)
if index_finger_up and not middle_finger_up:
pyautogui.moveTo(cursor_x, cursor_y)
elif index_finger_up and middle_finger_up:
current_time = time.time()
if current_time - last_click_time > click_delay:
pyautogui.click()
last_click_time = current_time
cv2.imshow("Virtual Mouse", frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
cap.release()
cv2.destroyAllWindows()
run_virtual_mouse()
```

APPENDIX B

(SCREENSHOTS)

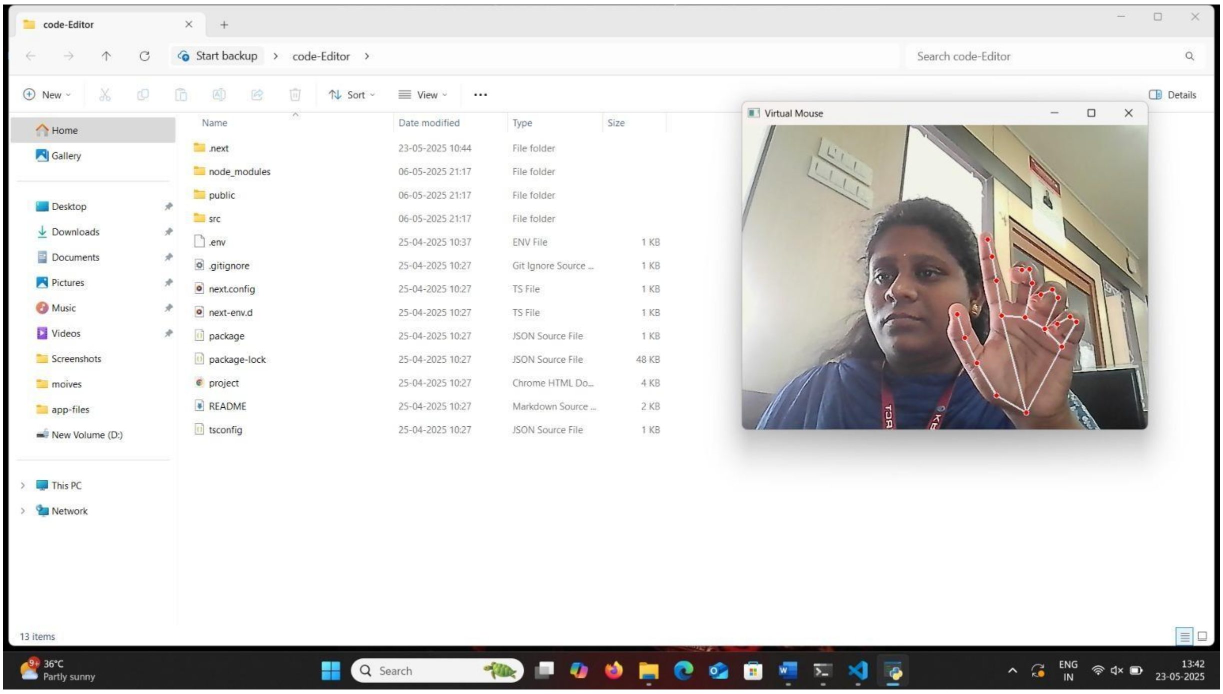

Fig. B2.1. Screenshot of the output

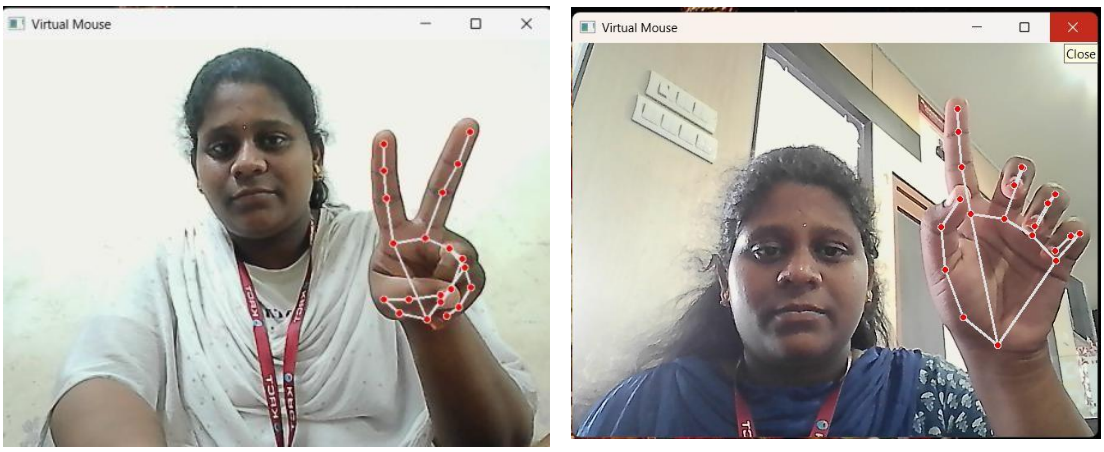

Fig.B2.2. Screenshot of the output

REFERENCES

1. Issam El Magrouni, Abdelaziz Ettaoufik, Siham Aouad, Abderrahim Maizate. "Hand Gesture Recognition for Virtual Mouse Control." International Journal of Interactive Mobile Technologies, Vol. 19, No. 2 (2025). EISSN: 1865-7923.

2. Rachana G, Pro Thouseef Ulla Khan. "Hand Gesture Control Virtual Mouse." International Journal of Advanced Research in Computer and Communication Engineering, Vol. 13, Issue 8, Aug 2024. DOI: 10.17148/IJARCCE.2024.13856.

3.SK. Jilani Basha, L. S. L. SOWMYA, SK. Rumiya, SK. Afroz. "Virtual Mouse with Hand Gestures Using AI." International Conference on Electronics and Sustainable Communication System (ICESC), IEEE (01 Aug 2023), DOI: 10.1109/ICESC57686.2023.10193709.

4.Asmoro, Jeffri Dian. Virtual mouse menggunakan Hand Gesture Recognition berbasis Hand Landmark Model untuk Pointing Device. Diss. UIN Sunan Ampel Surabaya, 2022.

5.Chua, S. N., et al. "Hand Gesture Control for Human–Computer Interaction with Deep Learning." Journal of Electrical Engineering & Technology (2022).

6.Agarwal, Prachi, et al. "Gesture Controlled Virtual Mouse." (2022).

7.Salvi, Mandar, et al. "Cursor Manipulation with Hand Recognition Using Computer Vision." INFORMATION TECHNOLOGY IN INDUSTRY 9.1 (2021).

8.Le, Nhat Vu, et al. "Hand Gesture Recognition System for Games." 2021 IEEE Asia-Pacific Conference on Computer Science IEEE, 2021.

9.Matlani, Roshnee, et al. "Virtual Mouse using Hand Gestures." 2021 International Conference on Technological Advancements and Innovations (ICTAI). IEEE, 2021.

10. Sarkar, Sankha, and India Sumanta Chatterjee. "A Vision Base Application for Virtual Mouse Interface Using Hand Gesture."