VIRTUAL MOUSE USING HAND GESTURE

A MINI PROJECT-II REPORT

Submitted By

NIRANJAN S	(621320205033)
DINESH KUMAR M	(621320205301)
RAHUL CR	(621320205045)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

ANNA UNIVERSITY: CHENNAI 600 025

May 2023

KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
NAMAKKAL- TRICHY MAIN ROAD, THOTTIAM, TRICHY.

COLLEGE VISION & MISSION STATEMENT

VISION

"To become an Internationally Renowned Institution in Technical Education, Research, and Development by Transforming the Students into Competent Professionals with Leadership Skills and Ethical Values."

MISSION

- ❖ Providing the Best Resources and Infrastructure.
- ❖ Creating Learner-Centric Environment and continuous Learning.
- ❖ Promoting Effective Links with Intellectuals and Industries.
- ❖ Enriching Employability and Entrepreneurial Skills.
- ❖ Adapting to Changes for Sustainable Development.

INFORMATION TECHNOLOGY

VISION:

To produce competent IT professionals, researchers and entrepreneurs with moral values in the field of Information Technology.

MISSION:

- ❖ Enrich the students' programming and computing skills through the best teaching-learning processes, laboratory practices, and through project-based learning.
- ❖ Inculcate real-world challenges, and emerging technologies and endeavor the students to become entrepreneurs or make them employable.
- ❖ Inculcating moral and ethical values to serve society and focus on students' overall development.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

- **PEO I:** Graduates shall become IT professionals with specialization in Software Engineering, Networking, Data Mining, and Cloud computing.
- **PEO II:** Graduates shall build IT solutions through analyzing, designing, and developing software and firmware solutions for real-world problems and social issues.
- **PEO III:** Graduates shall have professional ethics, team spirit, life-long learning, good oral and written communication skills, and adopt the corporate culture, core values, and leadership skills.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1: Professional skills:** Students shall understand, analyze and develop IT applications in the field of Data Mining/Analytics, Cloud Computing, Networking, etc., to meet the requirements of industry and society.
- **PSO2: Competency:** Students shall qualify at the State, National, and Intern the on-level competitive examination for employment, higher studies, and research.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identity, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reason informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
NAMAKKAL- TRICHY MAIN ROAD, THOTTIAM, TRICHY.

BONAFIDE CERTIFICATE

Certified that this project report titled "**VIRTUAL MOUSE USING HAND GESTURE** "is the bonafide work of **" NIRANJAN S (621320205033), DINESH KUMAR M (621320205301) and RAHUL C R (621320205045) "** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Mr. N. PREMKUMAR, M.E., (Ph.D.),	Dr. N. MAGENDIRAN, M.E., Ph.D.,
Associate Professor,	Professor,
Department of Information Technology,	Department of Information Technology,
Kongunadu College of Engineering and	Kongunadu College of Engineering and
Technology, Thottiam, Trichy- 621215	Technology, Thottiam, Trichy- 621215

Submitted for the Mini project -II Viva-Voce Examination held on.............

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to our beloved chairman **Dr.PSK.R.PERIASWAMY,** for providing the facilitation to make us extensively internationalized to meet global competence with almost perfection.

We would like to express our sincere thanks to our Principal **Dr.R.ASHOKAN, M.S.,M.Tech., Ph.D.,** for the facilities and the encouragement given to the progression and completion of this project.

We proudly render our immense gratitude to **Mr.N.PREMKUMAR, M.E.,(Ph.D).,** Associate Professor and Head of the Department, Department of Information Technology for his effective leadership, encouragement and supportive guidance to this project.

We proudly render our thanks to our Mini project-II Coordinator **Dr.N.MAGENDIRAN, M.E.,Ph.D..,** Professor, Department of Information Technology for his valuable ideas, encouragement and supportive guidance throughout this project.

We wish to extend our heartfelt regard and sincere thanks to our Mini project -II guide **Dr.N. MAGENDIRAN, M.E.,Ph.D..,** Professor, Department of Information Technology for his Motivation, Continuous encouragement and expert guidance throughout this project.

We wish to extend our sincere thanks to all faculty members of Information Technology Department for their valuable suggestions, kind cooperation and encouragement on successful completion of this project.

ABSTRACT

A virtual mouse using hand gestures to control the mouse cursor and perform tasks on the computer is known as a virtual mouse. This interface is designed to provide users, especially those with physical limitations or those who prefer a hand-free approach, a straightforward and natural method to interact with computers. The system uses a webcam to capture the user's hand movements and processes them using computer vision algorithms. By performing different hand gestures, such as moving the hand left or right, the user can control the movement of the cursor on the screen. The system also includes additional features, such as clicking and scrolling, which are activated by specific hand gestures. The computer performs the appropriate operations after processing the collected photos to identify the hand motions. Machine learning methods have been utilized by the system to recognize and learn various hand motions. NumPy, math, wax, and the mouse will be required to implement this system. The virtual mouse using hand gesture is a promising alternative to traditional mouse devices, particularly for individuals with disabilities or for those who want to reduce their physical contact with the computer. The capacity to regulate a computer using hand gestures.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	TITLE
GUI	**G**raphical **U**ser **I**nterface
HCI	**H**uman **C**omputer **I**nteraction
HSV	**H**ue **S**aturation and **V**alues
IDE	**I**ntegrated **D**evelopment **E**nvironment
LED	**L**ight **E**mitting **D**iodes
MHI	**M**otion **H**istory **I**mages
OpenCV	**O**pen **S**ource **C**omputer **V**ision
RGB	**R**ed, **B**lue, **G**reen

CHAPTER 1
INTRODUCTION

Virtual mouse using hand gesture is a pointing device that detects two-dimensional movements relative to a surface. This movement is converted into the movement of a pointer on a display that allows to control the Graphical User Interface (GUI) on a computer platform.

Virtual mouse is used a lot of different types of mouse that have already existed in the modern day's technology, there's the mechanical mouse that determines the movements by a hard rubber ball that rolls around as the mouse is moved. Years later, the optical mouse was introduced that replace the hard rubber ball to a LED sensor to detects table top movement and then sends off the information to the computer for processing. On the year 2004, the laser mouse was then introduced to improve the accuracy movement with the slightest hand movement, it overcome the limitations of the optical mouse which is the difficulties to track high-gloss surfaces.

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

Despite the limitations, the computer technology still continues to grow, so does the importance of the human computer interactions. Ever since the introduction of a mobile device that can be interact with touch screen technology, the world is starting to demand the same technology to be applied on every technological device, this includes the desktop system. However, even though the touch screen technology for the desktop system is already exist, the price can be very steep.

1.1 REVIEW OF THE PHYSICAL MOUSE

The physical mouse is known that there are various types of physical computer mouse in the modern technology, the following will discuss about the types and differences about the physical mouse.

MECHANICAL MOUSE

Mechanical mouse is known as the trackball mouse that is commonly used in the 1990s, the ball within the mouse are supported by two rotating rollers in order to detect the movement made by the ball itself. One roller detects the forward/backward motion while the other detects the left/right motion. The ball within the mouse are steel made that was covered with a layer of hard rubber, so that the detection is more precise.

The common functions included are the left/right buttons and a scroll-wheel. However, due to the constant friction made between the mouse ball and the rollers itself, the mouse is prone to degradation, as overtime usage may cause the rollers to degrade, thus causing it to unable to detect the motion properly, rendering it useless.

Furthermore, the switches in the mouse buttons are no different as well, as long-term usage may cause the mechanics within to be loosed and will no longer detect any mouse clicks till it was disassembled and repaired. The following **Figure 1.1** shows the mechanical mouse.

Figure 1.1 Mechanical mouse, with top cover removed

The following **Table1.1** describes the advantages and disadvantages of the Mechanical Mouse.

Table 1.1 Advantage and disadvantage of the Mechanical Mouse

ADVANTAGE	DISADVANTAGE
• Allows the users to control the computer system by moving the mouse. • Provides precise mouse tracking movements	• Prone to degradation of the mouse rollers and button switches, causing to be faulty. • Requires a flat surface to operate.

OPTICAL AND LASER MOUSE

A mouse that commonly used in these days, the motions of optical mouse rely on the Light Emitting Diodes (LEDs) to detect movements relative to the underlying surface, while the laser mouse is an optical mouse that uses coherent laser lights. Comparing to its predecessor, which is the mechanical mouse, the optical mouse no longer relays on the rollers to determine its movement, instead it uses an imaging array of photodiodes.

The purpose of implementing this is to eliminate the limitations of degradation that plagues the current predecessor, giving it more durability while offers better resolution and precision. However, there's still some downside, even-though the optical mouse is functional on most opaque diffuse surface, it's unable to detect motions on the polished surface. Furthermore, long term usage without a proper cleaning or maintenance may leads to dust particles trap between the LEDs, which will cause both optical and laser mouse having surface detection difficulties. Other than that, it's still prone to degradation of the button switches, which again will cause the mouse to function improperly unless it was disassembled and repaired. The following **Figure 1.2** shows the optical mouse.

Figure 1.2 Optical Mouse, with top cover removed

The following **Table1.2** describes the advantages and disadvantages of the Optical and Laser Mouse.

Table 1.2 Advantage and disadvantage of the Optical and Laser Mouse

ADVANTAGES	DISADVANTAGES
• Allows better precision with lesser hand movements. • Longer life-span.	• Prone to button switches degradation. • Does not function properly while on a polished surface.

1.2 PROBLEM STATEMENT

It's no surprised that every technological device has its own limitations, especially when it comes to computer devices. After the review of various type of the physical mouse, the problems are identified and generalized. The following describes the general problem that the current physical mouse suffers:

- Physical mouse is subjected to mechanical wear and tear.
- Physical mouse requires special hardware and surface to operate.
- Physical mouse is not easily adaptable to different environments and its performance varies depending on the environment.
- Mouse has limited functions even in present operational environments.
- All wired mouse and wireless mouse has its own lifespan.

1.3 MOTIVATION OF VIRTUAL MOUSE

It is fair to say that the Virtual Mouse will soon to be substituting the traditional physical mouse in the near future, as people are aiming towards the lifestyle where that every technological device can be controlled and interacted remotely without using any peripheral devices such as the remote, keyboards, etc. it doesn't just provide convenience, but it's cost effective as well.

CONVENIENT

In order to interact with the computer system, users are required to use an actual physical mouse, which also requires a certain area of surface to operate, not to mention that it suffers from cable length limitations. Virtual Mouse requires none of it, as it only a webcam to allow image capturing of user's hand position in order to determine the position of the pointers that the user want it to be.

The user will be able to remotely control and interact the computer system by just facing the webcam or any other image capturing devices and moving your fingers, thus eliminating the need to manually move the physical mouse, while able to interact with the computer system from few feet away.

COST EFFECTIVE

A quality physical mouse is normally cost from the range of 30 ringgit to a hefty 400 ringgit, depending on their functionality and features. Since the Virtual Mouse requires only a webcam, a physical mouse are no longer required, thus eliminating the need to purchase one, as a single webcam is sufficient enough to allow users to interact with the computer system through it, while some other portable computer system such as the laptop, are already supplied with a built-in webcam, could simply utilize the Virtual Mouse software without having any concerns about purchasing any external peripheral devices.

1.4 PROJECT SCOPE

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

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

The application can be started when the user's gesture was captured in real time by the webcam, which the captured image will be processed for segmentation to identify which pixels values equals to the values of the defined color.

The segmentation is completed, the overall image will be converted to Binary Image where the identified pixels will show as white, while the rest are black. The position of the white segment in the image will be recorded and set as the position of the mouse pointer, thus resulting in simulating the mouse pointer without using a physical computer mouse. The software application is compatible with the Windows platform. The functionality of the software will be coded with C++ programming language code with the integration of an external library that does the image processing known as the OpenCV.

1.5 PROJECT OBJECTIVE

The purpose of this project is to develop a Virtual Mouse application that targets a few aspects of significant development. For starters, this project aims to eliminate the needs of having a physical mouse while able to interact with the computer system through webcam by using various image processing techniques. Other than that, this project aims to develop a Virtual Mouse application that can be operational on all kind of surfaces and environment. The following describes the overall objectives of this project:

- To design to operate with the help of a webcam. The Virtual Mouse application will be operational with the help of a webcam, as the webcam are responsible to capture the images in real time. The application would not work if there are no webcam detected.
- To design a virtual input that can operate on all surface. The Virtual Mouse application will be operational on all surface and indoor environment, as long the users are facing the webcam while doing the motion gesture.

- To program the camera to continuously capturing the images, which the images will be analyzed, by using various image processing techniques.
- The Virtual Mouse application will be continuously capturing the images in real time, where the images will be undergoing a series of process, this includes HSV conversion, Binary Image conversion, salt and pepper noise filtering, and more.
- To convert hand gesture/motion into mouse input that will be set to a particular screen position.

The Virtual Mouse application will be programmed to detect the position of the defined colors where it will be set as the position of the mouse pointers. Furthermore, a combination of different colors may result in triggering different types of mouse events, such as the right/left clicks, scroll up/down, and more.

IMPACT, SIGNIFICANCE AND CONTRIBUTION

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

CHAPTER 2

LITERATURE REVIEW

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

2.1 VISUAL PANEL

To overcome the stated problems, Zheng you et al. (2001) [6], proposed an interface system named Visual Panel that utilize arbitrary quadrangle-shaped planar object as a panel to allow the user to use any tip-pointer tools to interact with the computer. The below **Figure2.1** shows the overview of visual panel.

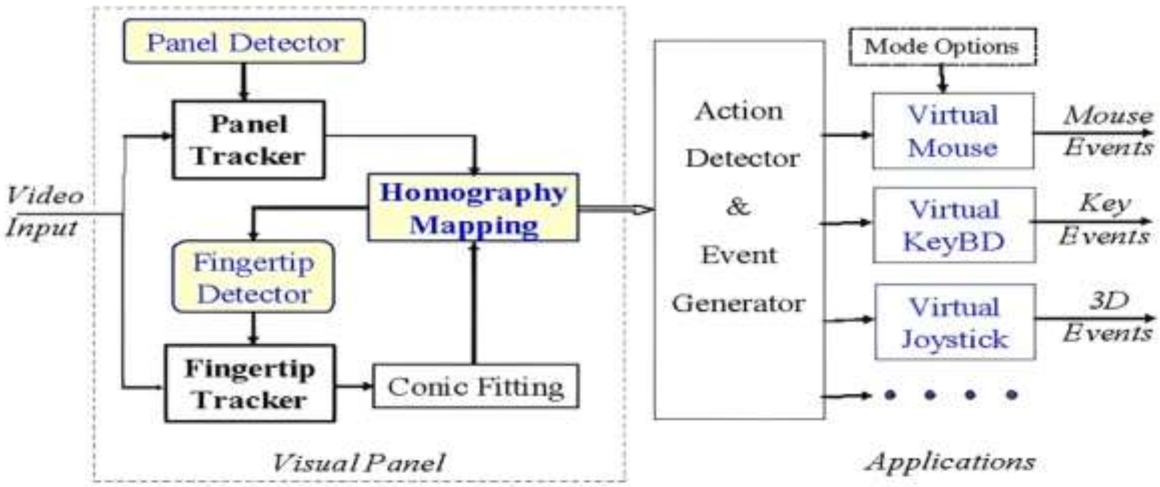


Figure 2.1 The system overview of Visual Panel

The interaction movements will be captured, analyzed and implement the positions of the tip-pointer, resulting accurate and robust interaction with the computer. The overall system consists of panel tracker, tip-pointer tracker, holography, calculation and update, and action detector and event generator as it can simulate both mouse and keyboard. However, it can still require a certain area and material to operate. Zheng you et al., have mentioned that the system can accepts any panel as long as it is quadrangle shaped, meaning any other shape besides stated shape are not allowed.

MOUSE SIMULATION USING COLOURED TAPES

Kamran Niyazi et al. (2012) [4], mentioned that to solve the stated problem, ubiquitous computing method is required. Thus, color tracking mouse simulation was proposed. The said system tracks two color tapes on the user fingers by utilizing the computer vision technology. One of the tapes will be used for controlling the movement of the cursor while the other will act as an agent to trigger the click events of the mouse. The below **Figure2.2** shows the architecture of mouse simulation.

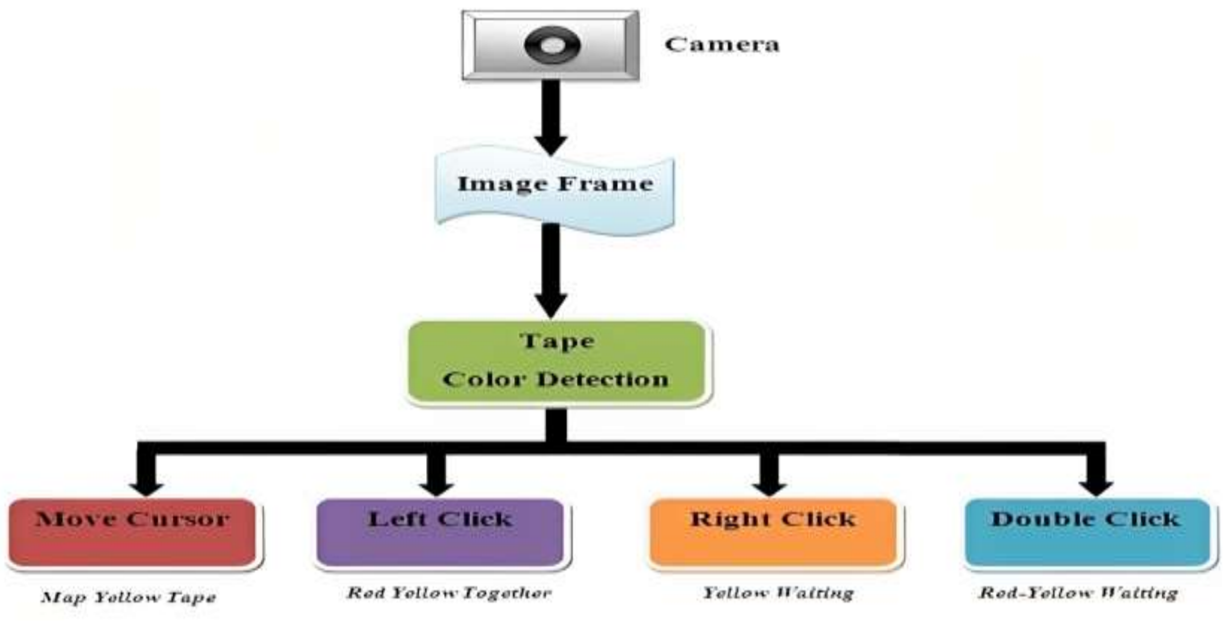


Figure 2.2 The system architecture of the mouse-simulation (Niyazi, 2012)

To detect the colors, the system is first required to process the captured image by separating the hand pixels from the non-hand pixels, which can be done by background subtraction scheme that segments the hands movement information from the non-changing background scene.

In order to implement this, the system requires to capture a pair of images to represent the static workplace from the camera view. When subtraction process is complete, the system will undergo another process that separates the RGB pixels to calculate the probability and differentiate the RGB values to determine which part are the skin and which are not.

The RGB pixels will be converted into HSV color plane in order to eliminate the variation in shades of similar color. The **Figure2.3** image will be converted to Binary Image and will undergo a filtering process to reduce the noise within the image.

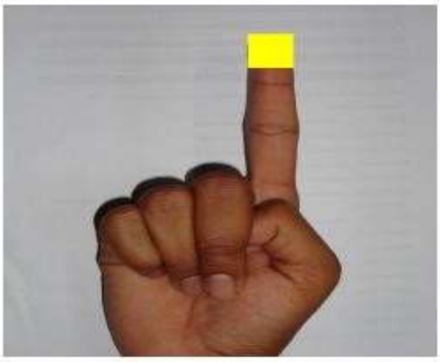


Figure 2.3 Yellow color tape for cursor movement (Niyazi, 2012)

Banerjee, A., Ghosh, A., Bharadwaj, K., & Saikia, H. (2014) [1]. Mouse control using a web camera based on color detection Even though the proposed system solved most of the stated issues, but there are limited functions offered by the proposed system as it merely able to perform common functions, such as: cursor movements, left/right click, and double clicks. While other functions, such as the middle click and mouse scroll were ignored.

VIRTUAL MOUSE USING A WEBCAM

Another color detection method proposed by Kazim Sekeroglu (2010) [5], the system requires three fingers with three color pointers to simulate the click events. The proposed system is capable of detecting the pointers by referring the defined color information, track the motion of the pointers, move the cursor according to the position of the pointer, and The **Figure2.4** can simulate the single and double left or/and right click event of the mouse.

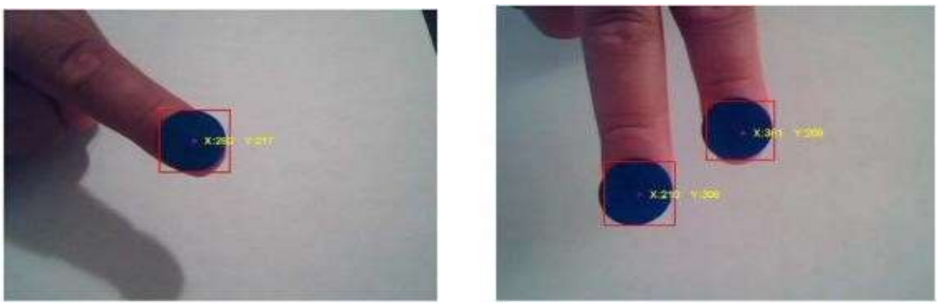


Figure 2.4 Input image using one and two pointers (Sekeroglu, 2010)

To detect the colors, they have utilized the MATLAB's built in "subtract" function, with the combination of the noise filtering by using median filter, which are effective in filtering out or at least reduce the "salt and pepper" noise.

The captured image will be converted to Binary Scale Image by using MATLAB's built in "im2bw" function to differentiate the possible values for each pixel. When the conversion is done, the captured image will undergo another filtering process by using "breakopen" to remove the small areas in order to get an accurate number of the object detected in the image.

2.2 PORTABLE VISION-BASED (HCI)

Another "Ubiquitous Computing" approach proposed by Chu-Feng Lien (2015) , requires only finger-tips to control the mouse cursor and click events[2].

The proposed system doesn't require hand-gestures nor color tracking in order to interact with the system, instead it utilizes a feature name Motion History Images (MHI), a method that used to identify movements with a row of images in time. The **Figure2.5** can be described as a flow chart of HCI.

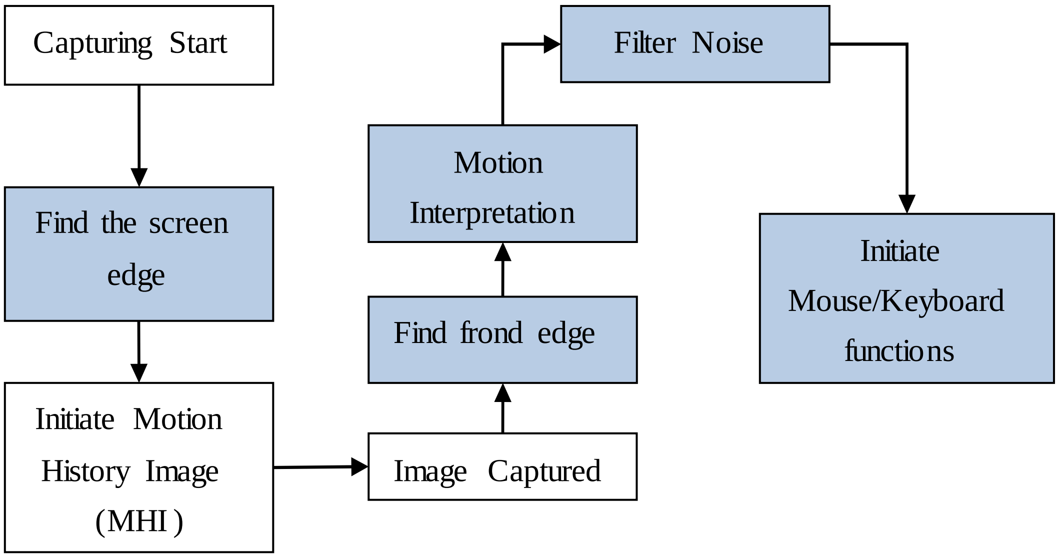


Figure 2.5 The Flow Chart of Portable Vision-Based (HCI)

Even though the proposed system process good accuracy in a well-controlled environment, it does have its own limitations. The proposed system is not capable to detect fast moving movements as the frame-rates are not able to keep up, thus leading to increase of error rate. Furthermore, due to the mouse click events occurred when the finger holds on a certain position, this may lead to user constant finger movements to prevent false alarm, which may result inconvenience. The following **Table 2.1** describe about the authors

TABLE 2.1 LITERATURE REVIEW

S.NO:	AUTHORS	ALGORITHM	YEAR	SUMMARY
1	Banerjee,A., Ghosh,A.,Bharadwaj,K.,Saikia,H.et .al.,	HAAR Cascades Algorithm	2014	Mouse control using a web cam based on color detection
2	Chu-Feng, L.et.al., Algorithm	Vision-Based HCI	2008	The system uses a portable camera for hands-free mouse control without the need for a surface
3	Park, H.et.al.,	HSV Algorithm	2008	A method for controlling mouse movement using a real-time camera
4	Niyazi,K.et.al.	Two Colored Tapes technique	2012	Mouse Simulation Using Two Colored Tapes
5	Sekeroglu,K.et.al.,	computer vision algorithm.	2010	system uses a webcam to track hand movements and move the mouse accordingly without the need for a surface.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system consists of the generic mouse and trackpad 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.

DISADVANTAGES

- Limited Accuracy
- Limited Gesture Recognition
- Limited Compatibility
- Learning Curve
- Physical Limitations

3.2 PROPOSED SYSTEM

Virtual Mouse using Hand Gesture system uses a camera-based approach to track hand gestures and movements for controlling a computer. The system uses a standard webcam to capture images of the user's hand and applies computer vision algorithms to detect and recognize various hand gestures. These gestures are then mapped to specific computer commands, allowing the user to control the computer with intuitive and natural movements. The proposed system aims to offer a user-friendly and accessible interface for controlling computers, with potential applications in gaming, design, and accessibility for users with disabilities. The system can be easily integrated into existing computer systems and software, making it a practical and cost-effective solution for gesture-based input.

ADVANTAGES

- Intuitive and User-Friendly
- Accessibility
- Flexibility
- Cost-Effective
- Potential for Applications in Different Fields.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

PROCESSOR: Intel dual Core, i5

RAM: 8 GB

HARD DISK: 256GB SSD

CAMERA: Web Camera with 720 mega pixels.

4.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM: Windows 11

LANGUAGE USED: Python 3.5.8

IDE: Visual studio code (VSC)

CHAPTER 5

DESIGN SYSTEM

5.1 SYSTEM ARCHITECTURE

Virtual Mouse using Hand Gesture system architecture work together to provide a seamless and efficient user experience for controlling the mouse using hand gestures in Python. The system processes the user's hand gestures, maps them to corresponding mouse actions, and provides visual feedback to the user, allowing them to interact with the computer naturally and easily as shown in **Figure5.1.**

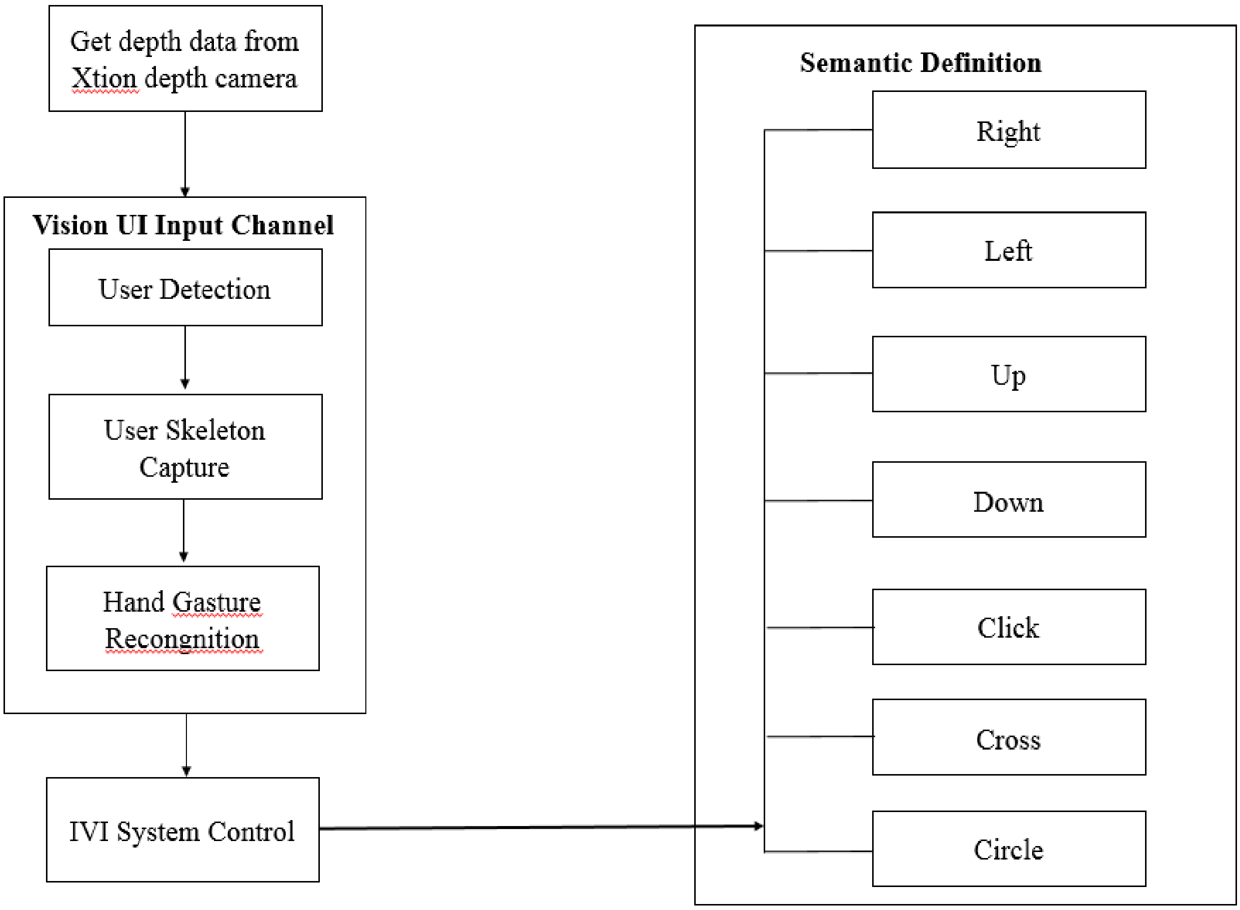


Figure 5.1 system architecture

5.1.1 IMAGE PROCESSING

The first component of the system is responsible for capturing video frames from a webcam or a depth sensor camera and performing image processing operations to detect and track hand gestures. It involves several sub-components such as image acquisition, hand detection, feature extraction, and gesture recognition. The image processing component analyzes the captured video frames to identify the user's hand and extract relevant features such as hand position, shape, and orientation.

5.1.2 GESTURE MAPPING

The second component of the system maps the detected hand gestures to corresponding mouse actions such as moving the cursor, clicking, and scrolling. It involves the creation of a mapping function that maps different hand gestures to specific mouse actions. The mapping function should be robust and accurate to ensure that the user's intended action is correctly interpreted.

5.1.3 USER INTERFACE

The third component of the system is responsible for providing a user-friendly interface that allows the user to control the mouse using hand gestures or voice commands. The user interface component displays the mouse cursor on the screen and provides visual feedback to the user regarding the detected hand gestures and mapped mouse actions. The user interface should be intuitive and easy to use to ensure that the user can control the mouse with ease and confidence.

CHAPTER 6

MODULE DISCRIPTION

HAAR Cascades are machine learning object detection algorithms that are used to identify faces in an image or a real-time video. The HAAR Cascade algorithm uses edge or line detection features that are proposed by Viola and Jones within their research paper named “Rapid Object Detection employing Boosted Cascade of Simple Features”.

6.1 HAAR CASCADE ALGORITHM

STEP 1: Importing OpenCV

STEP 2: Importing XML file

STEP 3: Importing test Image

STEP 4: Converting the image to greyscale

STEP 5: Detecting Multi-scale faces

STEP 6: Mentioning sides of the rectangle for face detection

STEP 7: Displaying the detected image.

6.2 MODULES

Here are some of the modules that may be included in a virtual mouse using hand gesture system:

- Camera or sensor array module
- Preprocessing module
- Gesture recognition module
- Virtual mouse control module
- User interface module
- Calibration module

6.3 MODULE DISCRIPTION

Camera or Sensor Array Module

Park, H. (2008). A method for controlling mouse movement using a real-time camera This module is responsible for capturing images of the user's hand movements and translating them into data that can be processed by the system [3]. It may consist of a single camera or multiple cameras/sensors, depending on the complexity of the system.

Preprocessing Module

The preprocessing module is responsible for preprocessing the images captured by the camera or sensor array to remove noise and enhance the relevant features of the hand movements. This is typically done using image processing techniques, such as filtering and segmentation.

Gesture Recognition Module

The Recognition module is responsible for recognizing the hand gestures made by the user and translating them into virtual mouse commands. It may use machine learning algorithms or rule-based systems to classify the gestures and generate the corresponding mouse commands.

Virtual Mouse Control Module

Virtual mouse module is responsible for translating the virtual mouse commands generated by the gesture recognition module into actions that can be understood by the computer or electronic device. This may involve emulating mouse movements and clicks, as well as keyboard inputs.

User Interface Module

Interface module provides the user with a visual interface that displays the virtual mouse movements and allows the user to interact with the system. It may also include settings for customizing the gesture recognition and virtual mouse control parameters.

Calibration Module

The calibration module allows the user to calibrate the system to their specific hand movements and preferences. This may involve performing a set of predefined gestures to train the system on the user's hand movements, or adjusting various settings to optimize the system's performance.

CHAPTER 7

SYSTEM IMPLEMENTATION

During the process of color recognition, it contains 2 major phases which are the calibration phase and recognition phase. The purpose of the calibration phase is to allow the system to recognize the Hue Saturation Values of the colors chosen by the users, where it will store the values and settings into text documents, which will be used later on during the recognition phase. While on the recognition phase, the system will start to capture frames and search for color input with based on the values that are recorded during the calibration phase. The phases of the virtual mouse are as shown in **figure7.1**.

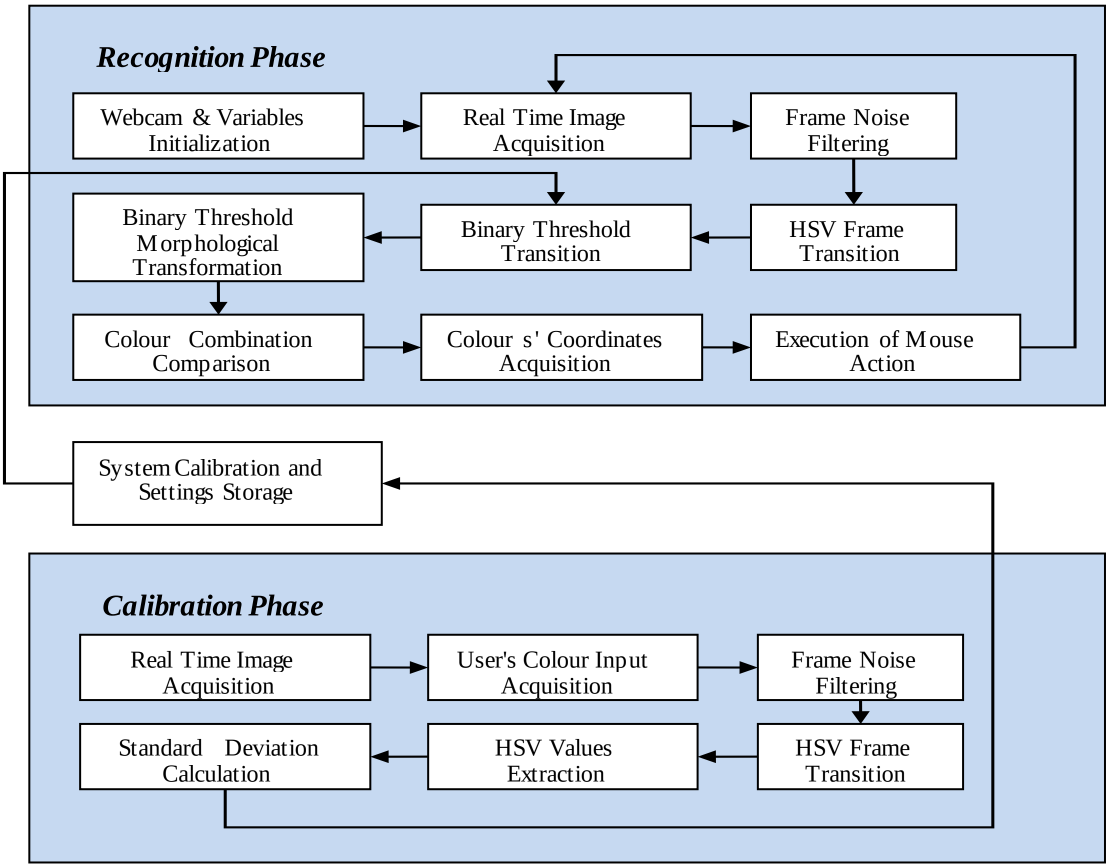


Figure 7.1 Virtual mouse block diagram

7.1 CALIBRATION PHASE

Real Time Image Acquisition

The program will start of by capturing real-time images via a webcam where it will a wait for users' color input. The size of the acquired image will be compressed to a reasonable size to reduce the processing loads of processing the pixels within the captured frame.

User's Color Input Acquisition

The program acquires the frames that consist of input colors submitted by the users, the captured frame will be sent for process where it will be undergone a series of transition and calculation to acquire the calibrated HSV values.

Frame Noise Filtering

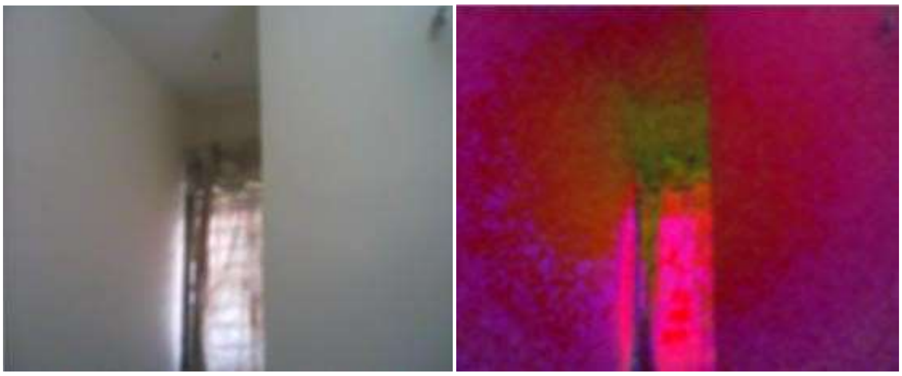
Every captured frame consists of noises that will affect the performance and the accuracy of the program, therefore the frame requires to be noise free. It can be applied on the captured frames to cancel out the unwanted noise. The **Figure7.2** will be used, which is a common smoothing method to eliminate noise in a frame. This can be done by using Gaussian Blur (Input Array src, Output Array dist., Size k size, double sigma X, double sigma Y=0, int border Type = BORDER_DEFAULT).

After Before

Figure 7.2 The comparison between un-filtered and filtered frame

HSV Frame Transition

The captured frame requires to be converted from a BGR format to a HSV format. Which can be done by using cvtColor (src, dist., CV_BGR2HSV). Here are some examples of HSV and RGB features below **Figure 7.3**.



Before After

Figure 7.3 The comparison between RGB and HSV frame

HSV Values Extraction

In order to acquire the HSV values, the converted frame require to be split into 3 single different planes, to do that the frame needs to be divided from a multi-channel array into a single channel array, which can be done by using **split** (const Mat& **src**, Mat* **mv begin**).

Standard Deviation Calculation

To obtain the maximum and the minimum of the HSV values, it requires to gone through the Standard Deviation calculation, a measurement used to quantify the amount of variation / dispersion among other HSV values. Furthermore, to obtain an accurate range of values, three-sigma rule are required in the calculation, so that chances of the captured values have a very high possibility to fall within the three-sigma intervals.

7.2 RECOGNITION PHASE

Webcam & Variables Initialization

On the early stage of the recognition phase, the program will initialize the required variables which will be used to hold different types of frames and values where each are will be used to carry out certain task. Furthermore, this is the part where the program collects the calibrated HSV values and settings where it will be used later during the transitions of Binary Threshold.

Real Time Image Acquisition

The real time image is captured by using the webcam by using (cv: Video Capture cap (0)), where every image captured are stored into a frame variable (cv:mat), which will be flipped and compressed to a reasonable size to reduce process load.

Frame Noise Filtering

Similar to the noise filtering during the calibration phase, Gaussian filters will be applied to reduce the existing noise of the captured frames. This can be done by using **Gaussian Blur** (Input Array **src**, Output Array **dist.**, Size **k size**, double **sigma X**, double **sigma Y**=0, int **border Type**=BORDER_DEFAULT).

HSV Frame Transition

The captured frame requires to be converted from a BGR format to a HSV format. Which can be done by using cvt Color (src, dst, CV_BGR2HSV).

Binary Threshold Transition

The converted HSV frame will undergone a range check to check if the HSV values of the converted frame lies between the values of the HSV variables.

The result of the range check will convert the frame into a Binary Threshold, where a part of the frame will set to 255 (1 bit) if the said frame lies within the specified HSV values, the frame will set to 0 (0 bit) if otherwise.

Binary Threshold Morphological Transformation

After the binary threshold is obtained, the frame will undergone a process called Morphological Transformation, which is a structuring operation to eliminate any holes and small object lurking around the foreground. The transformation consists of two morphological operators, known as Erosion and Dilation.

The Erosion operator are responsible for eroding the boundaries of the foreground object, decreasing the region of the binary threshold, which is useful for removing small noises. As for Dilation, it is an opposite of erosion, it increases the region of the binary threshold, allowing eroded object to return to its original form. For the current project, both operators were used for morphological Opening and Closing, where Opening consists of combination of erosion followed by dilation, which is very useful in removing noise, whereas Closing is the opposite of Opening, which is useful in closing small holes inside the foreground object.

Color Combination Comparison

After obtaining results from Morphological Transformation process, the program will calculate the remaining number of objects by highlighting it as blobs, this process requires cv Blob library, which is an add-on to OpenCV. The results of the calculation will then send for comparison to determine the mouse functions based on the color combinations found within the captured frames.

Colors Coordinates Acquisition

For every object within the binary threshold, the program will highlight the overall shape of the object (cvRender Blobs(const Ipl Image *img Label, CvBlobs &blobs, Ipl Image *img Source, IplI mage *img Dest, unsigned short mode=0x000f, double alpha=1.);), where it will calculate the area of the shape and the coordinates of midpoint of the shapes. The coordinates will be saved and used later in either setting cursor positions, or to calculate the distance between two points to execute various mouse functions based on the result collected.

Execution of Mouse Action

The program will execute the mouse actions based on the colors combinations exist in the processed frame. The mouse actions will perform according to the coordinates provided by the program, and the program will continue on acquire and process the next real-time image until the users exit from the program.

CHAPTER 8

RESULT AND DISCUSSION

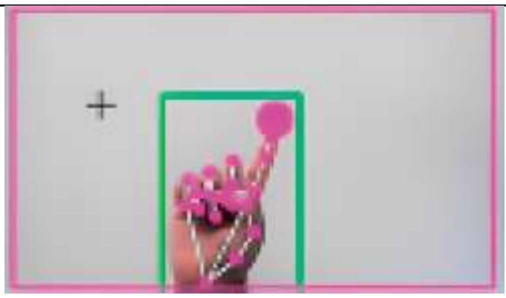
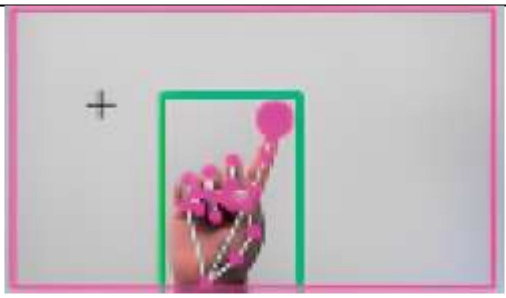
In order to achieve accuracy, and consistency of the Virtual Mouse color recognition, testing phase have been conducted on various scenarios.

8.1 VIRTUAL SCREEN MATCHING

An AI virtual mouse system employs the transform algorithm rule to convert coordinates from the top of the digital camera screen to the full-screen PC window and control the mouse. As illustrated, a rectangular box is created in relation to the computer window within the camera area, where we may travel using the mouse indicator once the hands are identified and we confirm that the appropriate finger is present to do the necessary mouse action. Detecting that the finger is up and doing the designated mouse operation we can confirm that Media Pipe is up right now by looking at the tip Id Victimization of the specific finger that we set victimization on and the associated co-ordinates of the arms that are up. Then the appropriate mouse action is used in accordance with the resulting determination.

- To move the mouse indication inside the computer window. If the finger is up with tip Id = 1 or both the finger and tip are up with tip Id = 0 and a couple, the Python AutoPy module is trained to move the mouse pointer across the computer window.
- The mouse's left button to be clicked. As demonstrated in the pyinput Python module is used to instruct the computer to click the left button if the thumb finger with tip ID = 0 and the finger with tip ID = 1 are both up and the distance between the 2 fingers is less than 30px. The following **Table8.1** describes the color combination to trigger and execute the mouse functions.

Table: 8.1 The overall color combination for specific Mouse Functions

	Capturing video exploitation, the digital camera.
	Shows a rectangular box for the computer screen's cursor-moving area.
	Identifying which finger is up.
	Shows the mouse pointer gliding over the computer screen.
	Shows a motion that guiding a computer to click left.

	Shows a motion that instructs a computer to click left.
	A motion that instructs the computer to scroll up.
	A gesture that causes the computer to scroll down.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The virtual mouse using hand gesture technology has become increasingly popular as it provides an alternative way of controlling devices, improved accessibility, convenience, and hygiene.

However, there are some limitations to this technology such as lower precision and requiring more physical effort from the user, and it may not be suitable for all types of applications or users.

Overall, the virtual mouse using hand gesture technology has the potential to be a valuable tool for many users, particularly those who have difficulty using traditional input devices.

FUTURE ENHANCEMENT

- Multi-user support refers to the ability of a technology to allow multiple users to interact with a device simultaneously. This can enable collaborative work and gaming among multiple users.

APPENDICES

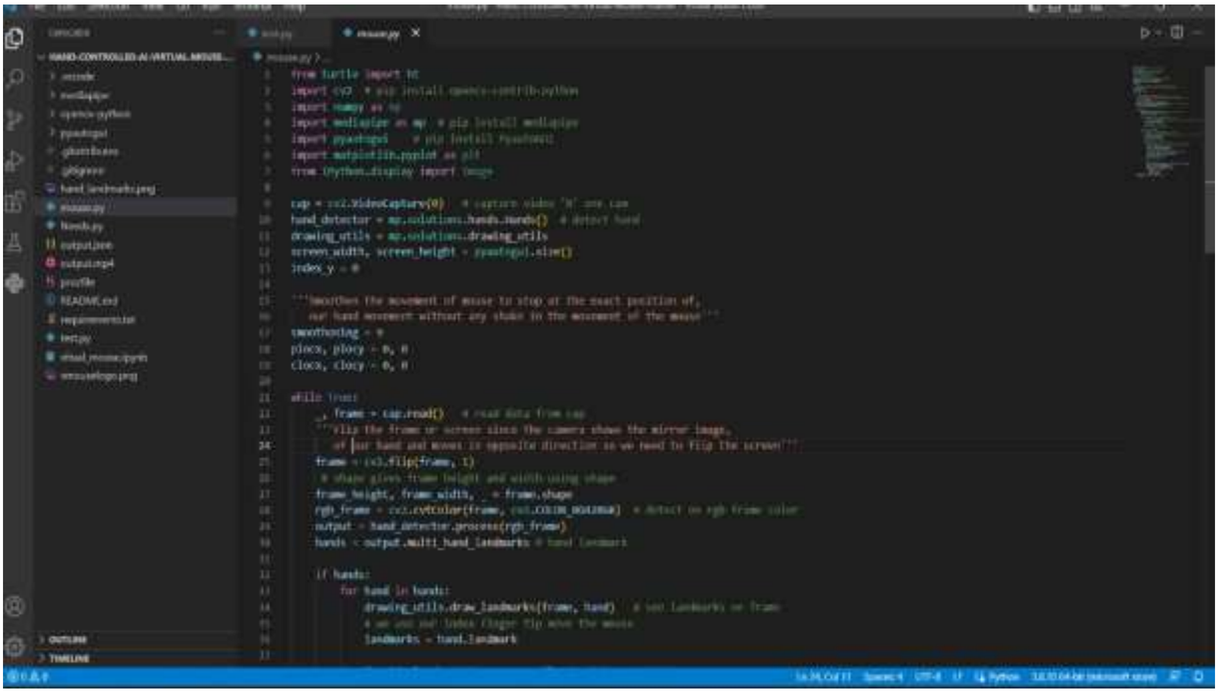
SAMPLE CODE

```
import eel
import os
from queue import Queue
class ChatBot:
started = False
userinputQueue = Queue()
def isUserInput():
return not ChatBot.userinputQueue.empty()
def popUserInput():
return ChatBot.userinputQueue.get()
def close_callback(route, websockets):
# if not websockets:
#     print('Bye!')
exit()
@eel.expose
def getUserInput(msg):
ChatBot.userinputQueue.put(msg)
print(msg)
def close():
ChatBot.started = False
def addUserMsg(msg):
eel.addUserMsg(msg)
def addAppMsg(msg):
eel.addAppMsg(msg)
def start():
path = os.path.dirname(os.path.abspath(__file__))
eel.init(path  + r'\web', allowed_extensions=['.js', '.html'])
try:
```

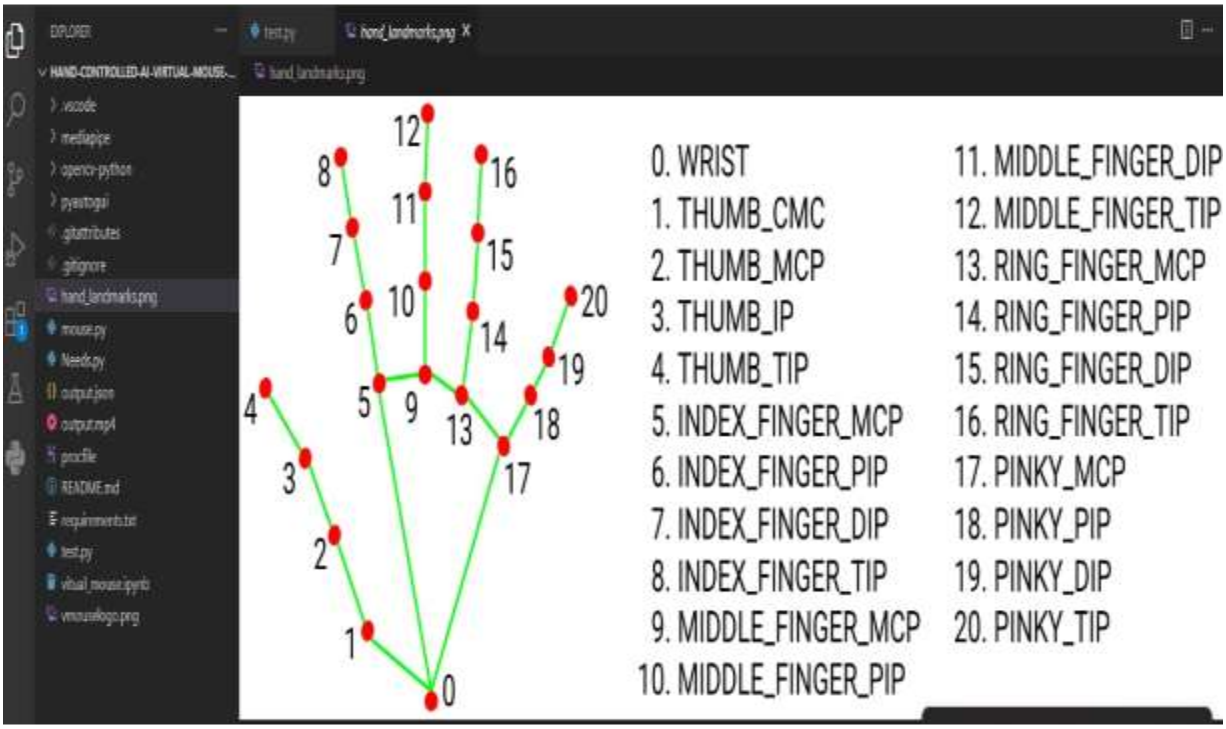
```
eel.start('index.html', mode='chrome',
host='localhost',
port=27005,
block=False,
size=(350, 480),
position=(10,100),
disable_cache=True,
close_callback=ChatBot.close_callback)
ChatBot.started = True
while ChatBot.started:
try:
eel.sleep(10.0)
except:
#main thread exited
break
except:
pass
# Imports
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
Enum for mapping all hand gesture to binary number.
Represent gesture corresponding to Enum 'Gest',
stores gesture being used.
prev_gesture : int
Represent gesture corresponding to Enum 'Gest',
stores gesture computed for previous frame.
frame_count : int
total no. of frames since 'ori_gesture' is updated.
hand_result : Object
sbcontrol.fade_brightness(int(100*currentBrightnessLv) , start =
sbcontrol.get_brightness(display=0))
def changesystemvolume():
volume = cast(interface, POINTER(IAudioEndpointVolume))
currentVolumeLv = volume.GetMasterVolumeLevelScalar()
currentVolumeLv += Controller.pinchlv/50.0
def scrollVertical():
else:
Controller.prev_hand = None
cv2.imshow('Gesture Controller', image)
if cv2.waitKey(5) & 0xFF == 13:
break
GestureController.cap.release()
cv2.destroyAllWindows()
# uncomment to run directly
# gc1 = GestureController()
# gc1.start()
```

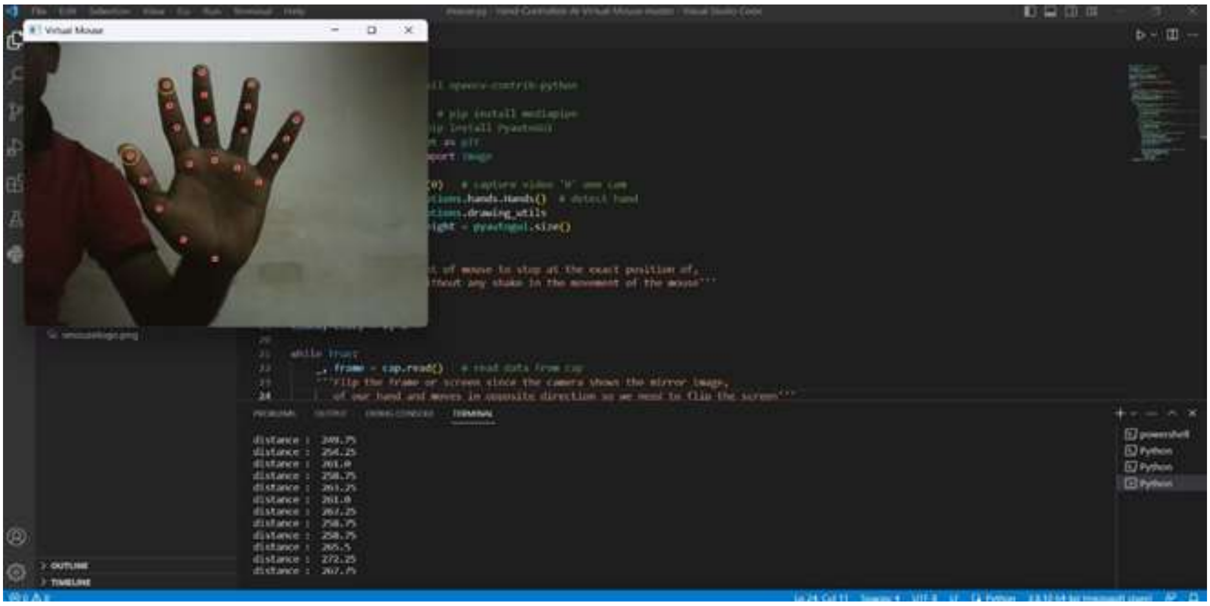
SAMPLE SCREENSHORT



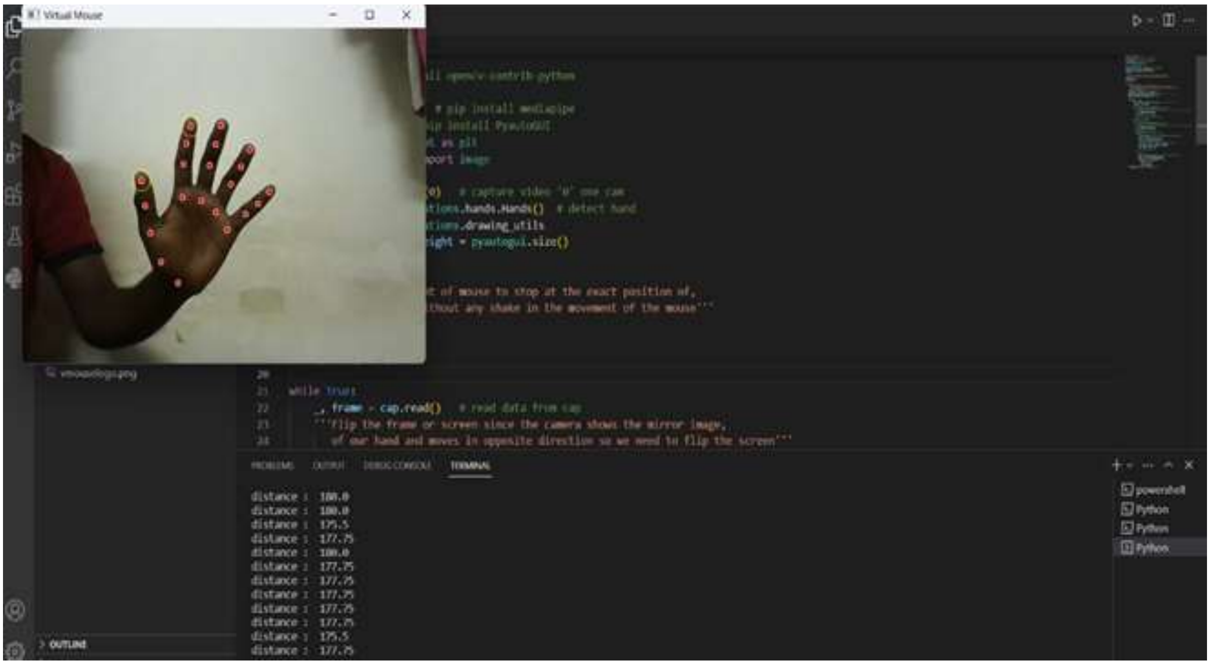
HAND POSE METHOD



HAND LANDMARKS



CAPTURING THE HAND



MOVING THE MOUSE CURSOR

REFERENCES

[1] Banerjee, A., Ghosh, A., Bharadwaj, K., & Saikia, H.et.al., (2014). Mouse control using a web camera based on color detection. arXiv preprint arXiv:1403.4722.

[2] Chu-Feng, L.et.al., (2008). Portable Vision-Based HCI. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38(6), 1337-1348.

[3] Park, H.et.al., (2008). A method for controlling mouse movement using a real-time camera. In Proceedings of the 2008 ACM symposium on Applied computing (pp. 1329-1330).

[4] Niyazi, K.et.al., (2012). Mouse Simulation Using Two Colored Tapes. International Journal of Information Science and Technology, 2(2), 57-60.

[5] Sekeroglu, K.et.al (2010). Virtual Mouse Using a Webcam. In Proceedings of the 2010 IEEE 18th Signal Processing and Communications Applications Conference (pp. 939-942).

[6] Zhengyou, Z., Ying, W. and Shafer, S.et.al., (2001). Visual Panel: Virtual Mouse, Keyboard and 3D Controller with an Ordinary Piece of Paper. In Proceedings of the 2001 IEEE Symposium on Visual Languages and Human-Centric Computing (pp. 171-172).