****

**MINI-PROJECT**

**On**

**“Virtual mouse control using hand gesture”**

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

**Submitted by**

**Anitha P (1DS19IS017)**

**Avani K S (1DS19IS025)**

**Gowthami Holla (1DS19IS042)**

**Kruthi R Aithal (1DS19IS052)**

**Under the guidance of**

**Mr. Suresh Kumar M**

**Asst Professor**

**Department Of Information Science and Engineering**

**DAYANANDA SAGAR COLLEGE OF ENGINEERING**

S M Hills, Kumara Swamy Layout, Bengaluru-560078

**2021-22**

**CONTENTS**

Chapter 1 Introduction….…………………………………….……Pg No.

* 1. Overview
  2. Problem statement
  3. Objectives and Scope of Project
  4. Motivation of Project

Chapter 2 Literature Survey…………………………………………. Pg No.

Chapter 3 Requirements ……………………………………….……Pg No.

3.1 Software Requirements

3.2 Hardware Requirements

Chapter 4 System Analysis …………………………………………. Pg No.

4.1 Existing system

4.2 Proposed system

Chapter 5 System Design.…………………………………………. Pg No.

5.1 Introduction

5.2 Architecture diagram

5.3 Use case diagram

5.4 Flow chart diagram

5.5 Sequence diagram

Chapter 6 Modules ……………….…………………………………. Pg No.

6.1 Modules

6.2 Module description

Chapter 7 Pseudocode ………………….……………………………. Pg No.

Chapter 8 Testing ……………………………………………………. Pg No.

8.1 Basics of software testing

8.2 Test cases

Chapter 9 Results …………………………….………………………. Pg No.

Chapter 10 Conclusion and Future Scope………………….…………. Pg No.

References …………………….………………………………………. Pg No.

**ABSTRACT**

This project proposes a way to control the position of the cursor with the bare hands without using any electronic device. The hand gesture is the most effortless and natural way of communication. While the operations like clicking and dragging of objects will be performed with different hand gestures. Our method is to use a camera and computer vision technology, such as image segmentation, background subtraction and color tracking, to control mouse tasks (left clicking, right clicking, double-clicking and scrolling actions) and we show how it can perform everything as current mouse devices can. A color pointer has been used for the object recognition and tracking, so as to implement the module without any physical contact with the system. Click events of the mouse have been achieved by detecting the number of pointers on the images.

The proposed system will only require a webcam as an input device. The software that will be required to implement the proposed system are OpenCV and python. The output of the camera will be displayed on the system’s screen so that it can be further calibrated by the user. A low-resolution webcam is used which acts as a sensor and is able to track the users hand bearing color caps in 2 dimensions. The python dependencies that will be used for implementing this system are NumPy, math, wx and mouse.

**Chapter 1**

**INTRODUCTION**

**1.1 Overview**

It has been generations since we have been using hand gestures for communicating in human society. The shaking of hands, Thumbs up and Thumbs down signs have been ever existing in the environment. It is believed that gestures are the easiest way of interaction with anyone. So then why not apply it to the machines that we are using. In this work, we are demonstrating, real- gesture. The initial setup includes a low-cost USB web camera that can be used for providing the input to the system. The complete process is divided into 4 steps which are frame-capturing, image-processing, region-extraction, feature-matching.

This project proposes an AI virtual mouse system that makes use of the hand gestures and hand tip detection for performing mouse functions in the computer using computer vision. With the use of the AI virtual mouse system, we can track the fingertip of the hand gesture by using a built-in camera or web camera and perform the mouse cursor operations and scrolling function and also move the cursor with it.

The project covers as a hand recognition tool which could be used to move the mouse pointer, perform simple operations like clicking and other hand gesture operations like moving file from computer to computer through delicate socket programming and performing simple but fascinating operations that could be covered with the hand recognition.

**1.2 Problem Statement**

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.

**1.3 Objectives and Scope of Project**

1.3.1 *Objectives*

* The 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.
* For most laptop touchpad is not the most comfortable and convenient.
* Main objective pre-processing is to represent the data in such a way that it can be easily interpreted and processed by the system.
* Reduced cost of hardware.
* It focuses on extracting the features over the human hands and then matching their features to recognize the movement of the hand.

1.3.2 *Scope of the Project*

* Gesture based interfaces that are used for computer applications are more intuitive than established WIMP based interfaces, allowing the users to interact with computer applications.
* Older population can be benefitted by the use of gestures which can achieve easier interaction with the electronic devices.
* The quick development of hand gesture recognition system leads to substantial improvement of many different applications. For example, in Medicine, Doctors and surgeons can use it to visualize each patient scans more clearly, online learning and teaching via distance techniques are adopted and television sets can also be controlled. Among these, the most important application is the sign language interpretation for the people having impaired hearing.

**1.4 Motivation of Project**

* Gesture recognition can be seen as a way for computers to begin to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces or even GUIs which still limit the majority of input to mouse.
* The aim of this project is to enable users to interact more naturally with their computer by simple hand gestures to move the mouse and perform tasks.
* One of the major targets in building this project is considering the old and also left-handed people.
* Anyone with a computer and a camera should be able to take full advantage of this project.

**Chapter 2**

**LITERATURE SURVEY**

Several related works have been carried out on virtual mouse using glove-wearing hand gesture recognition, and using color tips in the hands to recognize gestures, but they aren't more accurate.

1. S. Shriram , B. Nagaraj , J. Jaya , S. Shankar and P. Ajay published a research article “Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread” , on 25 October 2021, that overcomes the limitation of battery for power and a dongle to connect the mouse to a PC, by employing webcam or a built-in camera for capturing of hand gestures and hand tip detection using computer vision. The above paper has been used as the main reference paper for this project.
2. D. L. Quam, “Gesture recognition with a DataGlove,” IEEE Conference on Aerospace and Electronics, vol. 2, pp. 755–760, 1990. View at: Publisher Site Google Scholar. An attempt has been made to detect hand gestures using a camera. Quam introduced a hardware-based system in this system, the user wears a DataGlove .
3. Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil proposed a study in “Cursor control System Using Hand Gesture Recognition.” in 2013. In this work, the limitation is stored frames are needed to be processed for hand segmentation and skin pixel detection.

<https://www.researchgate.net/profile/Nilesh_Uke/publication/239525791_Cursor_Control_System_Using_Hand_Gesture_Recognition/data/00b4951c14ad7ef5b0000000/Cursor-Control-System-Using-Hand-Gesture-Recognition.docx>

1. Vinay Kr. Pasi , Saurabh Singh, and Pooja Kumari proposed “Cursor control using Hand Gestures” in the IJCA Journal in 2016 . The system proposes the different bands to perform different functions of the mouse.

The limitation is it depends on various colors to perform mouse functions.

[https://www.irjmets.com/uploadedfiles/paper//issue\_4\_april\_2022/21879/final/fin\_irjmets1651395700.pdf](https://www.irjmets.com/uploadedfiles/paper/issue_4_april_2022/21879/final/fin_irjmets1651395700.pdf)

1. GESTURE Abhilash S S1, Lisho Thomas2, Naveen Wilson3, Chaithanya C4 “VIRTUAL MOUSE USING HAND” by Students Of Dept of computer science and engineering Mar Athanasius College of Engineering , December 2020.

**Chapter 3**

**REQUIREMENTS**

**3.1 Software Requirements**

* OS Required : Microsoft DOS, Microsoft Windows 3.1 or later , PC DOS
* OS Family : Windows
* Visual studio for workspace
* Language: Python
* Libraries: Open cv, Mediapipe, Numpy, Autogain / Pynput

**3.2 Hardware Requirements**

* Min RAM Size : 4MB
* Min Hard Drive Space : 25MB
* Min processor Type : Intel 386 or higher
* Web cam or built in camera

**Chapter 4**

**SYSTEM ANALYSIS**

* 1. **Existing System**

The existing system consists of a mouse that can be either wireless or wired to control the cursor, know we can use hand gestures to monitoring the system. The existing virtual mouse control system consists of the simple mouse operation using the colored tips for detection which are captured by web-cam, hence colored fingers acts as an object which the web-cam sense color like red, green, blue color to monitor the system, whereas could perform basic mouse operation like minimize, drag, scroll up , scroll down , left-click right-click using hand gestures without any colored finger because skin color recognition system is more flexible than the existing system.

In the existing system use static hand recognition like fingertip identification, hand shape, Number of fingers to defined action explicitly , which makes a system more complex to understand and difficult to use.

**4.2 Proposed System**

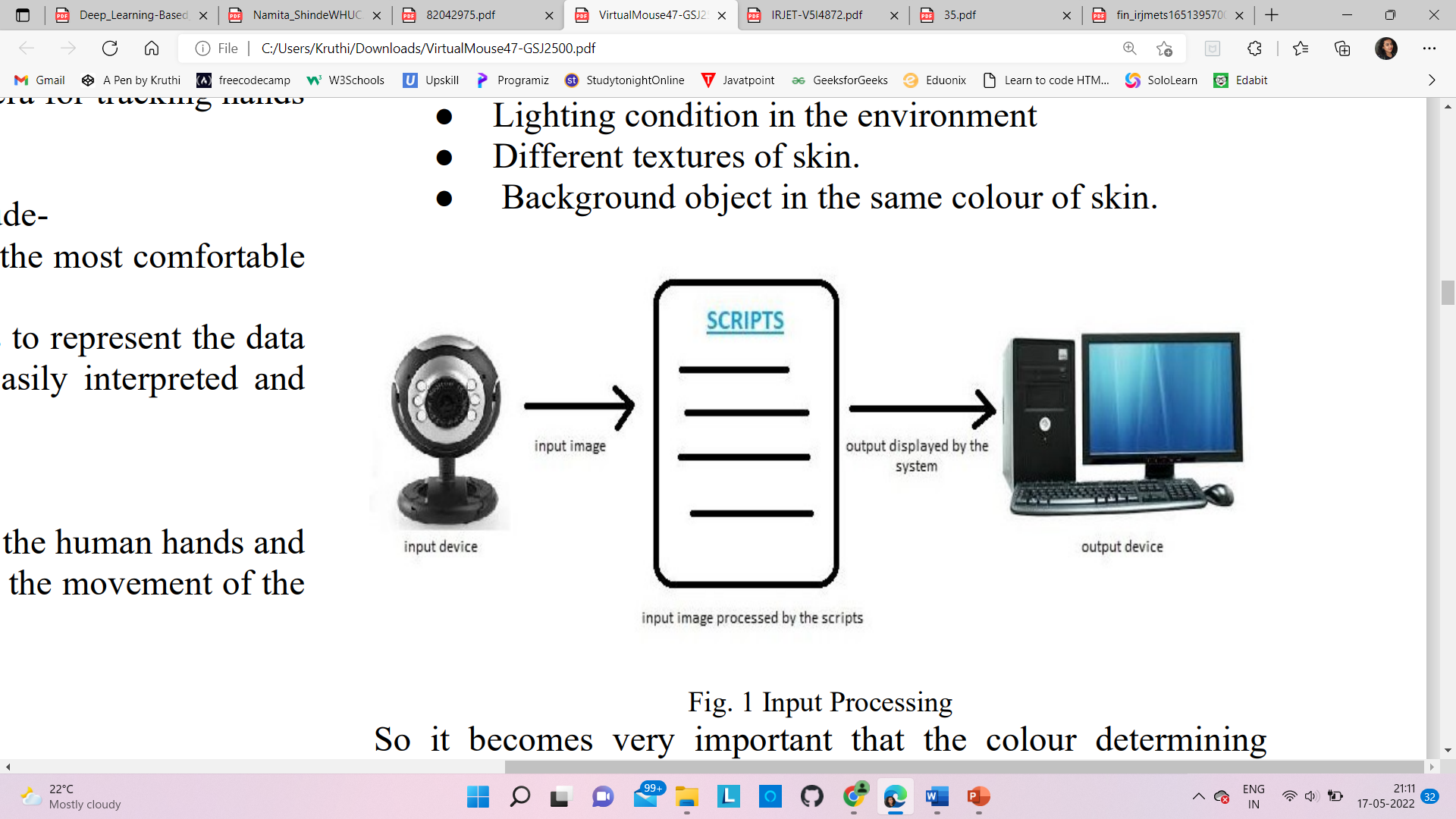
The system works by identifying the color of the hand and decides the position of the cursor accordingly but there are different conditions and scenario which make it difficult for the algorithm to run in the real environment due to the following reason as shown in Fig. 1 :

● Noises in the environment.

● Lighting condition in the environment

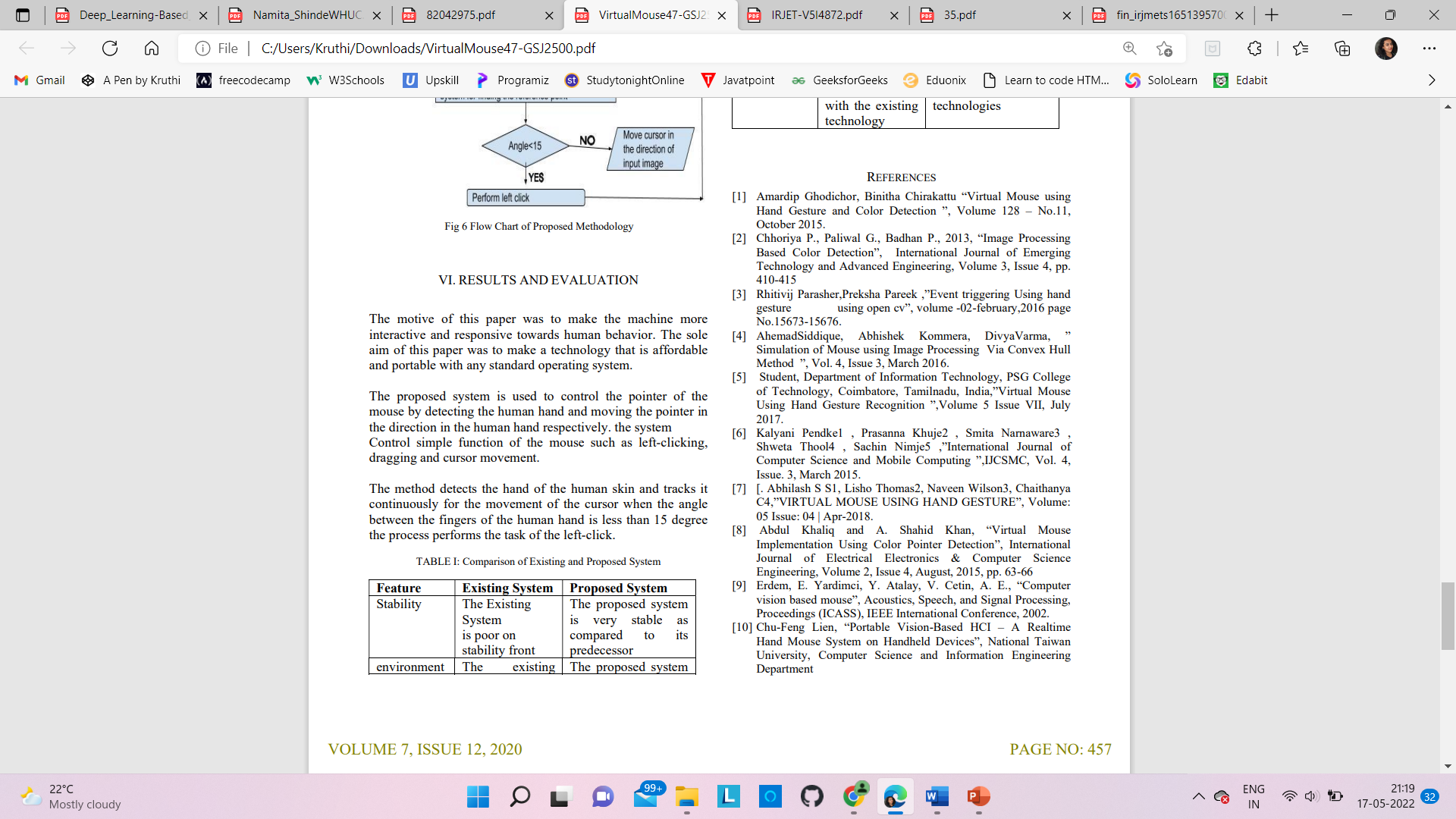
● Different textures of skin.

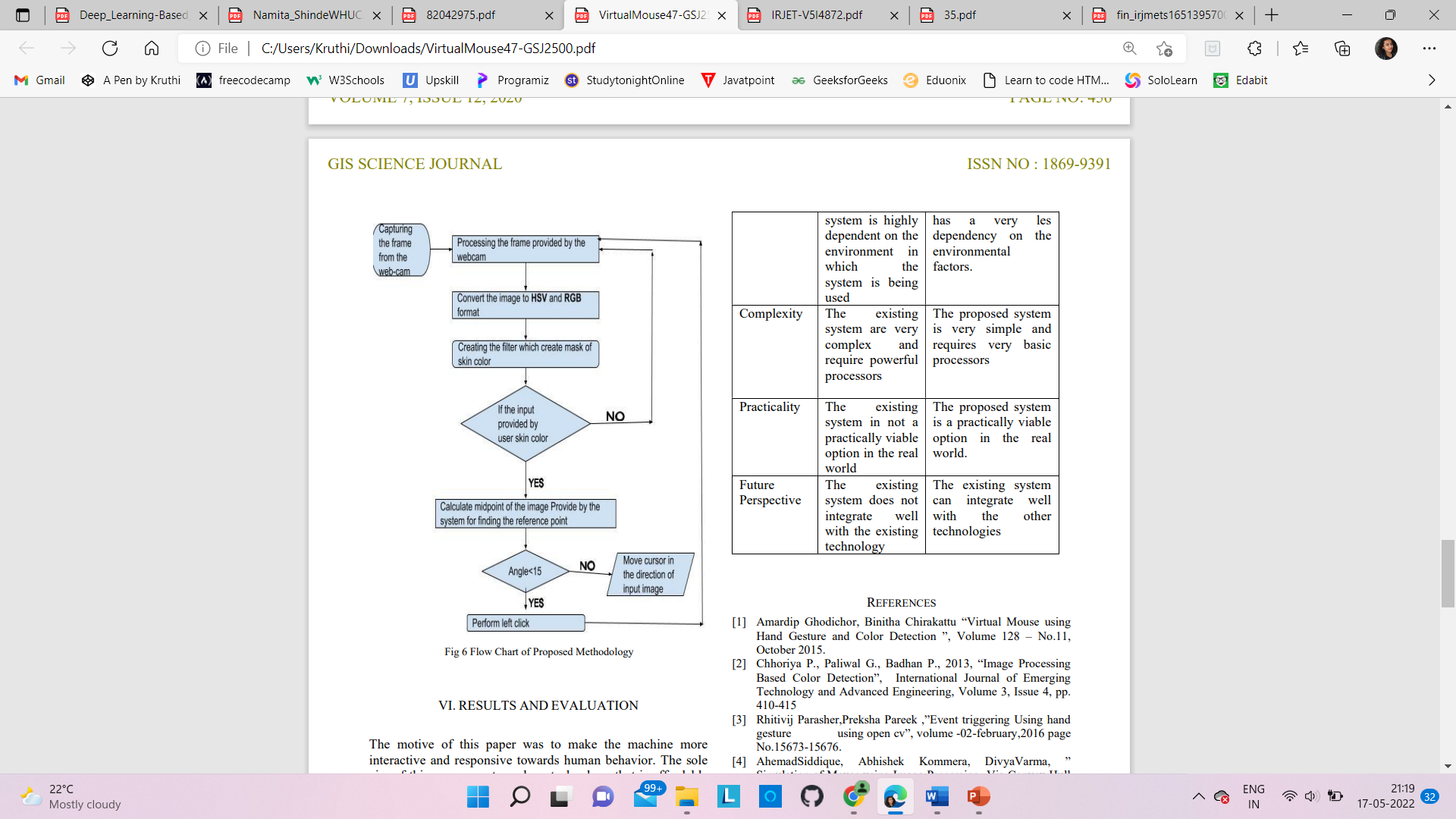
● Background object in the same color of skin.



So, it becomes very important that the color determining algorithm works accurately. The proposed system can work for the skin tone of any color as well as can work accurately in any lighting condition as well for the purpose of clicking the user needs to create a 15 degree angle between its two-finger the proposed system can easily replace the traditional mouse as well as the algorithm that requires colored tapes for controlling the mouse . The project can be developed with “zero-cost” and can easily integrate with the existing system.

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.





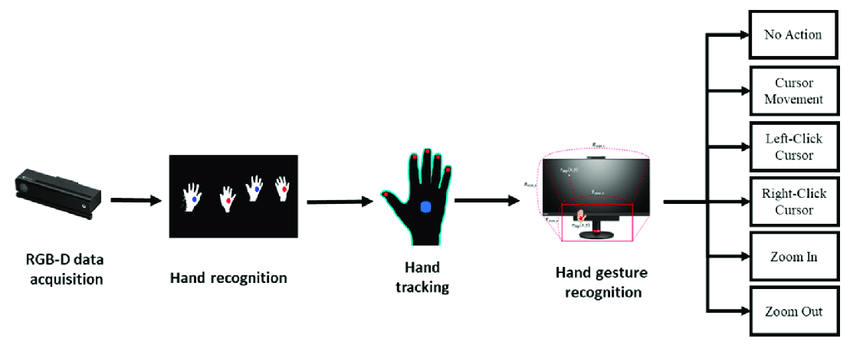
**Chapter 5**

**SYSTEM DESIGN**

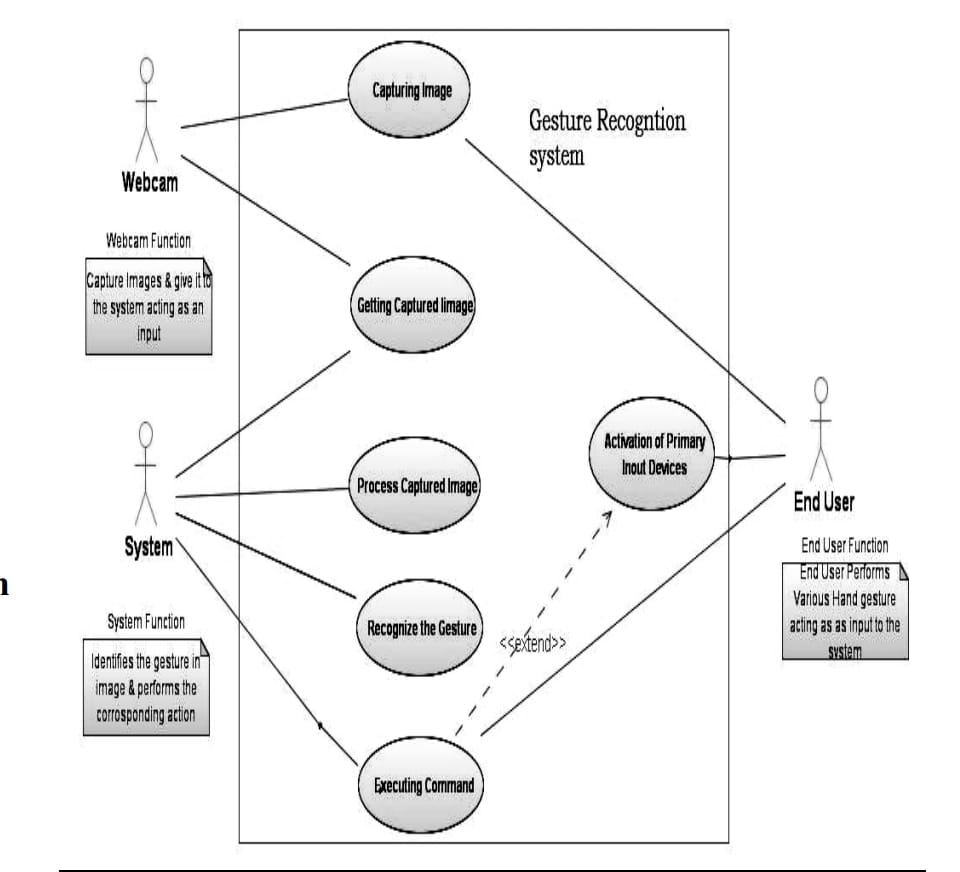
**5.1 Introduction**

During the process of hand recognition, it contains a major phase which is 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.

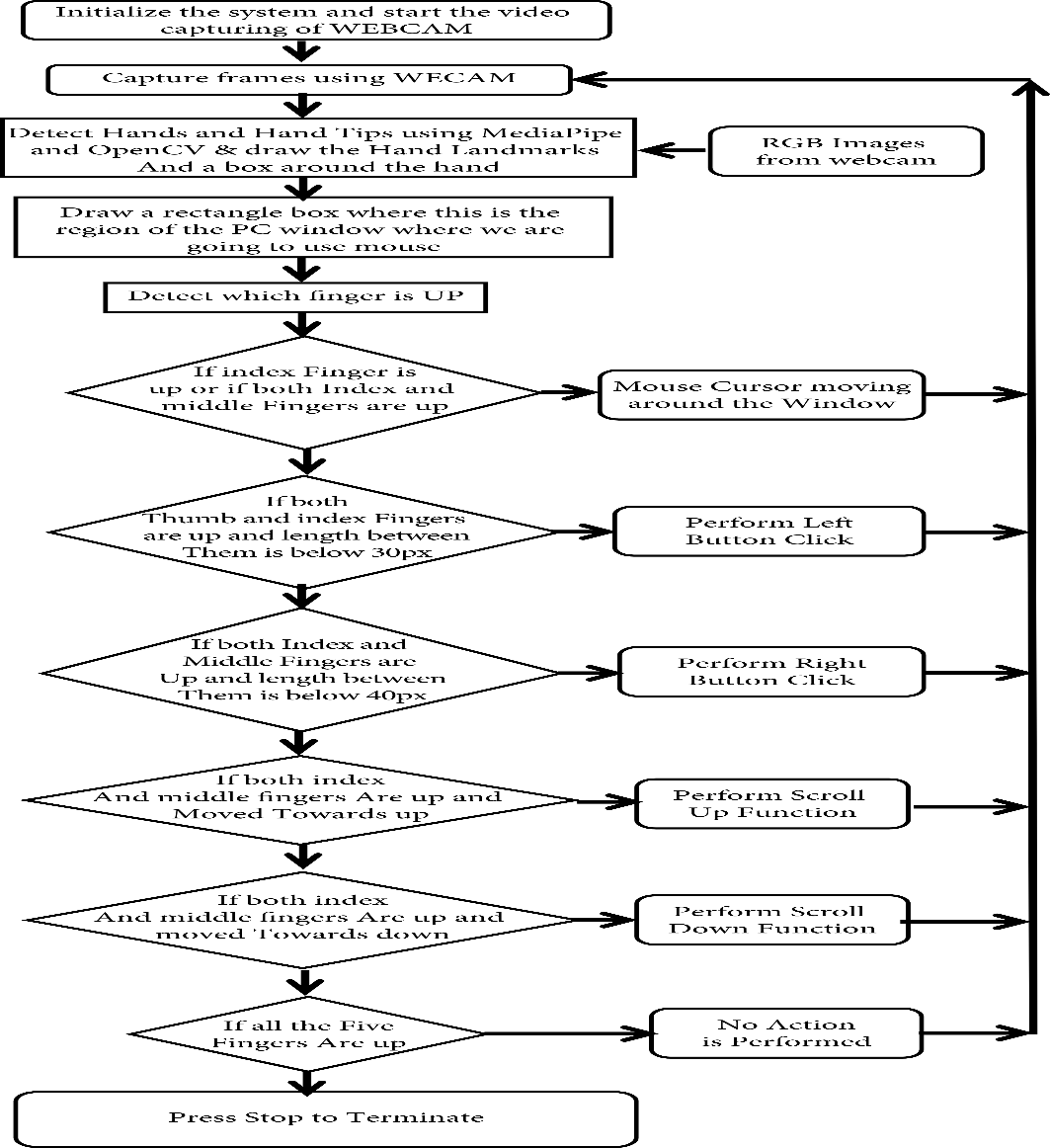
**5.2 Architecture diagram**



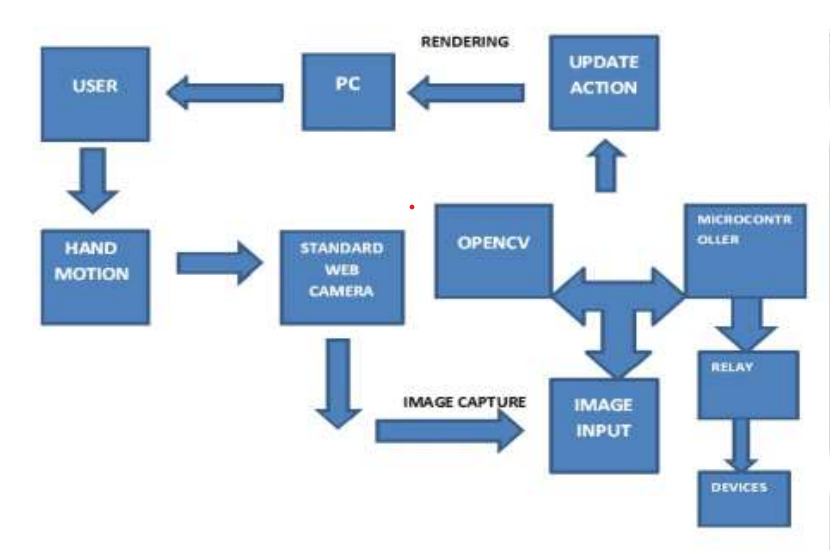
**5.3 Use case diagram**

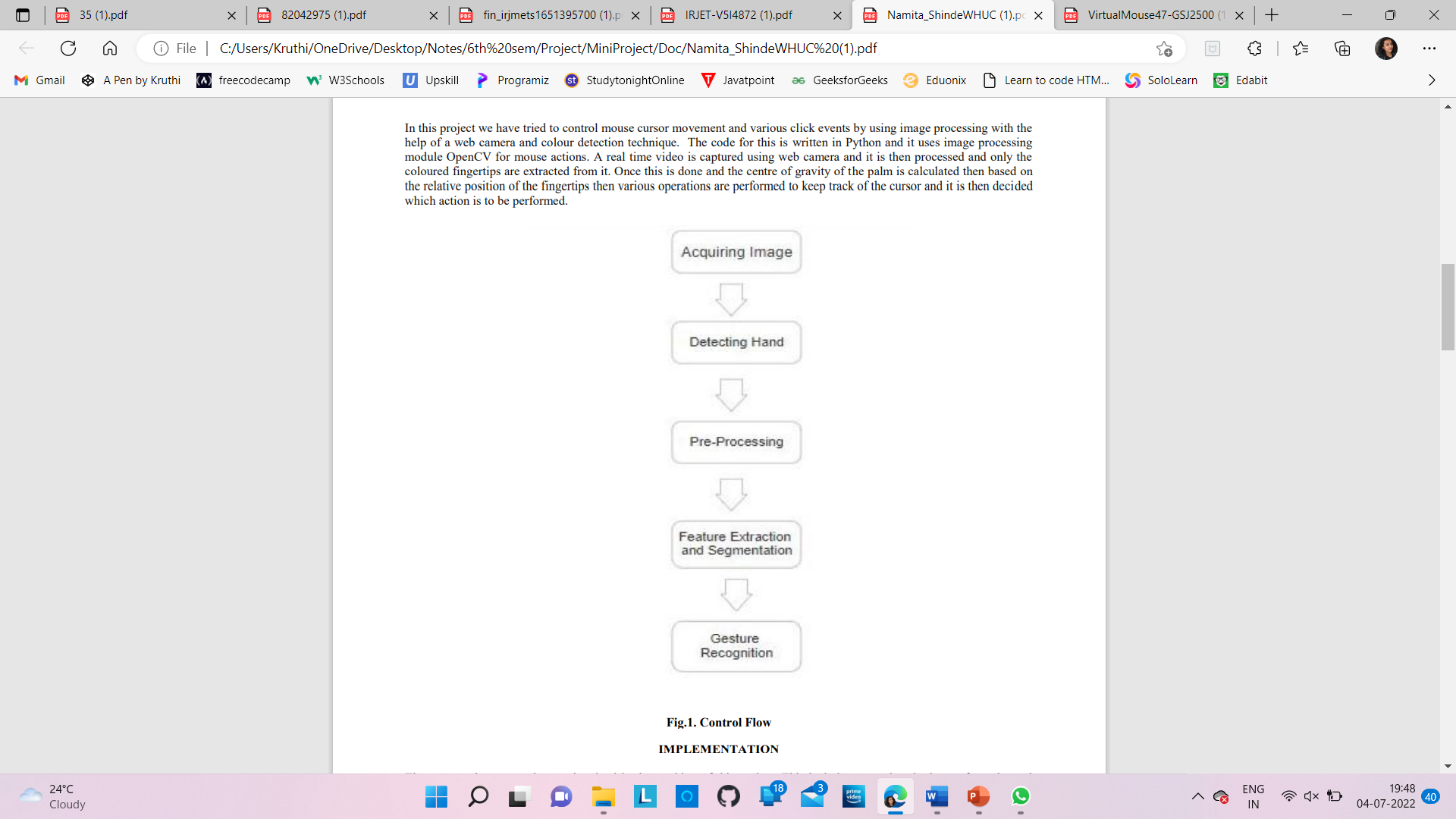


**5.4 Flow Chart diagram**



**5.5 Sequence diagram**

****



**Chapter 6**

**MODULES**

**6.1 Modules**

1. Real time Video Capturing and Inverting Image
2. Localizing and Enhancing Image
3. Finding Fingertips and Centroids
4. Identifying Gesture and Performing Clicking Action

**6.2 Module description**

*A. Real time Video Capturing and Inverting Image*

The web camera which acts as a sensor here captures the real time video of the hand gestures by creating a video capture object. This captured image is divided into several frames which is followed by the processing on individual frames. These individual bytes of images are inverted and are then concatenated to form a flipped image and this is done as the camera captures an inverted image by default. So, to restore the original data this is a mandatory step.

*B. Localizing and Enhancing Image*

The hand gesture is separated from rest of the image for proper segmentation. Generally, a bounding box is used for this which works based on skin color and the hand to be tracked. This image is then enhanced which involves and the blurring effect is removed for proper radiance.

*C. Finding Fingertips and Centroids*

To find the fingertips, first the center of hand is calculated and then a circle of increasing radius is made till it touches its first black pixel. Now, to find the fingertips we can use convex hull algorithm and we discard irrelevant areas using area filters. The largest contour is searched and by method of moments to find the center. This will help to locate the fingertip movement and identify the gesture.

*D. Identifying Gesture and Performing Clicking Action*

This step revolves around defining the position of cursor on the screen. We keep track the movement of centers obtained in previous step and compare them with their previous position. The coordinates of the centroid are sent to the cursor object. Since the object can move continuously so there can be numerous centroids for each frame. This tracking effect helps the mouse move as the end user moves his hands in the camera viewing angle. Patagium library of the OpenCV module can be used to perform various clicking events by creating various flags associated with the mouse buttons.

**Chapter 7**

**PSEUDOCODE**

while True:

check, img = cap.read()

# Reads frames from the camera

imgRGB = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

# Changes the format of the frames from BGR to RGB

lmList = handLandmarks(imgRGB)

# cv2.rectangle(img, (75, 75), (640 - 75, 480 - 75), (255, 0, 255), 2)

if len(lmList) != 0:

x1, y1 = lmList[8][1:]

# Gets index 8s x and y values (skips index value because it starts from 1)

x2, y2 = lmList[12][1:]

# Gets index 12s x and y values (skips index value because it starts from 1)

finger = fingers(lmList)

# Calling the fingers function to check which fingers are up

if finger[1] == 1 and finger[2] == 0:

# Checks to see if the pointing finger is up and thumb finger is down

x3 = numpy.interp(x1, (75, 640 - 75), (0, wScr))

# Converts the width of the window relative to the screen width

y3 = numpy.interp(y1, (75, 480 - 75), (0, hScr))

# Converts the height of the window relative to the screen height

cX = pX + (x3 - pX) / 7

# Stores previous x locations to update current x location

cY = pY + (y3 - pY) / 7

# Stores previous y locations to update current y location

autopy.mouse.move(wScr-cX, cY)

# Function to move the mouse to the x3 and y3 values (wSrc inverts the direction)

pX, pY = cX, cY

# Stores the current x and y location as previous x and y location for next loop

if finger[1] == 0 and finger[0] == 1:

# Checks to see if the pointer finger is down and thumb finger is up

autopy.mouse.click() # Left click

if finger[1] == 1 and finger[2] == 1:

pyautogui.click(button='right')

if finger[1] == 1 and finger[0] == 1:

pyautogui.scroll(200)

if finger[3] == 1 :

pyautogui.scroll(-200)

**Chapter 8**

**TESTING**

**8.1 Basics of Software testing**

Software testing can be divided into two steps:

**1.** **Verification:** It refers to the set of tasks that ensure that the software correctly implements a specific function.

**2.** **Validation:** It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Software Testing can be broadly classified into two types:

**1. Manual Testing:** Manual testing includes testing software manually, i.e., without using any automation tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

**2. Automation Testing:** Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves the automation of a manual process. Automation Testing is used to re-run the test scenarios quickly and repeatedly, that were performed manually in manual testing.

Software testing techniques can be majorly classified into two categories:

**1. Black Box Testing:** The technique of testing in which the tester doesn’t have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

**2. White-Box Testing:** The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

Software level testing can be majorly classified into 4 levels:

**1. Unit Testing:** A level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

**2. Integration Testing:** A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

**3. System Testing:** A level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements.

**4. Acceptance Testing:** A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

**8.2 Test Cases**

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

Performance in various environments

* + Brightness
  + Tilt and Rotation
  + Distance

The purpose of the testing phase is to ensure that the final deliverable program is able to perform flawlessly in terms of accuracy, consistency, and performance. To achieve that, the program has to able to recognize the input provided by the users with minimal adjustment, provide that. Furthermore, the program is required to be able to execute the mouse functions efficiently and accurately as well.

The following describes the outcome of the program testing in various environments:

|  |  |  |  |
| --- | --- | --- | --- |
| Property | Case | Screenshot | Result |
| Brightness | Normal  environment |  | All fingertips are successfully  recognized. The highlighted dots  indicate that the fingers are  identified, compared, and  gesture is recognized correctly. |
|  | Too much  light |  | The fingertips cannot be recognized.  Reason: The intensity of brightness  was too high that it greatly alters the  original RGB values of the targeted  colors, causing it to be  un-recognizable. |
|  | Very little  or no light |  | Some colors cannot be recognized.  Reason: The intensity of brightness  was too low that greatly alters the  original RGB values of the targeted  colors, causing it to be  un-recognizable. |
| Tilt and  rotation |  |  | The orientation of the hand does not  matter as long as the gestures are  clear enough to be interpreted by the  camera. Even though the orientation  is a little tilted, since the gesture is  clear, the gesture is recognized. |
| Distance | Near (around  15-20 cm away  from webcam) |  | All fingertips are successfully  recognized. The highlighted dots  indicate that the fingers are  identified, compared, and  gesture is recognized correctly. |
|  | Too near ( less  than 5 cm from  Webcam) |  | All fingertips cannot be recognized.  The fingers are too near to the  webcam and hence cannot be  recognized accurately. |
| Proper  gesture | Showing gestures  properly |  | The gesture and its orientation is  proper. Hence, the gesture is  recognized accurately. |
|  | Showing wrong  gestures or  gestures with  wrong  orientation |  | The finger is not in the proper  orientation. Hence the webcam fails  to recognize this pattern. |

**Chapter 9**

**RESULTS**

* 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.
* For most laptop touchpad is not the most comfortable and convenient
* Main objective pre-processing is to represent the data in such a way that it can be easily interpreted and processed by the system.
* Reduced cost of hardware
* It focuses on extracting the features over the human hands and then matching their features to recognize the movement of the hand.

**Chapter 10**

**CONCLUSION AND FUTURE SCOPE**

**10.1 Conclusion**

* This system is mainly aimed to reduce the use of hardware components attached with the computer.
* We have tried proposing a system which requires minimal efforts from the user’s side. Use of image processing and CV gives maximum benefit in the field of automation.
* As compared to normal execution, screen touch and recognition-based system, hand gestures recognition proves to be the best and most effective system. Further this system can be expanded to keyboard controls and thus can be highly used in games where both mouse and keyboard controls are required.
* The proposed model is so accurate that it can also be used in real-world applications. For example, it can be used to reduce the spread of COVID19 and eliminate the need for wearable devices.
* Recognition and the interpretation of sign language is one of the major issues for the communication with dump and deaf people. This project is pretty much expressive such that the dumb and deaf people could understand it.
* Not only did the users but it also demonstrated the potential in simplifying user interactions with personal computers and hardware systems

**10.2 Future scope**

There are as yet numerous changes that can be made to our framework like enhancing the execution of the present framework and including components, for example, broadening and contracting windows, shutting window, and so forth by utilizing the palm and different fingers. The present framework is variation to reflection and scale changes and requires legitimate hand signals, great enlightenment innovation and intense camera for the execution of mouse capacities.

Accuracy can simply be expanded at the cost of review by including more stages, however each progressive stage sets aside twice as much opportunity to discover harder negative examples and the applications which advantage from this innovation.

We introduce a picture seeing application for instance of where this innovation could prompt a more characteristic UI. The same could be said for exploring something like Google Maps or perusing envelopes on a screen. In any case, the applications reach a long way past that. They are especially convincing in circumstances where touch screens are not relevant or not as much as perfect.

For instance, with projection frameworks, there is no screen to touch. Here vision-based innovation would give a perfect substitution to touch screen innovation. Likewise, out in the open terminals, consistent utilize brings about the spread of soil and germs. Vision-based frameworks would evacuate the need to touch such setups, and would bring about enhanced connection.

**References**

1] Bharath Kumar Reddy Sandra, Katakam Harsha Vardhan, Ch. Uday, V Sai Surya, Bala Raju, Dr. Vipin Kumar “GESTURE-CONTROL-VIRTUAL-MOUSE”.

2] Vijay Kumar Sharma, Vimal Kumar, Md. Iqbal, Sachin Tawara, Vishal Jayaswal. “Virtual Mouse Control Using Hand Class Gesture”. Department of Computer Science and Engineering MIET, Meeru

3] Lira, M., Egito, J.H., Dall’Agnol, P.A., Amodio, D.M., Gonçalves, Ó.F., Boggio, P.S. “The influence of skin colour on the experience of ownership in the rubber hand illusion” 2017.

4] D. L. Quam, “Gesture recognition with a DataGlove,” IEEE Conference on Aerospace and Electronics, vol. 2, pp. 755–760, 1990. View at: Publisher Site | Google Scholar.

5] Abhilash S, Lisho Thomas, Naveen Wilson, “VIRTUAL MOUSE USING HAND GESTURE” Chaithanya Students of Dept of computer science and engineering Mar Athanasius College of Engineering