**A MINI PROJECT**

on

**VIRTUAL MOUSE USING HAND GESTURES**

*Submitted*

*In partial fulfilment for the requirement for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**Computer Science and Engineering.**

**By**

**S. ABHIGNA 20641A05J7**

**P. KARTHIK 18641A0536**

**T. JAYANTH 20641A05k6**

**J. PRASHANTH 21645A0515**

***Under the guidance of***

**Dr. Manyam Sukesh**

Associate Professor, Department of CSE.



**Department of Computer Science & Engineering**

**Vaagdevi College of Engineering**

(UGC Autonomous, Accredited by NAAC with “A”)

Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)

(2020-2024)

**VAAGDEVI COLLEGE OF ENGINEERING**

(UGC Autonomous, Accredited by NBA, Accredited by NAAC with “A”)

Bollikunta, Khila Warangal (Mandal), Warangal Urban –506005(T.S)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the mini project entitled “***VIRTUAL MOUSE USING HAND GESTURE”***

***is*** submitted by **S. ABHIGNA 20641A05J7, P. KARTHIK 18641A0536, T. JAYANTH 20641A05K6 AND J. PRASHANTH 21645A0515** in partial fulfilment of the requirements for the award of the Degree in Bachelor of Technology in Computer Science and Engineering during the academic year 2023-2024.

**GUIDE: HEAD OF THE DEPARTMENT:**

Dr. Manyam Sukesh Dr. N. Satyavathi

**EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

The development of the project in mini project though was an arduous task, it has been made by the help of many people. We are pleased to express our thanks to the people whose suggestions, comments, criticisms greatly encouraged us in betterment of the project.

We would like to express my sincere gratitude and indebtedness to my project Guide **Dr. MANYAM SUKESH** garu for his valuable suggestions and interest throughout the course of this project.

We would like to express my sincere thanks and profound gratitude to **Dr. K. PRAKASH** garu principal of Vaagdevi College of Engineering, for his support, guidance and encouragement in the course of our project.

We are also thankful to the Head of the Department Dr. **N. SATYAVATHI** garu for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are highly thankful to the Project Coordinators for their valuable suggestions, encouragement and motivations for completing this project successfully.

We are thankful to all other faculty members for their encouragement. We convey my heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed.

Finally,we would like to take this opportunity to thank my family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

**S**. ABHIGNA 20641A05J7

P. KARTHIK 18641A0536

T. JAYANTH 20641A05k6

J. PRASHANTH 21645A0515

**ABSTRACT**

This project promotes an approach for the human computer interaction (HCL). Where we use real time camera for controlling the mouse function. Our proposed project is on hand gesture-based system that allows users to control desktop mouse movements using hand gesture. To detect hand gesture movements, our system makes use of a desktop webcam. The goal is to control mouse cursor functions with a simple camera or webcam rather than a traditional or standard devices. Using only a camera, the Virtual Mouse provides an infrastructure between the user and the machine. It enables the user to interact with a machine without the need for any mechanical or physical devices, and even allows to control mouse functions. The domain of the project is AI/ML. The programming language we used in this project is python. This ai virtual mouse project is based on the concept of computer vision.

**TABLE OF CONTENTS**

*S.NO CONTENT PAGE NO*

**CHAPTER-1**

**INTRODUCTION** 07

1.1 Literature survey 08-09 1.2 Objective 10

1.3 Existing System 11-12 1.4 Proposed System 13

1.5 Software Requirements 14-15

1.6 Hardware Requirements 16

**CHAPTER-2**

**SYSTEM DESIGN** 17-20

2.1 Algorithm 21-24 2.2 Flowchart 25

**CHAPTER-3**

**CODE** 26-29

**CHAPTER-4**

**TASTE CASES** 30-31

**CHAPTER-5**

**RESULTS** 32-33

**CHAPTER-6**

**FEATURE SCOPE AND CONCLUSION** 34

6.1 Future Scope 34 6.2 Conclusion 34

**CHAPTER-7**

**BIBLIOGRAPHY** 35-36

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| **2.2** | FLOW CHART | 25 |
| **4** | ACCURACY OF MOUSE OPERATIONS | 30 |
| **4** | ACCURACY (BAR GRAPH) | 31 |
| **4** | COMPARATION GRAPH WITH OTHER MODELS | 31 |

**LIST OF SCREEN SHORTS**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| **1.3** | OPTICAL MOUSE | 11 |
| **2** | VIRTUALMOUSE BLOCK DIAGRAM | 17 |
| **2.1** | MEDIAPIPE HAND LANDMARK | 21 |
| **2.2** | BLOCK DIAGRAM OF HAND GESTURES. | 21 |
| **2.2** | CLICKS OF MOUSE OPERATIONS | 22-24 |
| **5** | GESTURE CONTROLLER | 32 |

**CHAPTER 1**

**INTRODUCTION**

With the development technologies in the areas of augmented reality and devices that we use in our daily life, these devices are becoming compact in the form of Bluetooth or wireless technologies. This paper 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. The main objective of the proposed system is to perform computer mouse cursor functions and scroll function using a web camera or a built-in camera in the computer instead of using a traditional mouse device. Hand gesture and hand tip detection by using computer vision is used as a HCI with the computer. 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.

While using a wireless or a Bluetooth mouse, some devices such as the mouse, the dongle to connect to the PC, and also, a battery to power the mouse to operate are used, but in this paper, the user uses his/her built-in camera or a webcam and uses his/her hand gestures to control the computer mouse operations. In the proposed system, the web camera captures and then processes the frames that have been captured and then recognizes the various hand gestures and hand tip gestures and then performs the particular mouse function. Python programming language is used for developing the AI virtual mouse system, and also, OpenCV which is the library for computer vision is used in the AI virtual mouse system. In the proposed AI virtual mouse system, the model makes use of the Media Pipe package for the tracking of the hands and for tracking of the tip of the hands, and also, some packages were used for moving around the window screen of the computer for performing functions such as left click, right click, and scrolling functions. The results of the proposed model showed very high accuracy level, and the proposed model can work very well in real-world application with the use of a CPU without the use of a GPU.

* 1. **Literature survey**
* Industrial robot control, sign language translation, smart surveillance, lie detection, visual environment manipulation, and rehabilitation devices for those with upper extremity physical limitations can all benefit from virtual hand gesture recognition. In addition to being inconspicuous, virtual hand gesture recognition systems may be a natural means of interacting with machines, making them an important sort of input mechanism. The ability to reliably detect gestures from numerous perspectives is a difficult component of this technology
* According to a survey and a sign language study, the hand gesture is the easiest and most natural manner of communicating among the many gesture communications modalities. With the help of recent breakthroughs in the field of computer vision and pattern recognition, real-time vision-based hand gesture identification is becoming more and more possible for Human Computer Interaction.
* For laptops, there are a variety of quick access techniques for hand and mouse gestures. Using our project, we could use the laptop or webcam to recognize the hand motion and operate the mouse and execute simple operations like mouse pointer control, select and deselect using left click. The completed project is a laptop based "Zero Cost" hand recognition system that employs simple algorithms to determine the hand, hand movements, and assign an action to each movement.
* A camera was used to capture hand motions using color detection methods in this project. The utilization of a web camera is the essential component of this technique. Abhik Banerjee and Abhirup Ghosh wrote this paper to cost-effectively construct a virtual human-computer interface device. There were some restrictions on their work, such as the need for a light operating system background and the absence of objects with vivid colors. Computers with a specific high configuration function well.
* In this study, which Yimin Zhou et al. reported, where a high-level hand feature extraction approach for real-time gesture detection was provided. The created system has good accuracy in both the extraction of flexional and extensional fingers. However, only computers with high configurations can use this method.
* Several color bands were used in this experiment, which was described by Pooja Kumari et al., where various colored bands carried out various tasks. The number of colors is used as the key to control mouse actions. But the system was managed by a number of colors. Instead of using different gestures, the number of colors is used to perform a function.
* This paper based on a background extraction and contours detection system was proposed by Aashni Haria et al. where they conducted two sets of assessments in order

to determine the correctness of their method. In the initial round of evaluations, they made use of settings with a variety of uniformly simple backdrops. For the second

assessment, they utilized backdrops that had a number of discrepancies. Ten times were given for each gesture in each setting the numbers' average. The accuracy gained was 85% and 80%, which was calculated as the percentage of times a given gesture was successfully identified. But working with it is incredibly slow.

* The operation of a cursor control system using hand gestures captured from a webcam through a color detection technique performed in this project which was published by Abhilash SS et al. However, it was limited to a few mouse actions and is inoperable against a static background.
* A detailed explanation of the algorithms and methodologies for the color detection of a virtual mouse was given in this project by Kollipara Sai Varun et al. In this paper, Open CV (OpenSource Computer Version Library) is primarily used for video capture. The highlight color provided by the user for mouse movement is used in this paper for color detection and mouse movement.
* This project based can be helpful for presentations as well as for minimizing workspace requirements and the weight of additional hardware. A common way to interact with computers without a mouse device is by using fingertip tracking as a virtual mouse. Kabid Hasan Shibly et al present a novel virtual mouse technique in this paper that makes use of fingertip detection and RGB-D images.

**1.2 OBJECTIVE**

* 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, 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.
* The scope of this project is to develop a virtual mouse that will be operated without touching any device or screen. In today’s world where we are adjusting our living while being in a pandemic, a touch less mouse controller will be useful to eliminate the risk of spreading infection through touch on public service devices. A virtual mouse will be introduced soon to replace the physical computer mouse in order to promote convenience while still allowing accurate interaction and control of the computer system. The virtual mouse can be used without touching the screen. This project can improve the scope of Human Computer Interaction technology to be explored more.
* This project's objective is to build a Virtual Mouse application that emphasizes a few crucial programming concepts.

The following describes the overall goals of this project:

1. The precision of the suggested AI virtual mouse is about 98%, which is higher than that of other systems that have already been put into use. Amidst the COVID-19 condition, it is not safe to use the devices by touching them because it may result in a hypothetical situation of propagation of the virus by contacting the devices, therefore the proposed AI virtual mouse can be utilized to control the PC mouse operations without using the actual mice.
2. Drawings in two dimensions and three dimensions can be produced using the same technique.
3. Virtual reality and augmented reality games can be played with AI without the requirement of wireless or wired mouse devices.

**1.3 Existing System**

The motions of an optical mouse, which is widely used today, rely on LEDs, or light-emitting diodes, to detect movements in relation to the surface underneath them, but a laser mouse is a type of optical mouse that makes use of coherent laser lights. The optical mouse replaces the mechanical mouse, which relied on rollers to control movement, by using an image array of photodiodes to identify movement.

Components of an optical mouse:

The optical system consists mainly of three parts:

* Lens
* Light Source
* Sensor

These three parts are assembled on a custom base plate, i.e. Printed Circuit Boards (PCB), and the clip. The lens is the largest part and is mounted on the base plate of the mouse. [5] In addition, it is still susceptible to button switch degradation, which will again result in improper mouse operation unless it is disassembled and repaired. Additionally, prolonged use without proper cleaning or maintenance can result in dust particles getting trapped between the LEDs, making it difficult for both optical and laser mice to detect surfaces.

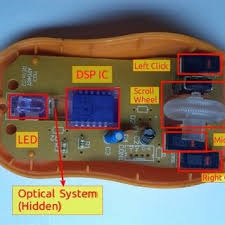


Figure 1.2 Optical Mouse with Top Cover Removed

**Advantages of the Optical and Laser Mouse:**

1. Enables more accuracy with fewer hand movements.
2. Extended life span.

**Disadvantages of existing system**

* Here the existing system has wired mouse, wireless mouse and image segmentation using hand.
* In wired mouse we have only the specific length of the wire, So we can’t use more distance to the computer because it is wired mouse and when it comes to wireless mouse we have to charge it. It will be run using Bluetooth. In these both we have to cost for physical devices.
* When it comes to image segmentation using hand. It will look our total hand as

image and segmentation is done. It will work only if we use color bands to our fingers. But in proposed system we will take land marks of our hands.

* Existing system takes more space and time complexity.

**1.4 PROPOSED SYSTEM**

To design a hand-tracking, finger-only virtual mouse that can track movements. We used several combinations of fingers to do various mouse movements, depending on whatever specific combination of fingers was identified. When there is a need to conserve space or when moving about, the use of a virtual mouse might be observed. The proposed system's users are not required to use any gadgets or sensors, and they are not required to paint their fingertips a particular colour. It makes computer user interaction possible without a hardware mouse device. It is moreover affordable and easy to utilize. For computer vision tasks, the OpenCV library is employed, and the Media Pipe framework is used for tracking and identifying hand movements. The application monitors and differentiates between hand motions and hand tips using machine learning algorithms. A machine learning pipeline uses a Google open-source technology referred to as Media Pipe.

The developer builds systems for application-related reasons as well as for graph-based system creation and analysis utilizing the Media Pipe framework. The pipeline configuration is where the actions in the Media Pipe-using system are carried out. The flexibility of the pipeline to run on many platforms enables scalability on desktops and mobile devices.

**Advantages of proposed system:**

* It is an open-source.
* It can be used to reduce the space for using the physical mouse.
* It can be used in situations, where we cannot use the physical mouse.
* The system eliminates the usage of devices.
* It improves the human-computer interaction.
* We can 2D and 3D figures easily.
* We can play real time virtual games.

**1.5 Software Requirements**

* Numpy
* Mouse
* Media Pipe
* OpenCV

**Numpy**

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

**Advantages of Numpy**

* Numpy is very useful for performing logical and mathematical calculations on arrays and matrices.
* This tool performs these operations much faster and more efficiently than Python lists.
* Numpy uses less memory and storage space, which is the main advantage.

**Mouse**

The mouse module in Python is a library that allows you to control the mouse cursor and buttons. It can be used to automate tasks such as clicking on buttons, moving the cursor, and scrolling. To use the mouse module, you first need to import it into your Python script. You can do this by using the following line of code:

* import mouse

**Media Pipe**

MediaPipe is a cross-platform framework for building machine learning pipelines and solutions for processing time-series data like video, audio, and images. It provides a suite of libraries and tools to quickly apply artificial intelligence (AI) and machine learning (ML) techniques in your applications.

**OpenCV**

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

**1.6 Hardware Requirement**

The hardware required to run and create the Virtual Mouse program is described below:

* Computer desktop or laptop
* The machine such as a desktop or laptop will be used to run a visual program that will display what the camera captured. To promote mobility, a notebook, which is a tiny lightweight, and affordable laptop computer, is offered.

System will be using

Processor : Core2Dual

Main Memory : 4GB RAM

Hard Disk : 320GB

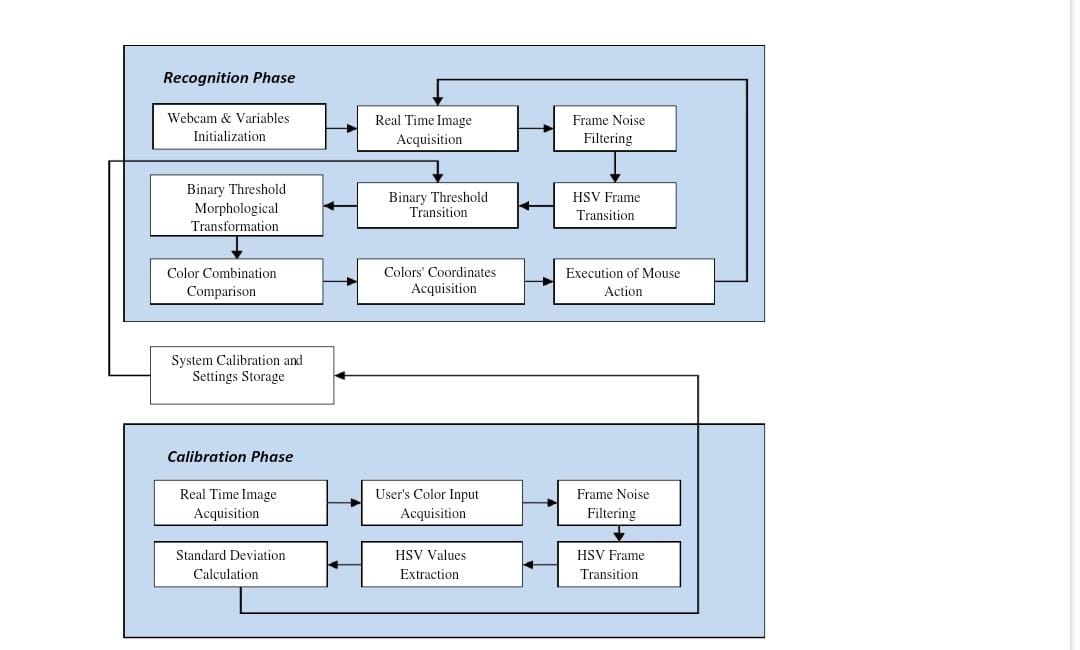
Display : 14” Monitor

* Webcam The use of a webcam for image processing allows the application to process images and determined the position of individual pixels**.**

**CHAPTER 2**

**SYSTEM DESIGN**

There are two main steps in the process of color recognition: the calibration and Recognition phase. In the calibration phase which will be utilized later in the recognition phase, The system will be able to identify the hue saturation value of the colors selected by the user it will safe the parameters and setting into text document for later use. The system will begin to take frames during the recognition phase and look for color input based on the values that have been stored during the calibration process phase. The following figure depicts the stages of the virtual mouse:



**FIGURE: Virtual Mouse Block Diagram**

**Calibration Phase**

1. **Acquisition of Real Time Image:**

The application will begin by using a webcam to capture live photos while it waits for human input on color. To lessen the processing demands of processing the pixel inside the captured frame, the size of the obtained image will be compressed to a manageable level.0

* **User's Color Input Acquisition:**

The application captures frames made up of user-submitted input colors. The captured frame is then transmitted for processing, where it goes through a number of changes and calculations to provide calibrated HSV values.

* **Frame Noise Filtering :**

Every taken frame contains noises that will impair the program's efficiency and accuracy so the frame must be noise-free. Filters must be added to the collected frames to remove the undesirable noise in order to accomplish that. For the current project, the widely used smoothing technique known as the Gaussian filter will be utilized to remove noise from a frame.

1. **Transition of HSV Frame:**

It is necessary to convert the acquired frame from BGR to HSV format.

1. **Extraction of HSV Values**:

The transformed frame must be split into three separate planes in order to obtain the HSV values; this can be accomplished by converting the frame from a multi-channel array into a single channel array using splitting.

1. **Calculation of Standard Deviation:**

The Standard Deviation calculation, a computation used to quantify the amount of variation / dispersion among other HSV values, must be performed in order to determine the maximum and lowest of the HSV values. Additionally, three-sigma rules must be used in the calculation to produce an accurate range of values, increasing the likelihood that the collected values will fall inside the three-sigma interval.

**Recognition Phase**

* 1. **Webcam and variable initialization:** Early in the recognition phase, the software will initialize the necessary variables that will be used to store various frame kinds and value ranges, each of which will be used to complete a specific task. Additionally, during this phase, the program gathers the calibrated HSV values and settings that will be applied later during the Binary Threshold transitions.
  2. **Real Time Image Acquisition:** Using (cv: Video Capture cap (0)), the real-time picture is taken using the camera. Each image is placed into a frame variable (cv: Mat), which is then flipped and compressed to a manageable size to lessen process load.
  3. **Frame noise Filtering:** The noise in the collected frames will be reduced using Gaussian filters, just like it was done during the calibration step.
  4. **HSV frame transition**: It is necessary to change the captured frame's format from BGR to HSV. using cvtColor (src, dst, CV\_BGR2HSV) for example.
  5. **Binary Threshold Transition:** A range check will be performed on the converted HSV frame to see if the HSV values fall within the range of the HSV variables collected during the calibration step. The frame will be converted into the binary system Threshold as a consequence of the range check, with a portion of the frame being set to 255 (1 bit) if the frame falls within the given HSV values and to 0 (0 bit) otherwise
  6. **Binary Threshold Morphological Transmission:** After obtaining the binary threshold, the frame will go through a procedure termed morphological conversion, which is a structural operation to get rid of any foreground gaps and tiny objects. Erosion and Dilation are two morphological operators that make up the transformation. In order to remove minor sounds, the Erosion operator works by eroding the foreground object's edges and reducing the area of the binary limit. In terms of dilation, it is the reverse of erosion and raises the binary threshold area, allowing an item that has been eroded to regain its former shape.
  7. **Color Combination Comparison:** The software will determine the remainder of the number of objects by highlighting them as blobs after collecting the results from the morphological transformation process; this procedure 23 necessitates the use of the cvblob library, an OpenCV add-on. In order to identify how the mouse behaves according to the color configurations identified within the collected frames, the calculation's

findings will then be sent for comparison.

* 1. **Colors’ Coordinates Acquisition:** The program will display the general shape of each object that falls within the binary threshold where it will compute the shape's area and midpoint coordinates. The coordinates will be kept and utilized subsequently to execute

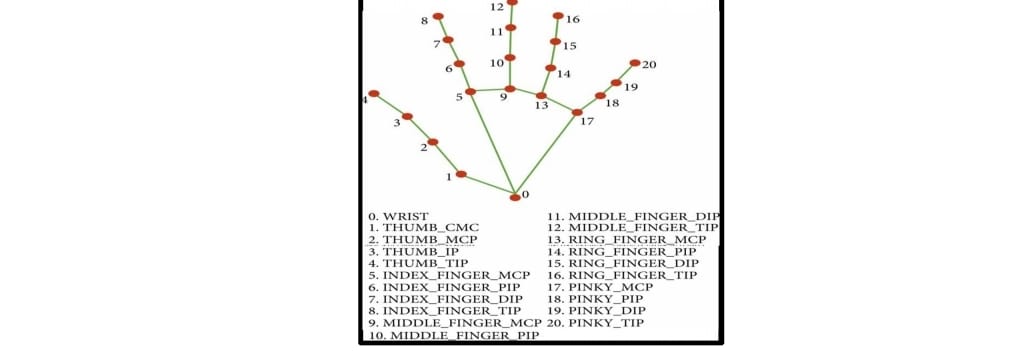
different mouse operations based on the data gathered, either in setting cursor positions or in calculating the separation between each of the spots.

Execution of Mouse Action: Based on the color combinations found in the processed frame, the application will carry out mouse operations. The mouse movements will be carried out in accordance with the coordinates that the software has supplied, and the application will keep acquiring and processing new real-time images up until the users depart it.

**2.1 Algorithm**

**Operations of cursor:-**

* Hand Landmark detection(fingers up(1) and down(0))
* Left click
* Double click
* Right click
* Scroll up
* Scroll down



**FIG: Media pipe hand landmark**

**METHODOLOGY:**

**Steps:-**

* It will detect the camera, video interface will be start
* The camera can extract and recognize human hand gestures from video interface
* Hand tracking functionality is done by media Pipe
* After the recognition the cursor move accordingly, to perform various operations.

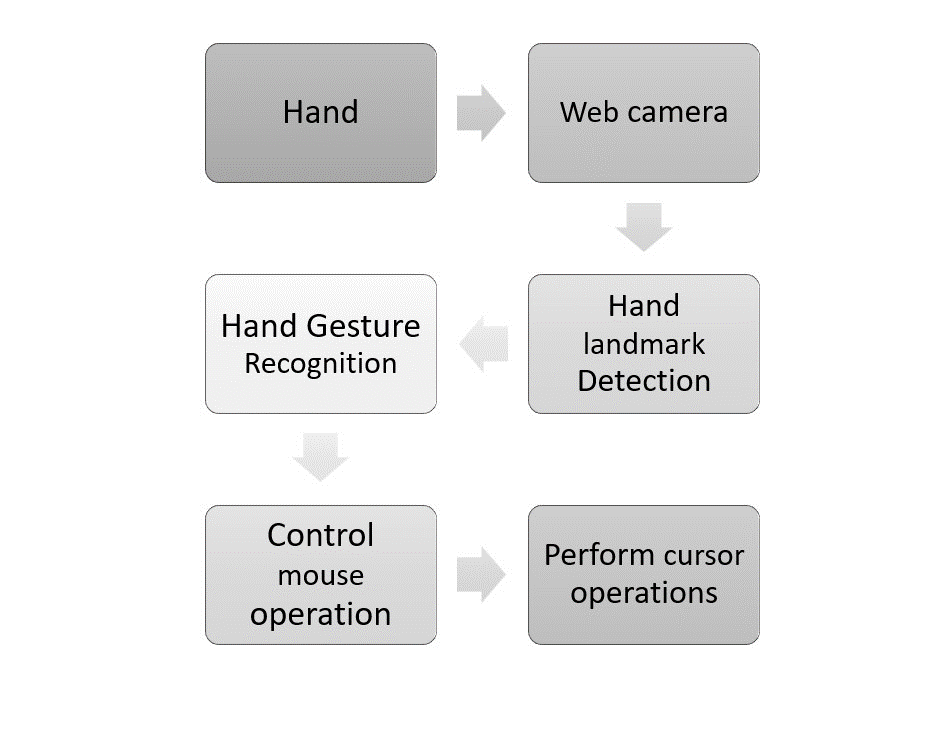


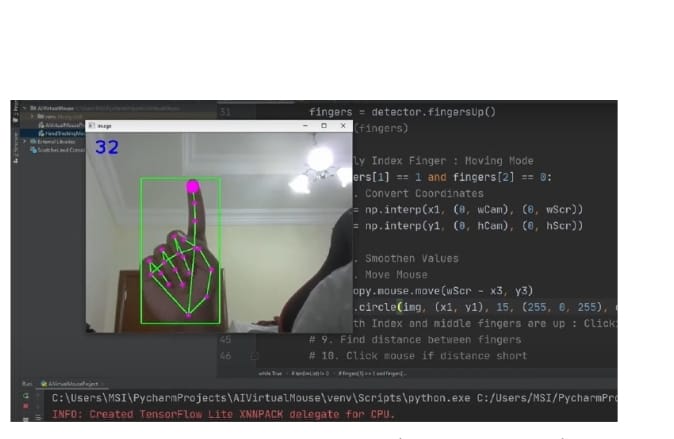
FIG NAME: Block diagram of hand gesture

**A. Camera used in the virtual gesture mouse project**

open-cv is python vision library that contains associate in the organized ai virtual mouse system depends upon the edges that are gotten by the camera in associate in nursing passing pc. pictures can be conveyed in concealing layered with 3 channels, grayscale with pixel values fluctuating from 0 (dull) to 255 (white), and twofold portraying dim or white characteristics (0 or 1) specifically.

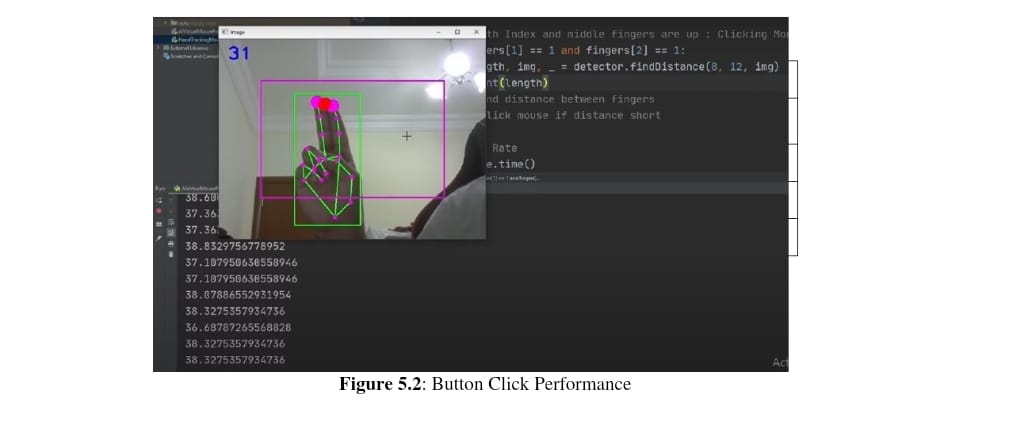
**B. Moving Hand through the Window using Rectangular Area**

The AI virtual mouse framework utilizes the instructive algorithmic rule, and it changes over the co-ordinates of tip from the camera screen to the pc window full screen for the mouse. Whenever the hands unit saw and keeping in mind that we've missing to see that finger is up for topic the specific mouse perform, Associate in Nursing rectangular box is attracted concerning the pc window at ranges the camera locale any spot we've a penchant to will every now and again move all through the window plan the mouse pointer



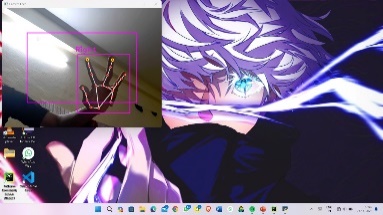
**C. for the mouse to perform left button click**

If both the index finger with tip id = 1 and the thumb finger with tip id= 0 are up and the distance between the two fingers is lesser than 30px and like both the tips gets attached, the computer is made to perform the left mouse button



**D. For the Mouse to Perform Double Click**

If the index finger with tip Id =1 and the middle finger with tip Id = 2 are up and the distance between the two fingers is lesser than 40px, and other finger are also in upward direction but the distance between them is greater than 40px, the computer is made to perform the double click



**E. for the mouse to perform right button click**

if the index finger with tip id =1 and the thumb finger with tip id = 0 are up and the distance between the two fingers is greater than 40px, rest of the finger are in downward direction, the computer is made to perform the right mouse button click.



**F. For the Mouse to Perform Scroll up Function**

If both the index finger with tip Id =1 and the middle finger with tip Id = 2 are up and the distance between the two fingers is lesser than 40px and if the two fingers are moved up the page, the computer is made to perform the scroll up mouse function.



**G. for the mouse to perform scroll down function**

if both the index finger with tip id = 1 and the middle finger with tip id = 2 are down and the distance between the two fingers is lesser than 40px, the two fingers are moved down the page, the computer is made to perform the scroll down mouse function.

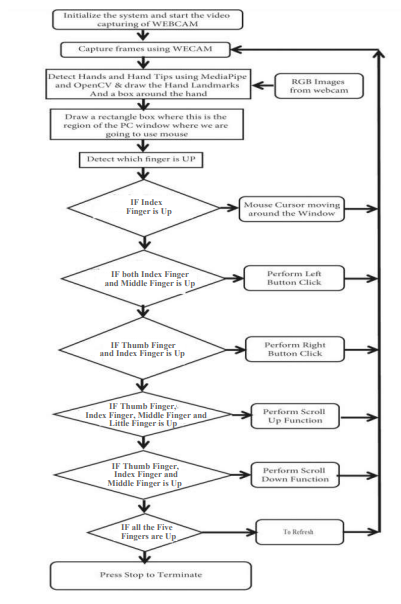


**H. To Perform Refresh Function**

If all the finger are in downward direction. The Finger are like wrist. When the fingers gets wristed and the wrist gets open immediately. This wrist function of finger performs the refresh function.



**2.2 Flow Chart**

****

**Fig : Virtual mouse using hand gesture**

**CHAPTER 3**

**CODE**

import cv2

from cvzone.HandTrackingModule import HandDetector

import mouse

import threading

import numpy as np

import time

frameR = 100

cam\_w, cam\_h = 640, 480

cap = cv2.VideoCapture(1)

cap set(3, cam\_w)

cap.set(4, cam\_h)

detector = HandDetector(detectionCon=0.9, maxHands=1)

l\_delay = 0

r\_delay = 0

double\_delay = 0

def l\_clk\_delay():

global l\_delay

global l\_clk\_thread

time.sleep(1)

l\_delay = 0

l\_clk\_thread = threading.Thread(target=l\_clk\_delay)

def r\_clk\_delay():

global r\_delay

global r\_clk\_thread

time.sleep(1)

r\_delay = 0

r\_clk\_thread = threading.Thread(target=r\_clk\_delay)

def double\_clk\_delay():

global double\_delay

global double\_clk\_thread

time.sleep(2)

double\_delay = 0

double\_clk\_thread = threading.Thread(target=double\_clk\_delay)

l\_clk\_thread = threading.Thread(target=l\_clk\_delay)

r\_clk\_thread = threading.Thread(target=r\_clk\_delay)

double\_clk\_thread = threading.Thread(target=double\_clk\_delay)

while True:

success, img = cap.read()

if success:

img = cv2.flip(img, 1)

hands, img = detector.findHands(img ,flipType=False)

cv2.rectangle(img, (frameR, frameR), (cam\_w - frameR, cam\_h - frameR), (255, 0, 255), 2)

if hands:

lmlist = hands[0]['lmList']

ind\_x, ind\_y = lmlist[8][0], lmlist[8][1]

mid\_x, mid\_y = lmlist[12][0], lmlist[12][1]

cv2.circle(img, (ind\_x, ind\_y), 5, (0, 255, 255), 2)

cv2.circle(img, (mid\_x, mid\_y), 5, (0, 255, 255), 2)

fingers = detector.fingersUp(hands[0])

# Mouse movement

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

conv\_x = int(np.interp(ind\_x, (frameR,cam\_w-frameR),(0, 1536)))

conv\_y = int(np.interp(ind\_y, (frameR,cam\_h-frameR),(0, 864)))

mouse.move(conv\_x,conv\_y)

print(conv\_x,conv\_y)

# Mouse Button Clicks

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

if abs(ind\_x-mid\_x) < 25 :

# Left Click

if fingers[4] == 0 and l\_delay == 0:

mouse.click(button = "left")

l\_delay = 1

l\_clk\_thread.start()

# Right Click

if fingers[4] == 1 and r\_delay == 0:

mouse.click(button = "right")

r\_delay = 1

r\_clk\_thread.start()

# Mouse Scrolling

if fingers[1] == 1 and fingers[2] == 1 and fingers[0] == 0 and fingers[4] == 0:

if abs(ind\_x-mid\_x) < 25:

mouse.wheel(delta=-1)

if fingers[1] == 1 and fingers[2] == 1 and fingers[0] == 0 and fingers[4] == 1:

if abs(ind\_x - mid\_x) < 25:

mouse.wheel(delta=1)

# Double Mouse Click

if fingers[1] == 1 and fingers[2] == 0 and fingers[0] == 0 and fingers[4] == 0 and double\_delay == 0:

double\_delay =1

mouse.double\_click(button="left")

double\_clk\_thread.start()

cv2.imshow("Camera Feed", img)

cv2.waitKey(1)

**CHAPTER 4**

**TEST CASES**

The proposed model of AI virtual mouse demonstrates the idea of computer vision technology and machine learning capabilities. For verifying and testing of the proposed AI virtual mouse system only a certain quantity of datasets are attainable. The hand tracking, finger-tip detection and gestures recognition have been performed in various illuminated conditions and at diversified distances from the camera. The results of the testing are given below. The test was performed 25 times by 4 persons resulting in 600 gestures with manual labelling, and this test has been made in different light conditions and at different distances from the screen, and each person tested the AI virtual mouse system 10 times in normal light conditions, 5 times in faint light conditions, 5 times in close distance from the webcam, and 5 times in long distance from the webcam, and the experimental results are tabulated in Table 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mouse Function Performed** | **Correct Operation** | **Incorrect or Failed Operation** | **Accuracy** |
| Pointer Movement | 100 | 0 | 100 |
| Left Click | 98 | 2 | 98 |
| Right Click | 94 | 6 | 94 |
| Scrolling Up | 98 | 2 | 98 |
| Scrolling Down | 99 | 1 | 99 |
| **Result** | **489** | **11** | **97.8** |

Table 1. Accuracy of mouse operations

The following bar graph shows the accuracy of operations that have been observed by testing the model under various illumination conditions and numerous operations. Table 1 shows the comparison between different functions and how they are accurate.Lets see the accuracy using bar graphs which is one of the most using stastical tool used to calculate the function parameters.

FIG 12. Accuracy of mouse operations (Bar Graph)

**Comparing with previous models**

To see how well the proposed model works against previous systems we have compared them and the figure 13 shows the comparison of the AI virtual mouse model with models mentioned in the related models section.

FIG 13. Comparison graph with other models

**CHAPTER 5**

**RESULTS**

Al virtual mouse using hand gesture recognition is a technology that allows users to control their computers or other devices without the need for a physical mouse. Instead, users can make hand gestures that are recognized by the Al- powered software to control the cursor and perform various actions. In our research studies and projects we have explored the use of Al virtual mouse using hand gesture recognition. Some of the key findings and results from these studies include:

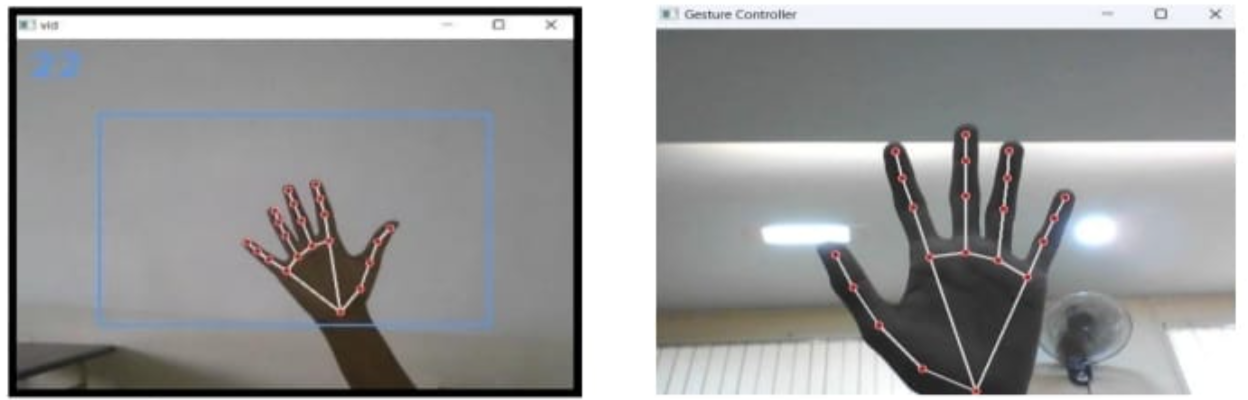


FIG:Results(Screen shot)

* Improved accessibility: Al virtual mouse using hand gesture recognition can greatly improve accessibility for users with physical disabilities or impairments that make it difficult to use a physical mouse or keyboard.
* Increased productivity: Some studies have shown that using Al virtual mouse with hand gestures can be faster and more efficient than using a physical mouse, especially for tasks that involve a lot of scrolling or zooming.
* Gesture recognition accuracy: The accuracy of the hand gesture recognition technology used in Al virtual mouse systems can vary depending on factors such as lighting conditions, camera quality, and the complexity of the gestures being used.
* User satisfaction: Overall, users tend to be very satisfied with Al virtual mouse using hand gesture recognition technology, particularly when it comes to its ease of use and ability to provide a more natural and intuitive way of interacting with computers.
* Overall, Al virtual mouse using hand gesture recognition is a promising technology with the potential to revolutionize the way we interact with computers and other devices. While there are still some challenges to be overcome, ongoing research and development in this are likely to lead to even more advanced and sophisticated Al virtual mouse systems in the future.

**CHAPTER 6**

**FEATURE SCOPE AND CONCLUSION**

**6.1 Future Scope**

This AI virtual mouse has some drawbacks such as drop in accuracy of some functions like right click operation and inability to perform other mouse functions such as dragging and dropping, and selecting text. Another major limitation is that this model cannot function in the dark or low light settings. These drawbacks can be addressed in the future and can be overcome.

Apart from the above-mentioned, additionally key board capability can be incorporated to emulate keyboard functions along with the mouse operations which proves to be the scope for the future.

**6.2 Conclusion**

The main objective of the AI virtual mouse system is to control the mouse cursor functions by using the hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a built-in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse functions. From the results of the model, we can come to a conclusion that the proposed AI virtual mouse system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems. Since the proposed model has greater accuracy, the AI virtual mouse can be used for real-world applications, and also, it can be used to reduce the spread of COVID-19, since the proposed mouse system can be used virtually using hand gestures without using the traditional physical mouse. The model has some limitations such as small decrease in accuracy in right click mouse function and some difficulties in clicking and dragging to select the text.

**CHAPTER 7**

**BIBILOGRAPHY**

1. J. Katona, “A review of human–computer interaction and virtual reality research fields in cognitive Info Communications,” Applied Sciences, vol. 11, no. 6, p. 2646, 2021.
2. D. L. Quam, “Gesture recognition with a DataGlove,” IEEE Conference on Aerospace and Electronics, vol. 2, pp. 755–760, 1990.
3. D.-H. Liou, D. Lee, and C.-C. Hsieh, “A real time hand gesture recognition system using motion history image,” in Proceedings of the 2010 2nd International Conference on Signal Processing Systems, July 2010.
4. S. U. Dudhane, “Cursor control system using hand gesture recognition,” IJARCCE, vol. 2, no. 5, 2013.
5. K. P. Vinay, “Cursor control using hand gestures,” International Journal of Critical Accounting, vol. 0975–8887, 2016
6. L. $omas, “Virtual mouse using hand gesture,” International Research Journal of Engineering and Technology (IRJET, vol. 5, no. 4, 2018.
7. P. Nandhini, J. Jaya, and J. George, “Computer vision system for food quality evaluation—a review,” in Proceedings of the 2013 International Conference on Current Trends in Engineering and Technology (ICCTET), pp. 85–87, Coimbatore, India, July 2013.
8. J. Jaya and K. Danushkoti, “Implementation of certain system for medical image diagnosis,” European Journal of Scientific Research, vol. 53, no. 4, pp. 561–567, 2011.
9. P. Nandhini and J. Jaya, “Image segmentation for food quality evaluation using computer vision system,” International Journal of Engineering Research and Applications, vol. 4, no. 2, pp. 1–3, 2014.
10. J. Jaya and K. Danushkoti, “Implementation of classification system for medical images,” European Journal of Scientific Research, vol. 53, no. 4, pp. 561–569, 2011.
11. J. T. Camillo Lugaresi, “MediaPipe: A Framework for Building Perception Pipelines,” 2019, <https://arxiv.org/abs/1906.08172>.
12. Google MAP, https://ai.googleblog.com/2019/08/on-devicereal-time-hand-tracking with.html
13. V. Bazarevsky and G. R. Fan Zhang. On-Device, MediaPipe for Real-Time Hand Tracking.
14. K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, “Realtime computer vision with openCV,” Queue, vol. 10, no.4, pp. 40–56, 2012.