Gesture-Nav: Hand-Controlled Virtual Mouse

By

TAMAL MALLIK (CSE21099 / 759)
AVISHEK MONDAL (CSE21019 / 679)
SOUVIK BAIDYA (CSE21088 / 748)

Bachelor Thesis submitted to

Indian Institute of Information Technology Kalyani

for the partial fulfillment of the degree of

Bachelor of Technology
in
Computer Science and Engineering

November, 2023

Certificate

This is to certify that the synopsis entitled **Gesture-Nav: Hand-Controlled Virtual Mouse** is being submitted by Tamal Mallick, (**Enrollment No.: CSE/21099 /759**), Avishek Mondal (**Enrollment No.: CSE21019/679** and Souvik Baidya (**Enrollment No.: CSE21088/748**, B.Tech., Indian Institute of Information Technology Kalyani, India, for the partial fulfillment of the requirements for the registration of the degree of Bachelor of Technology is an original research work carried by them under my supervision. The synopsis has fulfilled all the requirements as per the regulation of IIIT Kalyani and in my opinion, has reached the standards needed for submission. The works, techniques, and results presented have not been submitted to any other university or Institute for the award of any other degree or diploma.

Dr. Anirban Lakshman
AssistantProfessor
DepartmentofMathematics
IndianInstituteofInformationTechnologyKalyani
Kalyani − 741235, *W.B.*, *India.*

Declaration

We hereby declare that the work being submitted in this thesis entitled,**"Gesture-Nav: Hand-Controlled Virtual Mouse"**, submitted to Indian Institute of Information Technology, Kalyani in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the period from July, 2023 to November 2023 under the supervision of Dr. Anirban Lakshman, Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India, does not contain any classified information.

Tamal Mallick
$EnrollmentNo.:CSE/21099/759$
$IndianInstituteofInformationTechnologyKalyani$
$Kalyani-741235,W.B.,India.$

Avishek Mondal
$EnrollmentNo.:CSE/21019/679$
$IndianInstituteofInformationTechnologyKalyani$
$Kalyani-741235,W.B.,India.$

Souvik Baidya
$EnrollmentNo.:CSE21088/748$
$IndianInstituteofInformationTechnologyKalyani$
$Kalyani-741235,W.B.,India.$

Acknowledgement

We hereby acknowledge our deep sense of gratitude to our supervisor Dr. Anirban Lakshman, Department of Mathematics, Indian Institute of Information Technology Kalyani, for providing us with adequate facilities, ways, and means by which we are able to do this work. We express our sincere gratitude to him, his valuable time, insightful suggestions, and constant support have been indispensable for our progress.

Finally, we would also like to thank our friends, college faculties and family members who in one way or another helped us in doing this work.

Tamal Mallick
$EnrollmentNo.:CSE/21099/759$
$IndianInstituteofInformationTechnologyKalyani$
$Kalyani-741235,W.B.,India.$

Avishek Mondal
$EnrollmentNo.:CSE/21019/679$
$IndianInstituteofInformationTechnologyKalyani$
$Kalyani-741235,W.B.,India.$

Souvik Baidya
$EnrollmentNo.:CSE21088/748$
$IndianInstituteofInformationTechnologyKalyani$
$Kalyani-741235,W.B.,India.$

Abstract

This thesis presents the design and implementation of an Virtual Mouse system using **computer vision** and **gesture recognition techniques**. The system utilizes OpenCV and MediaPipe libraries for hand detection and landmark tracking, enabling real-time interpretation of hand gestures using camera. The virtual mouse allows users to control their computer through intuitive hand movements, translating gestures into mouse actions. Key functionalities include **cursor movement, left-click, right-click, and scroll operations, changing Tabs,opening virtual keyboard (if available), minimizing, collapsing tabs** and many more. Sensitivity parameters and gesture thresholds can be configured at anytime for personalized user experiences. The paper details the workflow, algorithmic structure, and integration of **multi-threading** for efficient **real-time processing**. The virtual mouse system demonstrates potential applications in accessibility and human-computer interaction. The abstract encapsulates the core contributions of the paper, serving as a concise overview for readers and information services.

Abbreviations Used

Abbreviation	**Description**
GUI	Graphical User Interface
HCI	Human Computer Interaction
MHI	Motion History Images
IDE	Integrated Development Environment
OpenCV	Open-Source Computer Vision
NUI	Natural User Interface

Table 1: Abbreviations and Descriptions

Contents

Chapter 1

1 Introduction

In the realm of human-computer interaction (HCI), the virtual mouse has emerged as a revolutionary tool, offering an alternative to the traditional mouse for controlling computer cursors. This innovative technology utilizes hand gestures or other motion-based inputs to translate user movements into precise cursor movements. [2]

At the heart of virtual mouse technology lies the power of image processing and computer vision algorithms. These algorithms, often employing artificial intelligence (AI), are capable of tracking hand movements and gestures with remarkable accuracy. By analyzing the captured video feed from a webcam or other depth sensors, the virtual mouse software can decipher the user's intentions and translate them into corresponding mouse actions.[4]

The benefits of virtual mouse technology extend far beyond the realm of traditional desktop computing. For individuals with disabilities, virtual mice offer a newfound level of independence and interaction with computers, Moreover, virtual mice promote ergonomic principles by reducing strain on the wrists and hands, which is particularly beneficial for individuals who spend extended periods working on computers.

As virtual mouse technology continues to evolve, its potential applications are expanding beyond cursor control. Researchers are exploring the use of virtual mice for gesture-based interactions in gaming, sign language recognition, and virtual reality (VR) environments.[6] The possibilities seem endless, with the potential to revolutionize the way we interact with computers and the digital world.

This project report delves into the intricacies of virtual mouse technology, exploring its principles, applications, and future directions. Through a comprehensive analysis of the technology's underlying algorithms, user interface considerations, and real-world implementations, we aim to provide a comprehensive understanding of this transformative innovation in HCI.

1.1 Problem Statement

In contemporary computing environments, traditional input devices like mice and keyboards may pose accessibility challenges for individuals with physical disabilities or limitations. The need for alternative, intuitive, and hands-free input methods has become increasingly evident. This project addresses the limitations of conventional input devices by proposing an AI-driven Virtual Mouse that enables users to interact with computers through hand gestures, providing a more inclusive and adaptable solution.

1.2 Objective

The primary objective of this project is to design, implement, and evaluate a Virtual Mouse system that leverages computer vision and gesture recognition technologies. The specific goals include :

Hand Gesture Recognition:

Develop a robust hand gesture recognition system capable of accurately interpreting a user's hand movements. [3]

Cursor Control:

Implement a mechanism for translating hand gestures into precise cursor movements on the computer screen, ensuring responsive and natural interaction. [8]

Click and Scroll Actions:

Enable the Virtual Mouse to perform essential actions such as left-click, right-click, and scroll, mimicking the functionalities of a physical mouse. [6]

Customization:

Provide users with the ability to customize sensitivity parameters and gesture thresholds to accommodate individual preferences and comfort levels. [4, 5]

Real-time Performance:

Ensure real-time processing of hand gestures, minimizing latency and providing a seamless and responsive user experience. [7]

Accessibility:

Enhance accessibility for individuals with physical disabilities, allowing them to interact with computers using intuitive hand movements, thereby promoting inclusivity. [9]

User-Friendly Interface:

Design an intuitive user interface for configuring and controlling the Virtual Mouse, making it user-friendly and accessible to a diverse range of users. [1]

Evaluation and Optimization:

Conduct thorough evaluations to assess the accuracy, usability, and overall effectiveness of the Virtual Mouse. Iterate on the design to optimize performance based on user feedback. [2]

Chapter 2

2 Tools and Techniques Used

2.1 Libraries Used

MediaPipe

Media Pipe is a framework which is used for applying in a machine learning pipeline, and it is an open-source framework of Google.[4] The framework is useful for cross platform development since the framework is built using the time series data.

The Media Pipe framework is multimodal, where this framework can be applied to various audios and videos. The Media Pipe framework is used by the developer for building and analyzing the systems through graphs, and it has also been used for developing the systems for the application purpose.[2] The steps involved in the system that uses Media Pipe are carried out in the pipeline configuration. The pipeline created can run on various platforms allowing scalability in mobile and desktops. The Media Pipe framework is based on three fundamental parts, they are:

- Performance evaluation
- Framework for retrieving sensor data
- Collection of components

which are called calculators, and they are reusable. A pipeline is a graph which consists of components called calculators, where each calculator is connected by streams in which the packets of data flow through. Developers can replace or define custom calculators anywhere in the graph creating their own application. The calculators and streams combined create a data-flow diagram; the graph is created with Media Pipe where each node is a calculator, and the nodes are connected by streams. Single-shot detector model is used for detecting and recognizing a hand or palm in real time. The single-shot detector model is used by Media Pipe.[5] First, in the hand detection module, it is first trained for a palm detection model because it is easier to train palms. Furthermore, the non-maximum suppression works significantly better on small objects such as palms or fists. A model of hand landmark consists of locating joint or knuckle co-ordinates in the hand region

OpenCV

OpenCV is a computer vision library which contains image-processing algorithms for object detection. OpenCV is a library of python programming language, and real-time computer vision applications can be developed by using the computer vision library.[8] The OpenCV library is used in image and video processing and analysis such as face detection and object detection. [6]

PyAutoGUI

PyAutoGUI, a popular Python library, can be seamlessly integrated into the A1 virtual mouse system as a vital component. PyAutoGUI provides a wide range of functionalities that enhance the virtual mouse's capabilities and enable efficient computer control. PyAutoGUI offers methods for simulating mouse movements, clicks, scrolling and many more allowing the virtual mouse to accurately mimic the actions of a physical mouse.[3] This integration ensures that the virtual mouse system can perform tasks that require precise mouse interactions in a reliable and efficient manner.

2.2 Algorithm

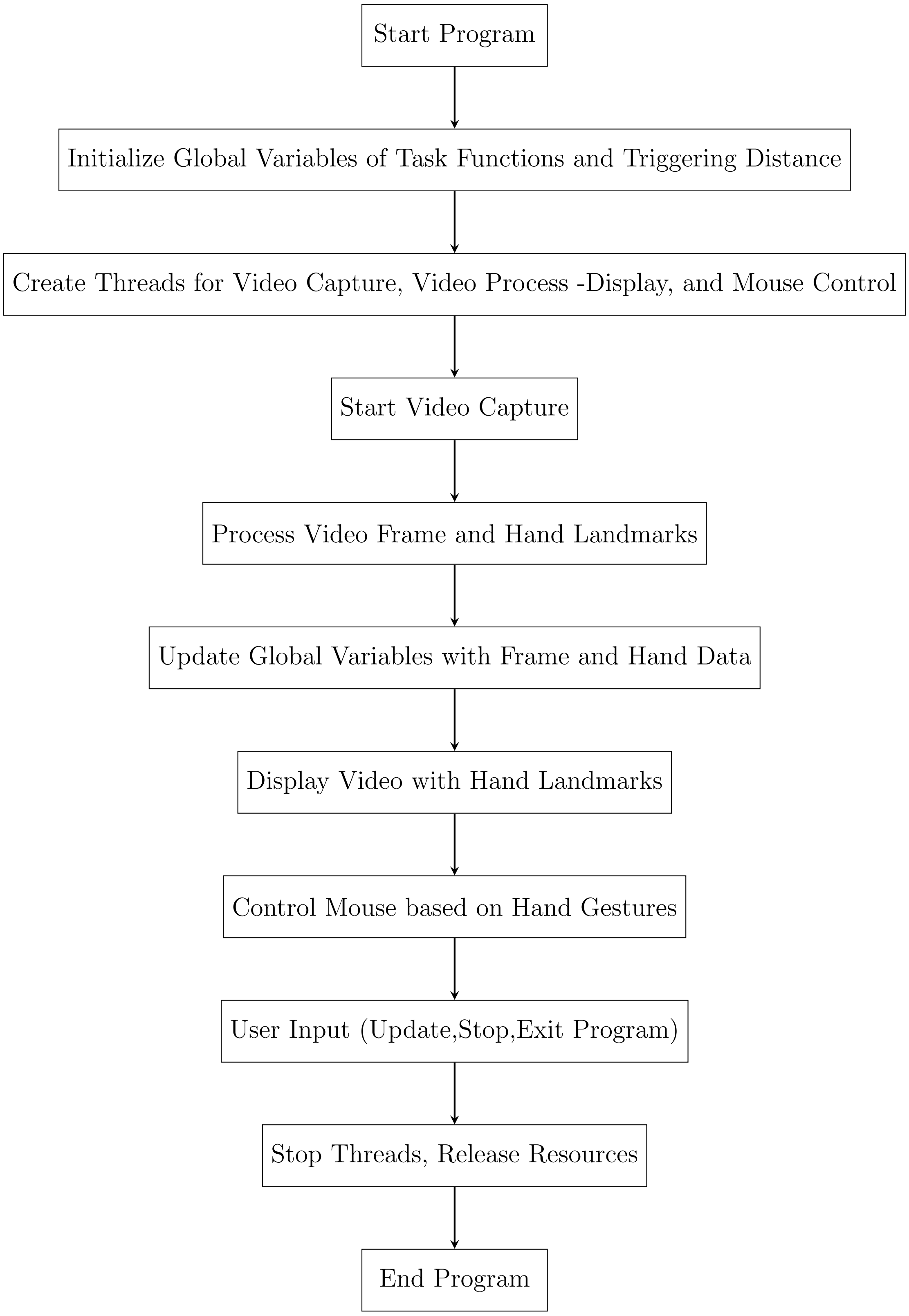
```
class Control Virtual Mouse:
        Initializing Global Variables
        Create objects of three threads to Video Capture,
        Video Process & Take Actions
        initialize Actions_sensitivities ,Actions_arry
        global_start = False

    def startl(self, user_input):
        self.upade(user_input)
        start all three Threads using start method
        return "Virtual Mouse is successfully started"
    def(self,update):
        Actions_sensitivities=sensitivities from user_input
        Actions_array =actions from user_input
        return "Updating the Virtual Mouse Preferences"
    def stop(self):
        stop all three Threads using stop method
        return "Virtual Mouse is successfully stopped"
```

```
class Video Capture:
        Initializing thread_id ,device_id
        self.capture=VideoCapture(device_id
        self.is_Running=True
    def run(self):
                global_start=True
                while self.is_Running=True:
                        global_frame= read(self.capture)
                        globa_frame_time=get current time()
    def stop(self):
        self.is_Running=False
        return "Camera is successfully stopped"
```

```
class Video Process:
        Initializing thread_id, device_id
        self.is_Running=True
    def run(self):
        while self.is_Running=True and global_start=True do
          get hands from global_frame
             get hand from hands
                 if hand not = NULL
                   mouse_start=1
                   get index, x1 & y1 from hand.landmark
                   adjust x1, y1 according to display size
                        if index=9 then:
                          draw circle with center(x,y)
                          x,y=x1,y1 # global var.
                        for index in range(0,20)do
                          mouse[index]=y1  # global variable
                ShowVideo(global_frame)
    def stop(self):
        self.is_Running=False
        return "Video Processing is successfully stopped"
```

```
class Taking Actions:
   Initializing thread_id
   self.acton=[storing required action like click, scroll]
   self.is_Running=True
   def run(self):
   while self.is_Running=True and global_start=True do
         if mouse_start=True and x,y within the display then
           move cursor to (x,y) position
         if absolute value(mouse[4]-mouse[8])<
         Actions_sensitivities[1] then
           execute self.action[1] (left mouse click)
         if absolute value(mouse[4]-mouse[5])<
         Actions_sensitivities[2] then
           execute self.action[Actions_array[2]]
         if absolute value(mouse[8]-mouse[12])<
         Actions_sensitivities[3] then
           execute self.action[Actions_array[3]]
         if absolute value(mouse[4]-mouse[20])<
         Actions_sensitivities[4] then
           execute self.action[Actions_array[4]]
        # We can add more actions manually or using loop
    def stop(self):
        self.is_Running=False
        return "Take Actions thread is successfully stopped"
```

2.3 Flow Chart of Virtual Mouse


Start Program
Initialize Global Variables of Task Functions and Triggering Distance
Create Threads for Video Capture, Video Process -Display, and Mouse Control
Start Video Capture
Process Video Frame and Hand Landmarks
Update Global Variables with Frame and Hand Data
Display Video with Hand Landmarks
Control Mouse based on Hand Gestures
User Input (Update,Stop,Exit Program)
Stop Threads, Release Resources
End Program

Chapter 3

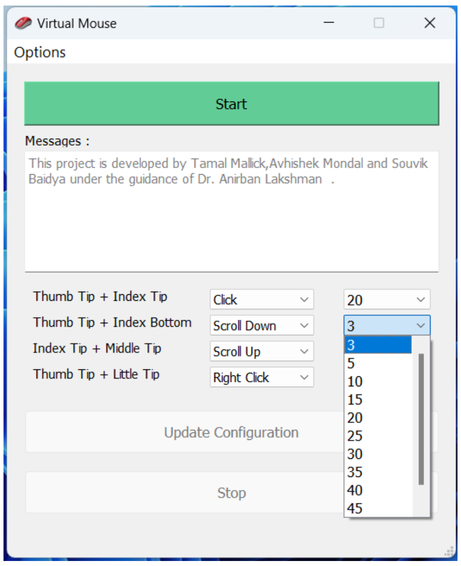
3 Features of Virtual Mouse

The program is being controlled by GUI; an application program has been developed for this purpose. Functionality are Listed below:

Moving Cursor
Left Click
Right Click
Scroll Up
Scroll Down
Tab Change
Open Window Change
Minimize all Tabs
Collapse a Window
Press Enter
Open On-Screen keyboard
Open File Manager
Screen Lock
Some Game Controls

3.1 Function 1 (Cursor Moving)

Hand landmark 9 is used for moving the cursor as it is a centred as well as one of the stable node in human hand. Mouse pointer will move according to the movement of landmark 9.

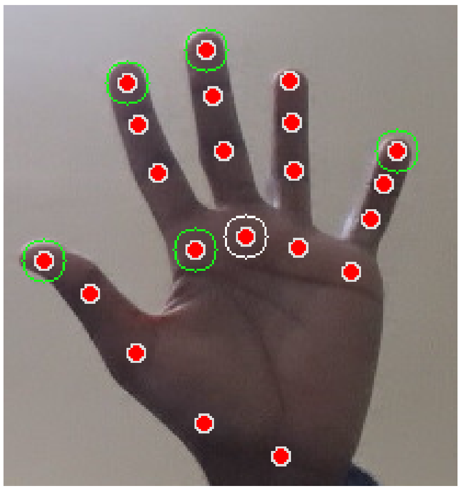


Messages :

This project is developed by Tamal Mallick,Avhishek Mondal and Souvik Baidya under the guidance of Dr. Anirban Lakshman .

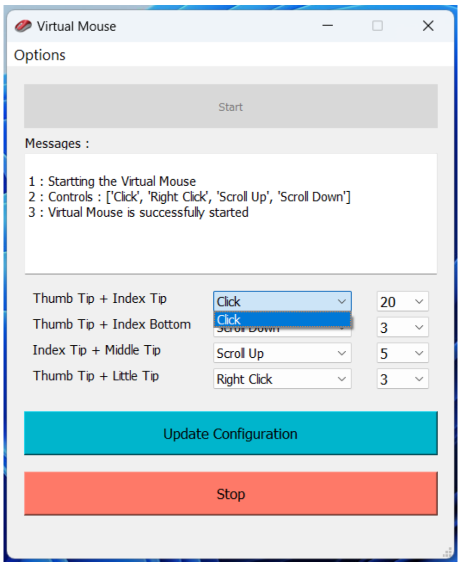
Thumb Tip + Index Tip	Click	20
Thumb Tip + Index Bottom	Scroll Down	3
Index Tip + Middle Tip	Scroll Up	
Thumb Tip + Little Tip	Right Click	

```
3
5
10
15
20
25
30
35
40
45
```



3.2 Function 2 (Click)

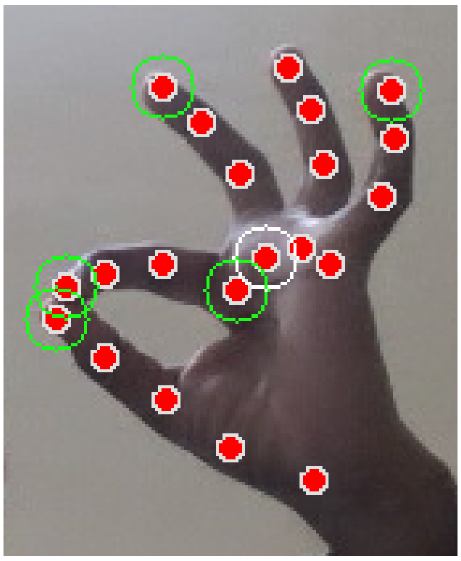
Bringing hand landmarks 4 (Tip of the Thumb) and 8 (Tip of the Index Finger) within a certain distance(triggering distance) activates function 2 which perform mouse click (Left Click) operation. As most of the functions can be changed and updated at any moment of program runtime so kept it as a constant so that the virtual mouse can be operated efficiently .



Messages :

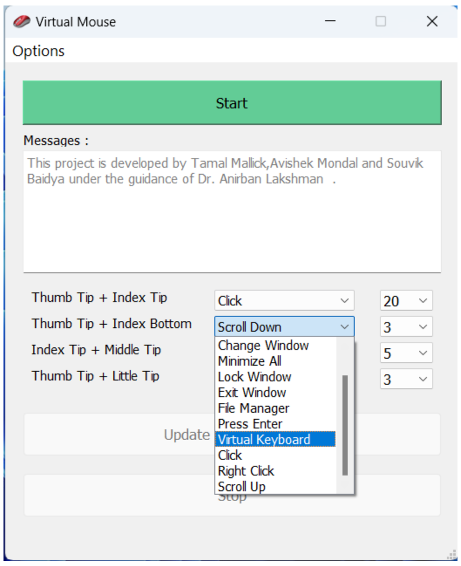
1 : Startting the Virtual Mouse
2 : Controls : ['Click', 'Right Click', 'Scroll Up', 'Scroll Down']
3 : Virtual Mouse is successfully started

Thumb Tip + Index Tip	Click	20
Thumb Tip + Index Bottom	Click	3
Index Tip + Middle Tip	Scroll Up	5
Thumb Tip + Little Tip	Right Click	3



3.3 Function 3 (Selectable)

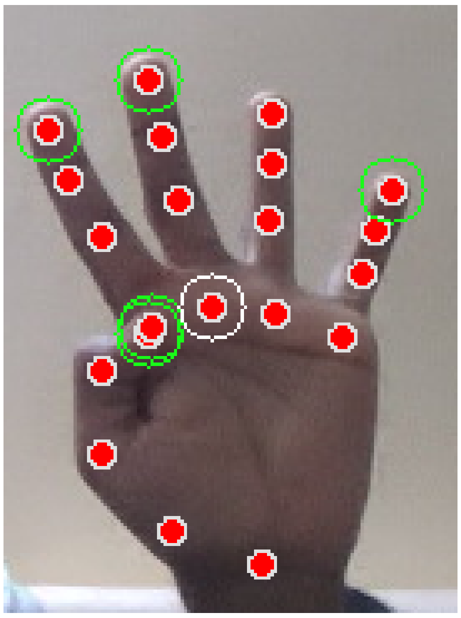
Bringing hand landmarks 4 (Tip of Thumb) and 5 (Bottom of the Index Finger) within a certain distance(triggering distance) activates function 3 which can be used to perform any one of the **Left Click ,Right Click ,Scroll Up,Scroll Down,Tab Change,Open Window Change,Minimize all Tabs,Open File Manager,Collapse a Window ,Press Enter ,Open On-Screen keyboard ,Screen Lock** any one of the task at a time. The **task** and **triggering distance** assigned to the function 3 **can be changed and updated at any moment of time using its user interface** and click method assigned to function 2 .This feature makes the virtual mouse highly dynamic and useful.



Messages :

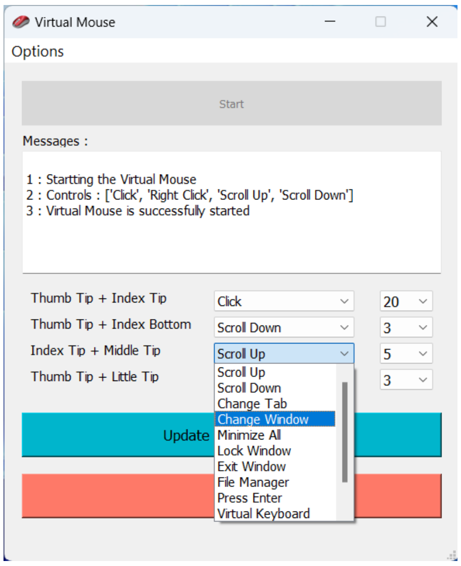
This project is developed by Tamal Malick,Avishek Mondal and Souvik Baidya under the guidance of Dr. Anirban Lakshman .

Thumb Tip + Index Tip	Click	20
Thumb Tip + Index Bottom	Scroll Down	3
Index Tip + Middle Tip		5
Thumb Tip + Little Tip		3



3.4 Function 4 (Selectable)

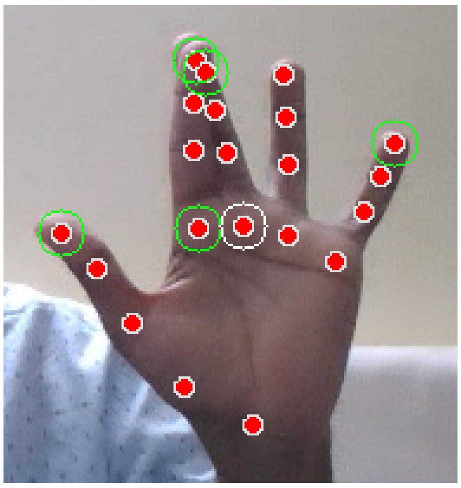
Bringing hand landmarks 8 (Tip of the Index Finger) and 12 (Tip of the Middle Finger) within a certain distance(triggering distance) activates function 4 which can be used to perform any one of the **Left Click ,Right Click ,Scroll Up,Scroll Down,Open File Manager,Tab Change,Open Window Change,Minimize all Tabs,Collapse a Window ,Press Enter ,Open On-Screen keyboard ,Screen Lock** any one of the task at a time. The **task** and **triggering distance** assigned to the function 3 **can be changed and updated at any moment of time using its user interface** and click method assigned to function 2 .This feature makes the virtual mouse highly dynamic and useful.



Messages :

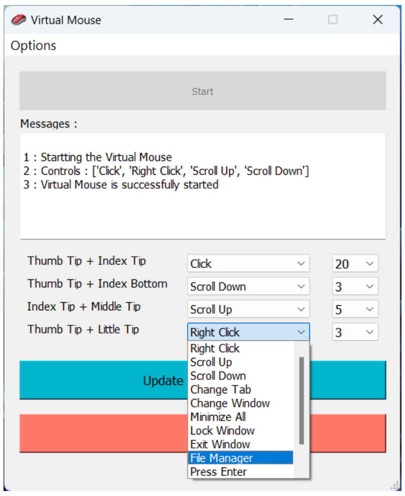
1 : Startting the Virtual Mouse
2 : Controls : ['Click', 'Right Click', 'Scroll Up', 'Scroll Down']
3 : Virtual Mouse is successfully started

Thumb Tip + Index Tip	Click	20
Thumb Tip + Index Bottom	Scroll Down	3
Index Tip + Middle Tip	Scroll Up	5
Thumb Tip + Little Tip		3



3.5 Function 5 (Selectable)

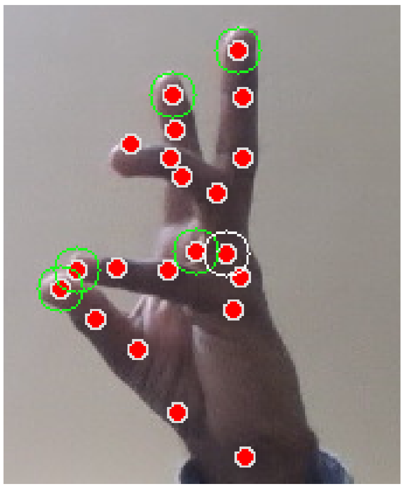
Bringing hand landmarks 4 (Tip of Thumb) and 20 (Tip of the smallest Finger) within a certain distance(triggering distance) activates function 5 which can be used to perform any one of the **Left Click ,Right Click ,Scroll Up,Scroll Down,Open File Manager,Tab Change,Open Window Change,Minimize all Tabs,Collapse a Window ,Press Enter ,Open On-Screen keyboard ,Screen Lock** any one of the task at a time. The **task** and **triggering distance** assigned to the function 3 **can be changed and updated at any moment of time using its user interface** and click method assigned to function 2 .This feature makes the virtual mouse highly dynamic and useful.



Messages :

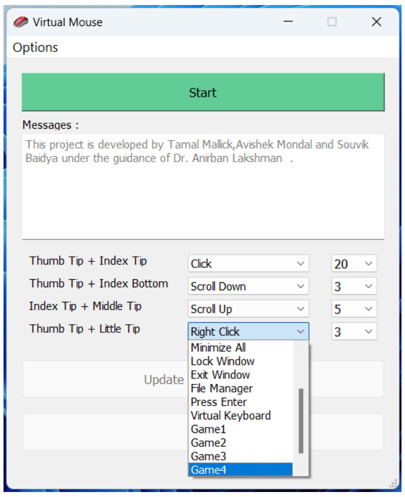
1 : Startting the Virtual Mouse
2 : Controls : ['Click', 'Right Click', 'Scroll Up', 'Scroll Down']
3 : Virtual Mouse is successfully started

Thumb Tip + Index Tip	Click	20
Thumb Tip + Index Bottom	Scroll Down	3
Index Tip + Middle Tip	Scroll Up	5
Thumb Tip + Little Tip	Right Click	3



3.6 Others

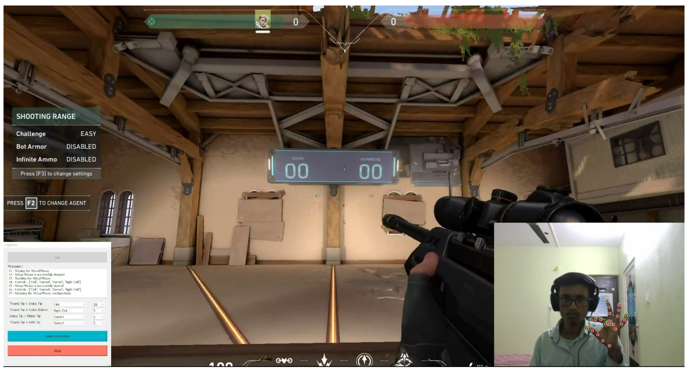
The program supports dynamic task execution, easily accommodating the addition of numerous actions like 'W,' 'S,' 'A,' 'D,' 'X,' 'R,' 'E,' and more for game control. It offers start, update, and stop options, and can be halted with **Shift + C**. Additionally, users can set triggering distances for functions 2 to 5 via the interface.



Messages :

This project is developed by Tamal Mallick,Avishek Mondal and Souvik Baidya under the guidance of Dr. Anirban Lakshman .

Thumb Tip + Index Tip	Click	20
Thumb Tip + Index Bottom	Scroll Down	3
Index Tip + Middle Tip	Scroll Up	5
Thumb Tip + Little Tip	Right Click	3



Gaming

In gaming, we are using Game mode 1 to 4 to perform activities like reloading and buying guns,shooting,using in game utilities in a video game, for our testing we have used VALORANT . We can assign different functions to these game modes by changing the control keys in the game or in program itself.For more information please visit ,

- `https://youtu.be/2ky4_TPi58M` or,
- `https://drive.google.com/drive/folders/1QC3p6UVEPqDbIh9Pla0QXFhP_69HhJ_O?usp=sharing`

to access a video Example of Game playing using this Virtual Mouse Technology.

Chapter 4

4 Performance Analysis

In the proposed AI virtual mouse system, the concept of advancing the Human-Computer interaction using computer vision is given. The hand gestures and finger-tips detection have been tested in various luminous conditions and also been tested from various distance from the different distance and different hand gesture. The test was performed 25 times 8 rounds by each person total of three and this test has been done from different distance and different light condition. Each person tested,

- **7 times in normal light condition** (Far distance, near distance, long distance),
- **5 times in faint condition** (Far distance, near distance, long distance),
- **5 times in dark condition**(Far distance, near distance, long distance).

4.1 Test Results of Functionalities

Mouse Function	**Success**	**Failure**	**Accuracy**
Mouse movement	90	10	90%
Left click	80	20	80%
Right click	70	30	70%
Scroll Down	80	20	80%
Scroll Up	80	20	80%
Others	75	25	75%

Table 2: Test Results of Functionalities

From the Table it can be seen proposed Virtual mouse system had achieved an accuracy of about 79

4.2 Test Results in Different Conditions

Case Id	**Scenario**	**Boundary Value**	**Expected Result**	**Result**
1	Used in normal environment	80%	In normal environment Hand gesture can be recognized easily	Passed
2	Used in Bright Light	70%	In bright brighter environment, software should work fine as it easily detects hand	Passed
3	Used in Dark environment	60%	In a dark environment, it may not work properly	Passed
4	Used at a near distance (15 cm)	60%	At short distance software should perform perfectly	Passed
5	Used a far distance (35 cm)	80%	At this distance, software should perform fine	Passed
6	Used further distance from camera	75%	Far distance, there will be some problem in detecting hand gesture	Passed

Table 3: Test Results in Different Conditions

Chapter 5

5 Conclusion and Future Scope

5.1 Conclusion

The virtual mouse project successfully demonstrates the feasibility of using hand gestures as an alternative to traditional mouse input. The project effectively integrates OpenCV, Media Pipe, and PyAutoGUI to achieve real-time hand detection, tracking, and cursor control. The implemented features, including cursor movement, clicking, scrolling, and brightness/volume control, provide a versatile and intuitive user experience.

The virtual mouse technology offers several advantages over traditional mouse input, including accessibility for individuals with disabilities, ergonomic benefits to reduce strain on wrists and hands, and portability for seamless transitions between devices.

5.2 Future Scope

The virtual mouse project presents promising avenues for future research and development:

- Enhance Hand Gesture Recognition: Continuously improve the accuracy and robustness of hand gesture recognition, particularly for complex and subtle gestures.

- Expand Feature Set: Incorporate additional features, such as gesture-based commands, application-specific interactions, and support for multiple users.

- Explore Integration with VR/AR: Investigate the integration of virtual mouse technology with virtual reality (VR) and augmented reality (AR) environments for enhanced user immersion and interaction.

- Mobile Application Development: Develop a mobile application version of the virtual mouse, leveraging smartphone cameras and gesture recognition capabilities.

- Enhanced Immersive Experiences: Virtual mouse technology has the potential to enhance gaming experiences by offering more immersive and interactive gameplay. Players can use hand gestures or motions to control elements within the game, providing a more natural and engaging gaming environment.

- Real-world Implementation and Testing: Conduct extensive real-world testing and user evaluation to gather feedback and refine the virtual mouse technology for practical applications.

The virtual mouse technology holds immense potential to revolutionize human-computer interaction, making computers more accessible, ergonomic, and intuitive for users worldwide. As technology continues to evolve, we can expect to see its adoption in various domains, from personal computing to healthcare, education, and entertainment.

Chapter 6

Bibliography

[1] C. Chen, Y. Chen, and C. Lin. Real-time hand gesture recognition for virtual mouse control using mediapipe and pyautogui. arXiv preprint arXiv:2111.15460, 2021.

[2] L. Cheng, X. Chen, and J. Luo. Hand gesture recognition for human-computer interaction: A survey. In *Advances in Artificial Intelligence and Intelligent Systems*, pages 519–531. Springer, Singapore, 2022.

[3] A. Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2017.

[4] PyImageSearch. Hand gesture recognition using mediapipe in python.

[5] Real Python. Controlling the mouse with hand gestures using python.

[6] R. Rojas. *Building Intelligent Systems with Python: A Hands-On Approach*. PACKT Publishing, 2022.

[7] A. Silberschatz, P. B. Galvin, and G. Gagne. *Operating System Concepts*. 2020.

[8] Sachin Singh. Hand gesture recognition for virtual mouse control. `https://medium.com/@sachin.singh.professional/ai-virtual-mouse-using-hand-gesture-recognition-%EF%B8%8F-cd189d5bc824`.

[9] A. S. Tanenbaum. *Modern Operating Systems*. 2019.