Python Based VR Mouse

[1]Hari Haran R,[2]Shine Raj,[3]Anbarasu Y, [4]Nalini Poornima, [5]V. Ramesh Babu, [6]M. Anand

[1,2,3,] IV Year B. Tech CSE Students,[4]Asst. Professor, [5,6]Professor
[1,2,3,4,5,6]Dept of Computer Science and Engineering,
[1,2,3,4,5,6]DR MGR Educational and Research Institute, Maduravayol, Chennai-95, Tamil Nadu, India
1anbarasu1999@gmail.com, [2]shineraja1plus@gmail.com, [3]hariharan021217@gmail.com
nalinipoornima.cse@drmgrdu.ac.in,rameshbabu.cse@drmgrdu.ac.in

Abstract

The use of hand gesture recognition in controlling virtual devices has become popular due to the advancement of artificial intelligence technology. A hand gesture-controlled virtual mouse system that utilizes AI algorithms to recognize hand gestures and translate them into mouse movements is proposed in this paper. The system is designed to provide an alternative interface for people who have difficulty using a traditional mouse or keyboard. The proposed system uses a camera to capture images of the user's hand, which are processed by an AI algorithm to recognize the gestures being made. The system is trained using a dataset of hand gestures to recognize different gestures. Once the gesture is recognized, it is translated into a corresponding mouse movement, which is then executed on the virtual screen. The system is designed to be scalable and adaptable to different types of environments and devices. All the input operations can be virtually controlled by using dynamic/static hand gestures along with a voice assistant. In our work we make use of ML and Computer Vision algorithms to recognize hand gestures and voice commands, which works without any additional hardware requirements.

1. INTRODUCTION

Gesture Controlled Virtual Mouse makes human-computer interaction simple by making use of Hand Gestures and Voice Commands. The computer requires almost no direct contact. All i/o operations can be virtually controlled by using static and dynamic hand gestures along with a voice assistant. This project makes use of state-of-art Machine Learning and Computer Vision algorithms to recognize hand gestures and voice commands, which works smoothly without any additional hardware requirements. It leverages models such as CNN implemented by Media Pipe running on top of pybind11. It consists of two modules: One which works direct on hands by making use of Media Pipe Hand detection, and the other which makes use of Gloves of any uniform colour. Currently, it works on the Windows platform

Key Components of the Project:
Creating a dataset for hand recognition for mouse control typically involves collecting images or video clips of hands in various poses and positions to train a machine learning model. Here's a simplified outline of how you might create such a dataset:

.

The aim of a hand recognition mouse project is to develop a system that allows users to control a computer's mouse cursor and perform various actions using hand gestures and movements. This technology aims to enhance user interaction with computers and devices by providing a natural, intuitive, and touchless interface. Here are the primary objectives and goals of such a project

• Gesture-Based Control: Develop a system that recognizes and interprets a wide range of hand gestures and movements, allowing users to navigate the computer interface, control applications, and perform actions like clicking, dragging, scrolling, and zooming..

• Enhanced Accessibility: Design the hand recognition mouse to be an assistive technology tool that enhances accessibility for individuals with physical disabilities or limitations, providing an alternative means of computer interaction..
• Natural User Interface: Create a user-friendly and intuitive interface that leverages natural hand movements, reducing the reliance on traditional input devices like a physical mouse or keyboard.
• Promote the utilization of open-source technologies to lower development costs and encourage community contributions

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

2. RELATEDWORKS

Real-time Hand Gesture Recognition for Human-Computer Interaction, Author Li, Yang, et al.(2014) This review article provides an overview of various hand gesture recognition techniques and their applications in human-computer interaction, which could be relevant to hand recognition mouse control. A Survey of Hand Gesture Recognition Techniques , Author Suryanaraya (2016) This survey paper provides a comprehensive overview of hand gesture recognition methods, which can be useful for understanding the techniques used in hand recognition mouse systems. Real-Time Hand Gesture Recognition Based on Depth Data ,Author Xiaoguang (2018) This paper discusses depth-based hand gesture recognition techniques, which are often used in 3D motion-sensing devices for mouse control. Vision-Based Hand Gesture Recognition for Human-Computer Interaction Author Murad, Mohammad et al.(2020) This review article focuses on vision-based hand gesture recognition techniques and their applications in human-computer interaction, which may include hand recognition mouse A Survey on Hand Gesture Recognition Techniques, Systems, and Applications Authors Pu, Zhang, et al. (2018) This survey paper provides insights into various hand gesture recognition systems and their potential applications, including computer mouse control.

As of my last knowledge update in September 2021, several existing systems and technologies incorporated hand recognition for mouse control and other human-computer interaction applications. leap Motion: Leap Motion is a hand tracking device that uses infrared sensors to accurately capture hand and finger
movements in 3D space. It has been used for various applications, including virtual reality (VR), gaming, and computer interaction.
Microsoft Kinect: The Kinect sensor, initially developed for the Xbox gaming console, features depth-sensing technology and can recognize hand gestures. It has been used in gaming, robotics, and research applications..

RealSense:

There are some related works carried out on virtual mouse using hand gesture detection by wearing a glove in the hand and also using color tips in the hands for gesture recognition, but they are no more accurate in mouse functions. The recognition is not so accurate because of wearing gloves; also, the gloves are also not suited for some users, and in some cases, the recognition is not so accurate because of the failure of detection of color tips. Some efforts have been made for camera-based detection of the hand gesture interface.

In 1990, Quam introduced an early hardware-based system; in this system, the user should wear a DataGlove [2]. The proposed system by Quam although gives results of higher accuracy, but it is difficult to perform some of the gesture controls using the system.

Dung-Hua Liou, ChenChiung Hsieh, and David Lee in 2010 [3] proposed a study on “A Real-Time Hand Gesture Recognition System Using Motion History Image.” The main limitation of this model is more complicated hand gestures.

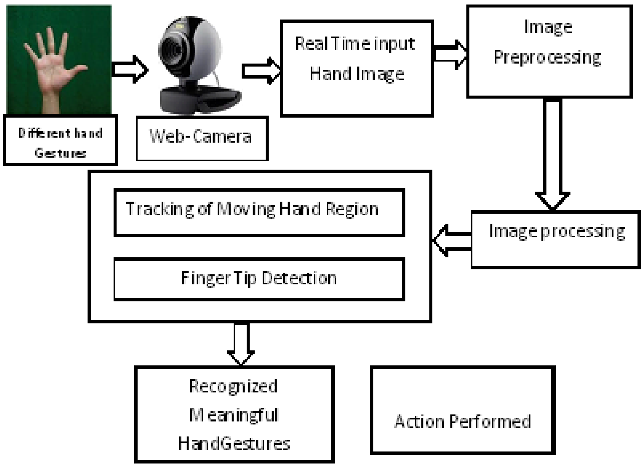
Monika B. Gandhi, Sneha U. Dudhane, and Ashwini M. Patil in 2013 [4] proposed a study on "Cursor Control System Using Hand Gesture Recognition." In this work, the limitation is stored frames are needed to be processed for hand segmentation and skin pixel detection

Vinay Kr. Pasi, Saurabh Singh, and Pooja Kumari in 2016 [5] proposed "Cursor Control using Hand Gestures" in the IJCA Journal. 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.

Chaithanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS in 2018 [6] proposed "Virtual Mouse Using Hand Gesture" where the model detection is based on colors. But, only few mouse functions are performed.

3. SYSTEMMETHODOLOGY

We presented the block diagram and flow chart of Virtual Mouse Control Using Hand Gesture Recognition in this section. We also provided a brief description of the system's operation and components.



SYSTEM IMPLEMENTATION:
Hardware Setup:

• Acquire Hardware Components: Procure the depth-sensing camera or sensor, processing unit (CPU/GPU), power supply, and any other necessary hardware.

• Mounting and Positioning: Install the depth-sensing camera in a suitable location, ensuring it has an unobstructed view of the user's hand. Proper positioning is essential for accurate hand tracking.

• Connect Hardware: Connect all hardware components, including the camera, processing unit, power supply, and any additional devices (e.g., wireless modules).

2. Software Development:

a. Choose Development Tools: Select the programming languages, frameworks, and development tools for software implementation. Popular choices include Python, C++, and libraries like OpenCV or TensorFlow.

b. Hand Tracking Implementation: Develop the hand tracking module, which captures depth data from the camera and identifies the user's hand in 3D space. Implement algorithms for hand detection and tracking.

c. Gesture Recognition: Create the gesture recognition module that 12nalyses hand movements and recognizes predefined gestures (e.g., cursor movement, clicking, scrolling). Train the system using a 13 dataset of hand gestures

d. User Interface (UI): Design and implement the user interface (UI) that allows users to configure settings, perform calibration, and receive feedback on recognized gestures. Develop UI components for gesture customization and system control.

e. Integration with Operating System: Integrate the software with the target operating system to control the mouse cursor and interact with applications. Implement the necessary drivers or APIs for seamless interaction.

f. Calibration and Personalization: Develop a calibration process that adjusts the system to the user's hand size and position. Implement features for gesture customization and personalization.

3. Testing and Validation:

a. Unit Testing: Test individual software components (hand tracking, gesture recognition, UI) to ensure they function correctly.

b. Integration Testing: Verify that all software modules work together seamlessly, simulating real-world usage.

c. User Testing: Conduct user testing with a diverse group of participants to gather feedback on usability, accuracy, and overall performance.

d. Performance Testing: Evaluate the system's responsiveness, accuracy, and resource usage in different scenarios and environments.

4. Calibration and User Setup:

a. Develop a user-friendly calibration process that guides users through aligning the system with their hand size and preferences.

b. Provide clear instructions and visual cues to assist users in completing the calibration successfully.

5. User Documentation:

a. Create comprehensive user guides and tutorials that explain how to set up, calibrate, and use the hand recognition mouse system effectively.14

6. Deployment:

a. Prepare the hand recognition mouse system for deployment on target devices or platforms. Ensure compatibilists a range of operating systems.

b. Address any compatibility issues that may arise during deployment.

7. Continuous Improvement:

a. Commit to providing software updates to enhance system performance, add new features, and address user feedback.

b. Stay informed about emerging technologies and trends in hand recognition and gesture control for potential system enhancements.

8. Ethical Considerations:

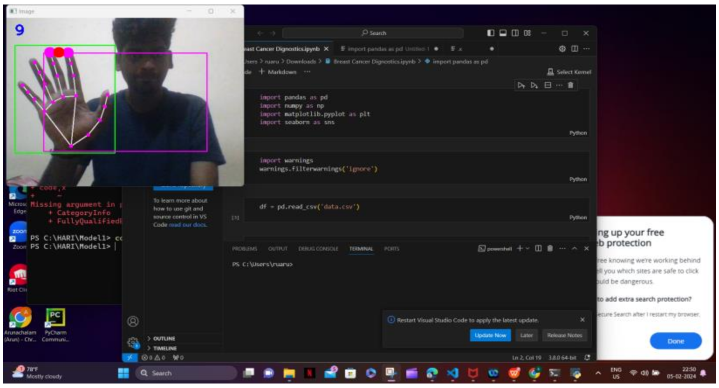
a. Ensure that the system is developed and used in accordance with ethical guidelines, respecting user privacy, data security, and consent.

9. Commercialization (Optional):

a. If the hand recognition mouse system is intended for commercial use, develop a marketing and sales strategy to introduce it to consumers or businesses**.**

Remember that iterative development and user feedback are essential throughout the implementation process to fine-tune the system and ensure that it meets performance, usability, and reliability standards.

Additionally, addressing ethical considerations and privacy protection is crucial in the development and deployment of such systems**.**



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
df = pd.read_csv('data.csv')
```

```
+ code,x
+    ~
Missing argument in p
    + CategoryInfo
    + FullyQualifiedE

PS C:\HARI\Model1> cc
PS C:\HARI\Model1> |
```

```
PS C:\Users\ruaru>
```

To learn more about how to use git and source control in VS Code read our docs.

Restart Visual Studio Code to apply the latest update.

ng up your free b protection

ree knowing we're working behind ll you which sites are safe to click uld be dangerous.

to add extra search protection?

ecure Search after I restart my browser.

Figure1.hand recognition

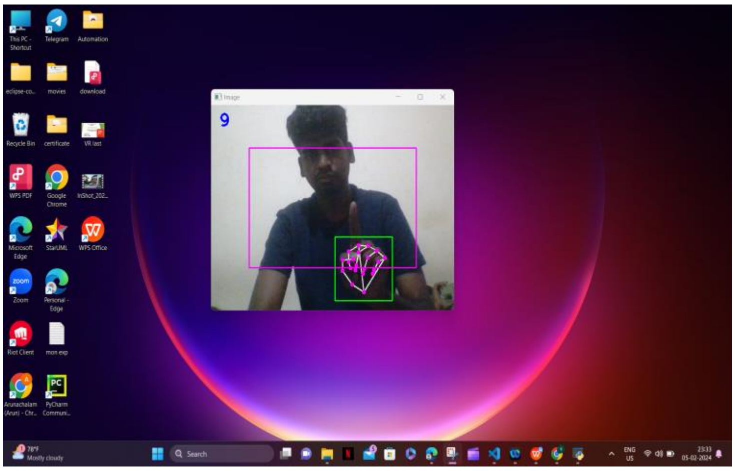


Figure2.finger recognition

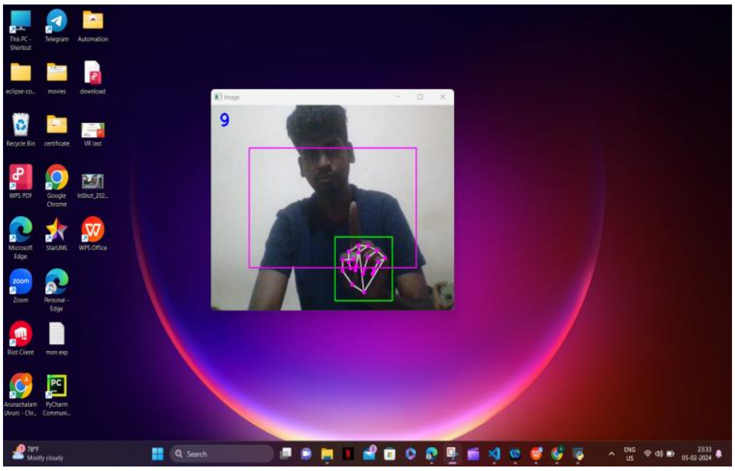


Figure3.cursor movement

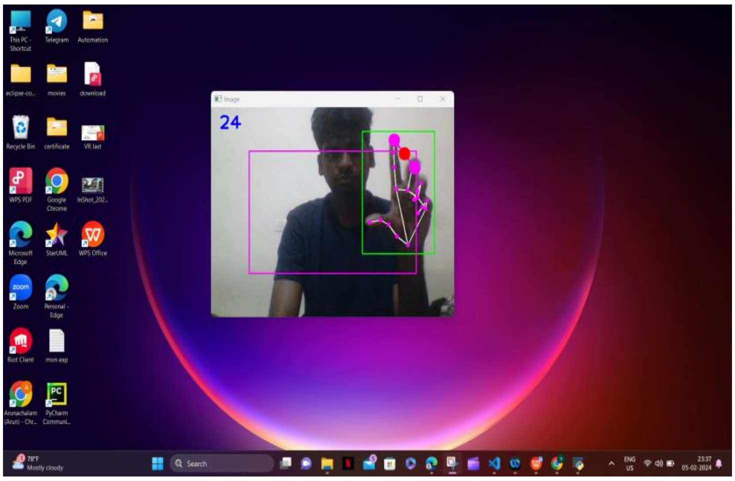


Figure4.click

SYSTEM ARCHITECTURE

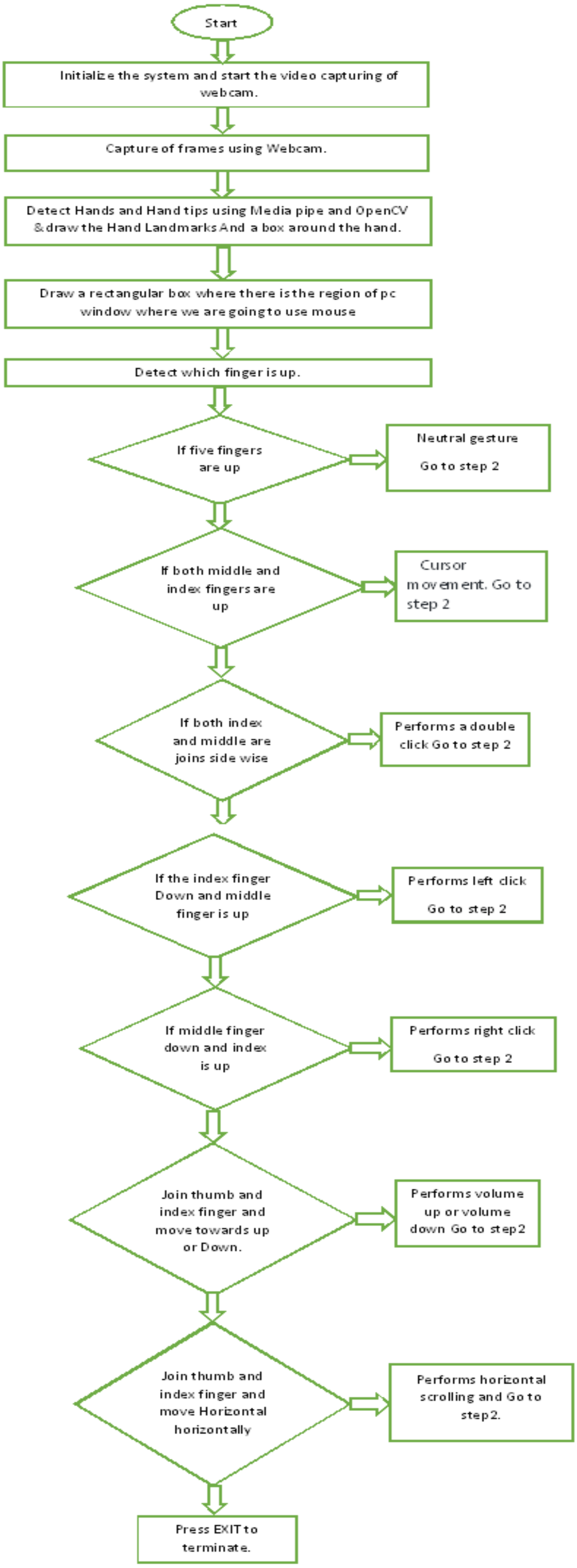

Start
Initialize the system and start the video capturing of webcam.
Capture of frames using Webcam.
Detect Hands and Hand tips using Media pipe and OpenCV & draw the Hand Landmarks And a box around the hand.
Draw a rectangular box where there is the region of pc window where we are going to use mouse
Detect which finger is up.
If five fingers are up
Neutral gesture Go to step 2
If both middle and index fingers are up
Cursor movement. Go to step 2
If both index and middle are joins side wise
Performs a double click Go to step 2
If the index finger Down and middle finger is up
Performs left click Go to step 2
If middle finger down and index is up
Performs right click Go to step 2
Join thumb and index finger and move towards up or Down.
Performs volume up or volume down Go to step 2
Join thumb and index finger and move Horizontal horizontally
Performs horizontal scrolling and Go to step 2.
Press EXIT to terminate.

Figure5.Architecture Diagram

4. RESULT AND DISCUSSION

For testing this algorithm Computer is set not to perform any
mouse actions on the screen. To perform No Action on the Screen, as shown in Fig-4.1, if all of the fingers are up with tip Id = 0, 1, 2, 3, and 4, the computer is set to not perform any mouse events on the screen. To use the mouse's right-button click. The computer is programmed to perform the right mouse button click 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 less than 40 px, as shown in figure 2. For navigating the computer window with the mouse cursor. The mouse cursor is made to move around the computer window using Python's AutoPy package if the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up, as shown in figure 3.

To perform a left-button click with the mouse. The computer is made to perform the left mouse button click using the ynput Python package 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 less than 30px, as shown in Fig-4.

Finally, the images above show the various mouse operations that can be carried out with hand gestures. Recognizing different fingertip ids allows you to perform various mouse operations. The aforementioned mouse functions can be performed even in low-light conditions.

5. 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. Hence, we will work next to overcome these limitations by improving the finger tip detection algorithm to produce more accurate result

REFERENCES

[1] S Guennouni, A Ahaitouf and A Mansouriss , “Multiple object detection using Open CV on an embedded platform”, 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), 2014, pp. 374-377..

[2] G. Chandan, Mohana A.H Jain “The Real Time Object Detection and Tracking Using Deep Learning and OpenCV”, 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), 2018, pp. 1305-1308.

[3] C. Chua, H. Guan, and Y. Ho. Model-based 3d hand posture estimation from a single 2d image. Image and Vision Computing, 20:191–202, 2002.

[4] Munir Oudah, Ali al-Naji, Javaan Chahl Hand Gesture Recognition Based on Computer Vision: A Review of Techniques,2020.

[5] Yoichi Sato, Kenji Oka, Hideki Koike, Yasuto Nakanishi, Video-Based Tracking of User's Motion for Augmented Desk Interface. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004.