

Mouse Cursor Control Using Hand Gesture

Neelam Jadhav¹, Akshay Pujari², Swami Wardule³, Mayur Patil⁴, Ravi Bele⁵, Disha Kate⁶

¹Professor, Department of Computer Engineering, Genba Sopanrao Moze, Pune, India
^{2,3,4,5,6}Student, Department of Computer Engineering, Genba Sopanrao Moze, Pune, India

ABSTRACT

The mouse is one of the wonderful inventions of Human-Computer Interaction (HCI) technology. In today's tech world, it's crucial to find new ways to interact with computers as their use grows. Many devices use touch screens, but they're not suitable for all applications. An alternative is a virtual mouse using Hand Tracking and Gestures. The goal is to create an application that controls the computer cursor through hand movements captured by a webcam. This system, based on Computer Vision, uses the Python programming language and the OpenCV library for real-time visual processing. The webcam captures hand gestures, enabling users to navigate the cursor and perform mouse operations like left-click, right-click, and double-click. The camera output is displayed on the monitor. To get started, set up your environment by installing Python and required libraries like OpenCV. Integrate your webcam into your application and implement hand detection techniques. Recognize different hand gestures and map them to cursor movements and mouse operations. You may also create a simple user interface for better interaction.

Keywords: Virtual mouse, Hand tracking, Webcam, OpenCV, Python, Yolo Algorithm Gestures.

INTRODUCTION

In computing, a mouse serves as a pointer, detecting two-dimensional movements on a surface. This translates into pointer motion on a display, facilitating control of the Graphical User Interface (GUI) on a computer. Over time, diverse mouse types have emerged. The mechanical mouse, utilizing a rubber ball, was succeeded by the optical mouse, replacing the ball with an LED detector for tracking tabletop movements. The 2004 introduction of the ray mouse aimed to refine movement precision, tackling limitations faced by the optical mouse in tracking highly polished surfaces. Yet, despite enhanced accuracy, mice have inherent limitations, both physically and in specialized contexts. For instance, a computer mouse is a consumable, necessitating replacement due to button deterioration or complete undetection by the computer. To address such issues, a virtual mouse, operated through a webcam or analogous image-capturing devices, emerges as a pragmatic alternative. This virtual interface utilizes software to monitor user gestures, translating them into pointer movements akin to a physical mouse. With the dynamic evolution of computer technology, the demand for novel interaction methods is urging. While touch screen technology prevails, it is not universally applicable. An apt solution lies in interactive modules like a virtual mouse, employing Hand Tracking and Gestures for user interaction.

The objective is to devise a Hand Tracking system rooted in Computer Vision, capturing hand gestures through a webcam with Hand detection techniques. This system empowers users to navigate the cursor and execute mouse operations using hand gestures. Python and the OpenCV library facilitate real-time computer vision for system implementation, with the camera output displayed on the monitor. This inventive approach seeks to amplify user-computer interaction in a society where universal feasibility of touchscreen technology is challenging.



LITERATURE SURVEY

Some previous methods for controlling a virtual mouse using hand gestures involved wearing a glove or using colored tips on fingers, but they were not very accurate. Wearing gloves can be uncomfortable for some people, and detecting colored tips can be unreliable.

In the 1990s, Quam introduced a system that required wearing a DataGlove, which was accurate but challenging to use for some gestures.

In 2010, Liou, Hsieh, and Lee proposed a system for real-time hand gesture recognition, but it struggled with complex gestures.

In 2013, Gandhi, Dudhane, and Patil proposed a system for cursor control using hand gestures, but it required processing many frames for hand detection.

In 2016, Pasi, Singh, and Kumari proposed a system using different bands on the hand for mouse functions, but it relied on different colors, which had limitations.

In 2018, Chaithanya C, Lisho Thomas, Naveen Wilson, and Abhilash SS proposed a "Virtual Mouse Using Hand Gesture" model. This model uses hand gestures for detection

METHODOLOGY

The AI virtual mouse system uses the webcam built into a laptop or PC to capture video frames. It uses a Python computer vision library called OpenCV to create a video capture object, which allows the webcam to start capturing video. The webcam captures the video frames, which are then passed to the AI virtual mouse system for processing

The AI virtual mouse system uses an algorithm to convert the coordinates of a fingertip from the webcam screen to the full computer window screen. This allows the system to control the mouse based on the finger movements captured by the webcam.

When the system detects hands and determines which finger is raised to perform a specific mouse function, it draws a rectangular box on the computer window in the webcam region. This box acts as a virtual screen that you can move around to control the mouse cursor on the computer screen

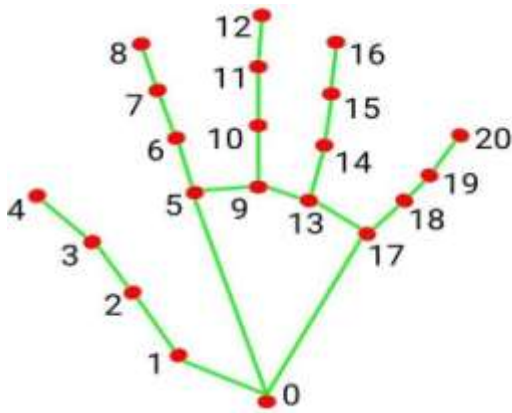
In this stage, the system detects which finger is raised using the tip ID of the respective finger. This information is obtained from the hand tracking model provided by MediaPipe. Based on the coordinates of the raised fingers, the system determines which finger is up.

Once the system knows which finger is raised, it performs a specific mouse function corresponding to that finger. For example, it might move the cursor if the index finger is raised, or perform a click if the thumb is raised.

Mouse Functions Depending on the Hand Gestures and Hand Tip Detection Using Computer Vision

To move the mouse cursor around the computer window, the system checks if the index finger is raised (identified by tip ID 1) or if both the index finger (tip ID 1) and the middle finger (tip ID 2) are raised. If either of these conditions is met, the system uses the AutoPy package in Python to control the mouse cursor's movement on the computer screen

To perform a left mouse button click, the system checks if both the index finger (tip ID 1) and the thumb finger (tip ID 0) are raised. It also checks if the distance between these two fingers is less than 30 pixels. If both conditions are met, the system simulates a left mouse button click on the computer.



- | | |
|-----------------------|-----------------------|
| 0. WRIST | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP | 13. RING_FINGER_MCP |
| 3. THUMB_IP | 14. RING_FINGER_PIP |
| 4. THUMB_TIP | 15. RING_FINGER_DIP |
| 5. INDEX_FINGER_MCP | 16. RING_FINGER_TIP |
| 6. INDEX_FINGER_PIP | 17. PINKY_MCP |
| 7. INDEX_FINGER_DIP | 18. PINKY_PIP |
| 8. INDEX_FINGER_TIP | 19. PINKY_DIP |
| 9. MIDDLE_FINGER_MCP | 20. PINKY_TIP |
| 10. MIDDLE_FINGER_PIP | |

METHODS

In the Methodology, the procedure of each part is being explained step by step and their working.

A. Webcam

The first important functioning tackle is a good working web camera. To capture the object used to instruct the mouse and replacethe working of it with the object. So that formerly if the object is captured in our frames using OpenCV also it can be used as our mouse to make the performing easier. The purpose of the Webcam is landing the hand gesture and movements generated by humans and store its image in memory.

B. Hand Tracking.

The movement of the pointer was controlled by the tip of the finger. so as to spot the tip of the finger, the centre of the palm should first be set up. the strategy used for locating the hand centre was espoused from and it has the advantage of being straightforward and easy to apply. The shortest distance between every point inside the inscribed circle to the contour was measured and the point with the largest distance was recorded as the centre. the space between the centre of hand and the hand contour was taken as the compass of the hand. The hand centre was calculated for every successive frame and using the hand centre, the tip of the finger would be known and used for hand tracking. The strategy used for relating the indicator and also the different fingers are described.

C. Gesture Recognition

The system is trained with hand gestures to fete different gestures. Once the gesture is honoured, it’s restated into a corresponding mouse movement, which is also executed on the virtual screen. The system is designed to be scalable and adaptable to different types of surroundings and devices.

There are two main ways for gesture recognition:

1. Fingertip identification
2. Number of fingers.

D. Cursor Control

Once the hand gestures are honoured, it will be an easy matter of mapping completely different hand gestures to specific mouse functions. It seems that controlling the Personal Computer cursor, operations are performed. The different operations that can beperformed are Right Click, Left Click, Double Click, Real- Time Operation, Pause/ Play any videotape.

RESULT

The virtual mouse system is to develop an alternative to the regular and traditional mouse system to perform and control the mouse functions. The OpenCV library has been used to track the hands that help to operate on this system using the

webcam or a built-in camera. By using OpenCV, the mouse and its basic operations like mouse pointing, selection, clicking and deselection can be controlled.

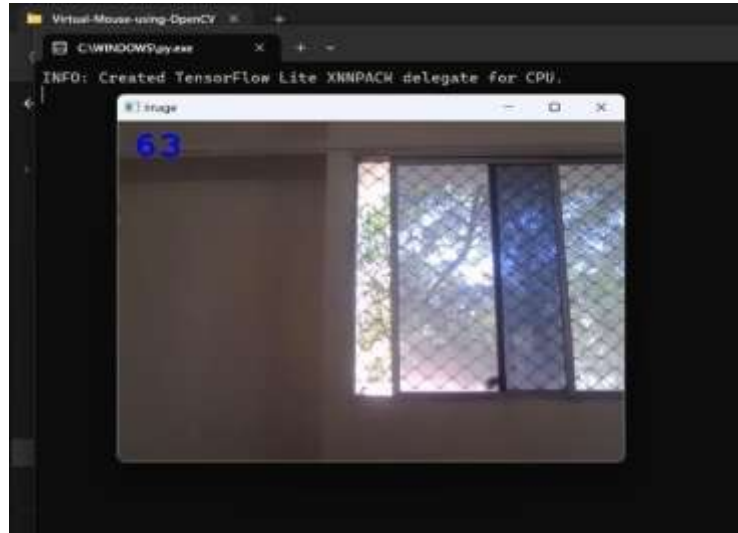


Figure 1: No hand detected

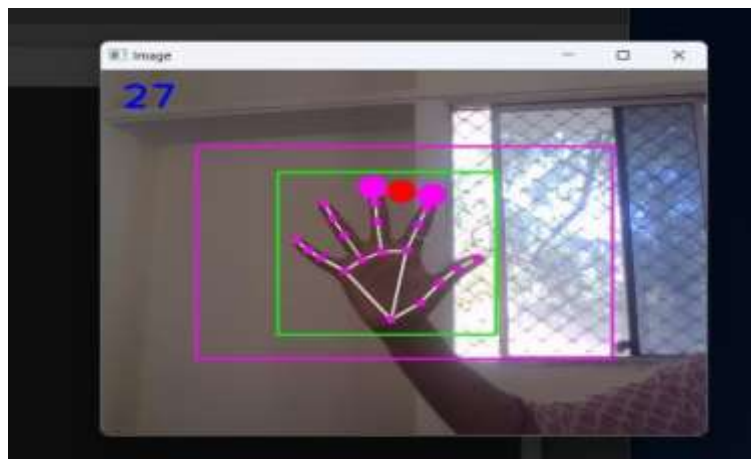


Figure 2: Hand detect

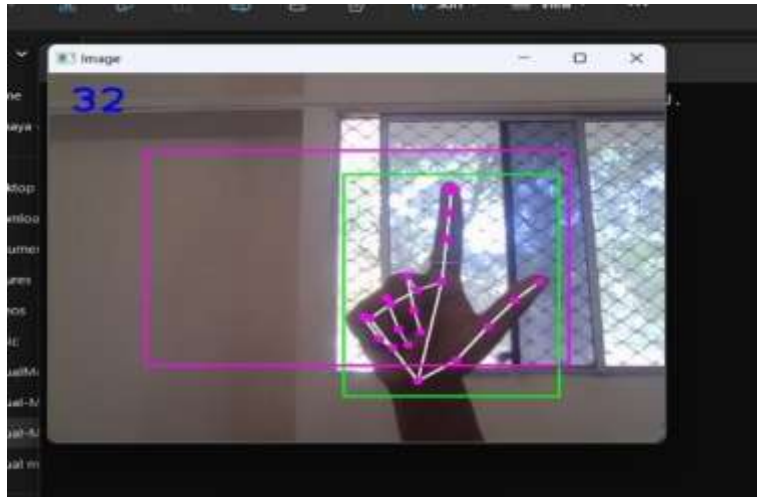


Figure 3: Scrolling Gesture

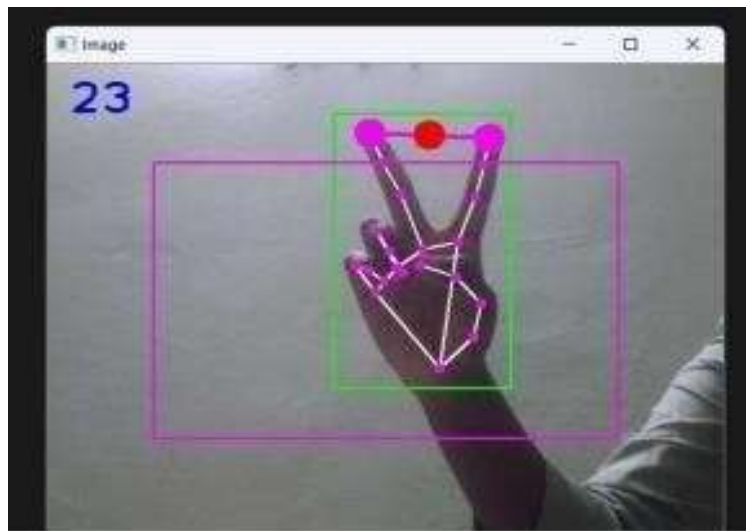


Figure 4: Clicking operation not performed (as distance between fingers is greater than 40)

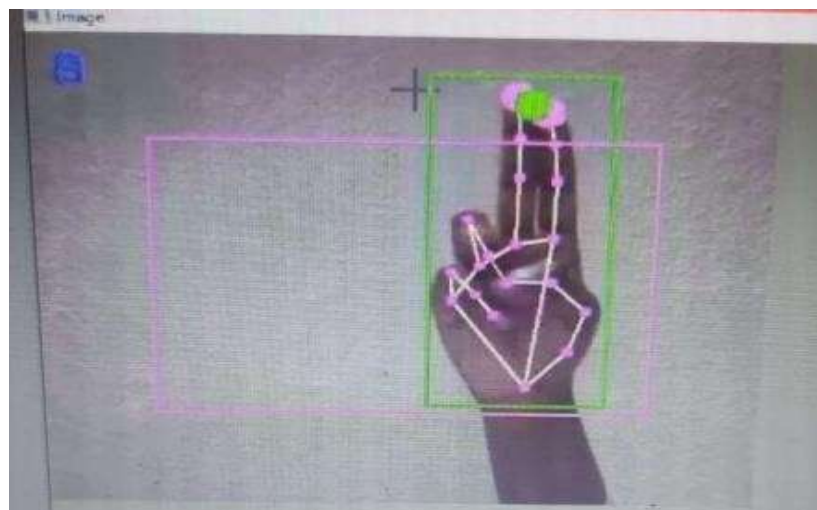


Figure 5: Clicking operation performed (As the distance between fingers is less than 40)

CONCLUSION

The system controls the mouse pointer and executes its operation using a real-time camera. This design is to form a system which will allow the hand gestures and control the computer/ laptop in line with those gestures. The system will benefit mobile systems where using pointing devices like the mouse is tough. All the mouse tasks like left and right clicking, double clicking and beginning the operations using gestures like tablet, maquila, command prompt etc. Before factual enforcing gesture comparison algorithms, skin discovery and hand segmentation from kept frames need to be done. The system is also developed in such a way that the user new to the system can simply have to install the set up and not run the full system.

REFERENCES

- [1]. Monali Shetty, Christina A. Daniel, Manthan K.Bhatkar, Ofrin P. Lopes, “Virtual Mouse Using Object Tracking”, IEEEConference Record # 48766, ISBN: 978-1-7281-5371-1, 2020.
- [2]. Vantukala VishnuTeja Reddy, Thumma Dhyanchand, Galla Vamsi Krishna, Satish Maheshwaram, “Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition” IEEE Xplore Journal HYD-CON, 2020.
- [3]. Onkar Yadav, Sagar Makwana, Pandhari Yadav, Prof. Leena Raut, “Cursor Movement By Hand Gesture”, International Journal Of Engineering Sciences & Research Technology, ISSN: 2277- 9655, 2017.
- [4]. D. A. Barhate and K. P. Rane, "A Survey of Fingertip Character Identification in Open-Air Using Image Processing and HCI," 3rd International Conference for Convergence in Technology (I2CT) , pp. 1-4, 2018.
- [5]. Mahajan J.R and C. S. Rawat, "Object Detection and Tracking using Cognitive Approach," 2017 International Journal of Scientific Research in Network Security and Communication (IJSRNSC), Vol.5, Issue.3, pp. 136-140, 2017.
- [6]. Abhik Banerjee, Abhirup Ghosh, Koustuvmoni Bharadwaj, “Mouse Control using a Web Camera based on Color Detection”,IJCTT, vol.9, 2014.
- [7]. A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, “Computer vision-based mouse”, Acoustics, Speech, and Signal Processing, Proceedings. (ICASS). IEEE International Conference.
- [8]. Hand Detection: Use techniques such as background subtraction, skin color detection, or deep learning-based methods to detect the presence and location of the hand in the video frame.
- [9]. Hand Tracking: Once the hand is detected, use algorithms like CAMShift, meanshift, or more advanced deep learning-based methods to track the movement of the hand over subsequent frames.
- [10]. Gesture Recognition: Define specific hand gestures that correspond to different mouse actions (e.g., moving up/down, left/right, clicking). You can use techniques such as contour analysis, convex hull, or machine learning classifiers to recognize these gestures.