Grenze International Journal of Engineering and Technology, Jan Issue

Virtual Mouse: A Contactless Interaction

Yash Shetty[1], Shreyash Kadlag[2], Shubham Thite[3], Yash Singh[4], Abhishek Kadam[5] and Archana Jadhav[6]
[1-6]Department of Information Technology, JSPM's Rajarshi Shahu College of Engineering, Pune
Email: Email: yashshetty2022@gmail.com

***Abstract*—Many researchers around the world are currently working to improve product interactivity while reducing the need for physical interaction. This research offers an interactive computer system that can function without the usage of a real mouse. While wireless or Bluetooth mice still require a battery and a dongle to connect to a PC, this system is device-free. Users may operate the mouse system via gestures thanks to computer vision technologies. Deep learning techniques are used by the system to facilitate mouse activities via various gestures exhibited by the user while in mouse control mode.**

***Index Terms*— Virtual Mouse, Deep Learning[13], Media-pipe, Human computer interaction, Computer Vision.**

I. INTRODUCTION

As wireless technologies and augmented reality continue to advance, the devices we use daily are becoming smaller. This research introduces a new computer vision-based system that replaces a traditional mouse with hand gestures and hand tip detection, using a web camera or built-in camera on the computer. The aim of the proposed system is to enable users to control the computer's mouse pointer and scrolling functions without needing an external mouse device. The system uses human-computer interaction to identify hand gestures and tips through computer vision. The AI virtual mouse system is built in Python and relies on the OpenCV computer vision package to track hand movements and finger tips. MediaPipe is used for hand tracking, while Pynput, Autopy, and PyAutoGUI are used to move the mouse cursor and perform actions like clicking, scrolling, and drag-and-drop. The system achieves high accuracy and can run on a CPU, making it suitable for real-world applications.

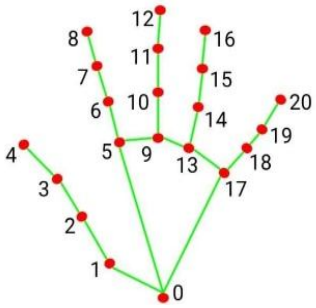

Figure 1. Media Pipe Hand Landmarks

Grenze ID: 01.GIJET.10.1.535

II. Review of Literature

It is possible to control the mouse using only our fingertips, without the need for any external device. This can be achieved by leveraging the capabilities of the Media Pipe framework to extract the hand region and the center of the palm from an image frame, and then obtaining the contours of the hand. After converting the color space from BGR to RGB, we can use an algorithm such as Autopy to track the movement of the fingers and count the number of fingers that are up. By interpreting different gestures, we can control the mouse and perform various operations.[1]

A model based on the single-shot detector architecture is employed for detecting the presence of a hand in a given frame captured by a webcam. Once a hand is detected successfully, a transformational algorithm is utilized in an AI virtual mouse system to map the coordinates of the fingertip from the webcam screen to the full computer window screen for controlling the mouse. To indicate the successful detection of the hand, a rectangular green box is displayed on the screen. The system also measures the distance between two fingers to enable various mouse operations.[2]

Hand gesture recognition and hand tracking on a laptop can be achieved through the use of MediaPipe framework and OpenCV. In addition, the Python module AutoPy can be trained to move the mouse pointer on the computer screen. This method can operate on real-world applications using only CPU resources instead of requiring a GPU.[3]

OpenCV is a popular computer vision library for Python that includes an AI-based virtual mouse system. This system relies on the edges captured by a camera on a computer to track hand movements and convert them into mouse movements on the computer screen. The system uses machine learning algorithms to accurately determine the coordinates of the hand tip and convert them into mouse movements on the full computer screen. This virtual mouse system uses only a small number of hand gestures and is not highly precise. [4]

This project uses a code for detecting colors to distinguish the background from green. The input consists of two rectangles, and the program calculates an average point from both shapes. This point serves as a virtual mouse pointer that moves in real-time as the user moves the mouse. If the color detection step is omitted, the program may encounter issues with accurately tracking the movement of the cursor.[5]

The system utilizes the frames captured by the computer's webcam or built-in camera in a laptop. A continuous loop is used to capture each frame until the program is terminated. These real-time video frames are then processed, converting the colors from BGR to HSV color space. The proposed system detects fingertip color pixels using color caps from the captured frames.[6]

In this system, a color cap is utilized to control the movement of the mouse cursor. Various mouse functions are triggered through hand gestures. The extraction of gestures and the color cap's location is carried out using MediaPipe and OpenCV. However, one of the drawbacks of this method is that the size of the cap is relatively large, leading to a more prolonged processing time and a delay in the cursor's movement. [7].

Here the author uses Matlab environment and hand gesture are recognised using Coloured finger tips YCbCr colour model is used for extracting the gesture and colour used to move the cursor [8].

The system in this article uses open cv to recognise hand gestures and background colour. Colour paper is necessary to differentiate between the hand motion and the background. After the gesture has been filtered, the desired mouse function is executed. The disadvantage of this approach is that you cannot do hand gestures without a colourful background [9].

The author of this paper used a media pipe and open cv to develop hand gesture recognition. A camera is used to record the image, and an Arduino is utilised to read and confirm the gesture. The author's accuracy is excellent, but the Arduino board adds an extra component to the system [10].

The author of this paper has created a system that uses hand movements to simulate a computer mouse. This method employs a cutting-edge skin colour segmentation methodology together with other processes like blob counting. MATLAB and webcam are used to capture the gestures. System is renowned for its simplicity since it has passed testing without experiencing any complexity problems [11].

According to this paper, 2D hand motions are used to build a mouse. To identify the hand motions, a camera-based colour recognition algorithm is utilised. To handle actions like hovering and clicking with the left or right mouse, a virtual HCI (Human Computer Interaction) system is created to recognise hand motions.[12].

III. Algorithms

MediaPipe is an open-source framework by Google, designed for applying machine learning pipelines to audio and video data. It consists of three main components: performance evaluation, sensor data access, and reusable

building blocks called calculators. These calculators form a data-flow diagram, creating a pipeline where data packets flow. MediaPipe uses a single-shot detector model for hand detection[14], initially detecting the palm and then locating 21 joint coordinates to create a hand landmark model. This approach is more convenient than real-time training and works well for small items like palms or fists.

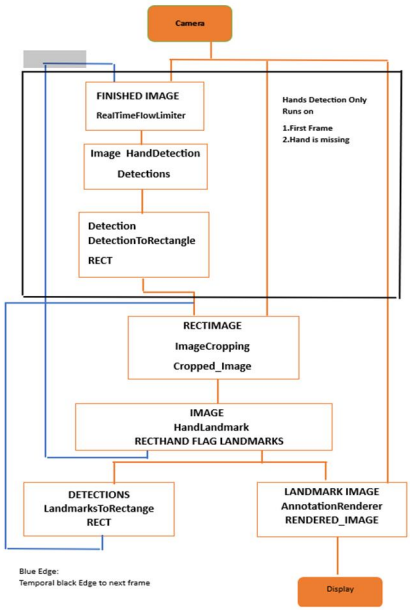

Figure 2. MediaPipe based Hand Recognition Graph

OpenCV is a popular computer vision library that provides powerful image-processing tools for object detection. By leveraging the OpenCV library in Python, developers can create real-time computer vision applications for a wide range of tasks, including image and video processing, as well as analysis tasks like face and object recognition. The library is a valuable resource for anyone looking to build sophisticated computer vision systems using Python.

IV. METHODOLOGY

Mouse Functions Using Computer Vision to Determine Hand Gestures and Hand Tip Detection.
Figure 3 illustrates how the AutoPy Python package is used to move the mouse pointer around the computer window if the index finger or both index fingers are raised with tip Id = 8 and middle finger are up with tip Id = 12.

Figure 3. Moving Cursor

Figure 4 illustrates how the AutoPy Python package is used to make the left click if the index finger is up with tip Id = 12.

Figure 4. Performing Left Click

Figure 5 illustrates how the AutoPy Python package is used to make the right click if the index finger is up with tip Id = 8.

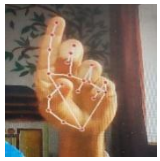

Figure 5. Performing Right Click

Figure 6 illustrates how the AutoPy Python package is used to make the neutral function activate i.e., no function is performed when all the fingers are detected as up.

Figure 6. Performing Right Click

In Python programming, the pynput package can be utilized to automate the execution of a double click on a computer. This can be achieved by setting certain conditions such as checking if both the middle finger (identified by tip Id = 12) and the index finger (identified by tip Id = 8) are raised and the distance between the two fingers is less than 63 pixels. A visual representation of this is shown in Figure.

Figure 7. Performing Double Click

If the index finger, ring finger, and little finger are up and the distance between the index finger and thumb of right hand is less than 63 pixels, and the hand is moved vertically, as shown in Figure 10, the system volume can be changed. If the same hand gesture is moved horizontally, the system brightness can be changed.

Figure 8. Hand Gesture for changing system volume and brightness

If the index finger, ring finger, and little finger are up and the distance between the index finger and thumb of left hand is less than 63 pixels, and the hand is moved vertically, as shown in Figure 11, the vertical scroll is performed. If the same hand gesture is moved horizontally, the horizontal scrolling is performed.

Figure 9. Hand Gesture for vertical and horizontal scroll

V. Results and Discussions

The AI virtual mouse system proposes the use of computer vision technology to improve human-computer interaction. Since there is a limited number of datasets available, it is challenging to compare the testing results

of the AI virtual mouse system. However, to evaluate the tracking of hand gestures and fingertip detection, different lighting conditions and distances from the webcam have been tested. The experimental test outcomes are summarized in the table. Everyone tested the AI virtual mouse system 30 times by 6 people, resulting in 650 motions with manual labelling. The test was conducted in various lighting conditions and distances from the screen. ten times in bright light, five times in dim light, five times up near to the webcam, and five times away from the webcam.

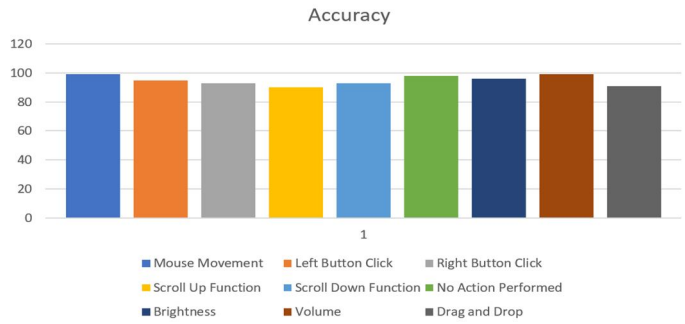

Figure 10. Graph of accuracy

VI. Conclusion

The main objective of the AI virtual mouse system is to eliminate the requirement for a physical mouse by allowing users to control the cursor through hand gestures. The proposed system can be deployed by utilizing a webcam or a built-in camera that detects hand movements and fingertips and processes the captured frames to perform various mouse functions. The proposed AI virtual mouse model has the potential for practical applications due to its high level of accuracy. Additionally, it can be particularly useful in preventing the spread of COVID-19 since the virtual mouse can be operated using hand gestures, rather than relying on a physical mouse, which can be a source of contamination. Based on my analysis, it can be inferred that the proposed AI virtual mouse system has shown significant improvements compared to the current models in terms of accuracy and has also addressed many of the limitations of the existing systems. Next, we'll focus on enhancing the fingertip detection algorithm to give more accurate results in order to get over the limitations.

References

[1] Sankha Sarkar, Indrani Naskar, Sourav Sahoo, Sayan Ghosh, Sumanta Chatterjee, A Vision Base Application For Virtual Mouse Interface Using Hand Gesture (2021).
[2] Shriram, B.Nagaraj, J.Jaya, S.Shankar, P. Ajay, Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread(2021).
[3] Abhishek R. Shukla, AI Virtual Mouse Using Hand Gesture Recognition (2022).
[4] Joy Guha, Shreya Kumari, Prof. Shiv Kumar Verma, AI Virtual Mouse Using Hand Gesture Recognition (2022).
[5] Kollipara Sai Varun, I. Puneeth, Dr. T. Prem Jacob, Virtual Mouse Implementation using OpenCV(2019).
[6] Kabid Hassan Shibly, Samrat Kumar Dey, Md. Aminul Islam, Shahriar Iftekhar Showrav, Design and Development of Hand Gesture Based Virtual Mouse (2019).
[7] Vantukala VishnuTeja Reddy, Thumma Dhyanchand, Galla Vamsi Krishna, Satish Maheshwaram.Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition (2020)
[8] Rachit Puri. Gesture Recognition Based Mouse Events (2020)
[9] Horatiu-Stefan Grif, Trian Turc. Human hand gesture-based system for mouse cursor control (2018).
[10] Telagarapu Prabhakar, Gurram Sunitha, Gudavalli Madhavi. AI And Hand Gesture Recognition Based Virtual Mouse (2022)
[11] Rokhsana Titlee, Ashfaq Ur Rahman, Hasan U Zaman, Hafiz Abdur Rahman, A Novel Design of an Intangible Hand Gesture controlled computer mouse using vision-based image processing (2017).
[12] Sandeep Thakur, Rajesh Mehra, Buddhi Prakash, Vision based Computer Mouse Control using Hand Gestures (2015).
[13] G. S. Mate, A. K. Kureshi, and B. K. Singh, An efficient CNN for hand x-ray classification of rheumatoid arthritis (2021).
[14] S. Adhav, R. Rekhawar, V. Tapale, S. Habre and A. Savagonmath, Survey on Healing Herbs Detection using Machine Learning (2022).