

Virtual Mouse Using Artificial Intelligence

P.SaiDhanush, P.Sharanam, P.Rohit, S.SaiGanesh

Zeenath Jaha Begum

Hyderabad Institute of Technology and Management

Abstract: The mouse is a small, movable device that allows you to control a variety of things on a computer. Currently, a wireless mouse or a Bluetooth mouse uses devices and is not completely free of devices because it relies on a battery for power and a dongle to connect to the PC. This limitation can be overcome in the proposed AI virtual mouse system by using a webcam or a built-in camera to capture hand gestures and detect hand tips using computer vision.

Keywords: Hand tracking, fingertip recognition, cursor movement, computer vision, and human-computer interaction.

INTRODUCTION:

AI virtual mouse system that uses computer vision to perform mouse functions in the computer by using hand gestures and hand tip detection. The proposed system's main goal is to perform computer mouse cursor and scroll functions using a web camera or a built-in camera in the computer rather than a traditional mouse device. As an HCI [1] with the computer, computer vision is used to detect hand gestures and tips. Using an AI virtual mouse system, we can track the fingertip of a hand gesture using a built-in

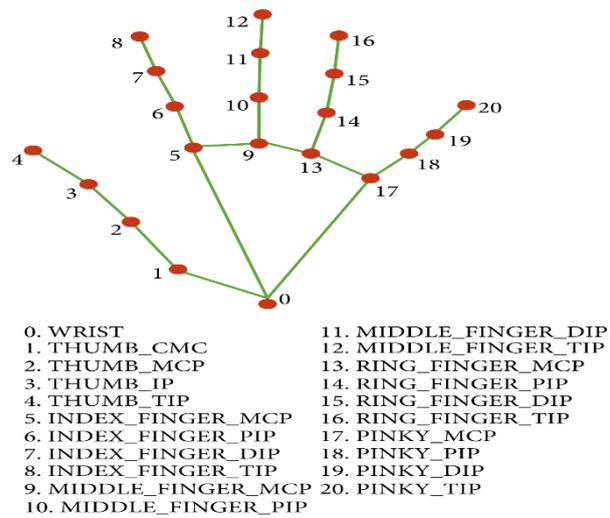
camera or web camera and perform mouse cursor operations, scrolling, and cursor movement. Python is a programming language is used in the development of the AI virtual mouse system, as is OpenCV, which is a computer vision library. The model in the proposed AI virtual mouse system makes use of the Media Pipe package for tracking the hands and the tip of the hands, as well as the Pyn put, Autopy, and PyAutoGUI packages for moving around the computer's window screen to perform functions such as left click, right click, and scrolling functions.

Some devices, such as the mouse, the dongle to connect to the PC, and a battery to power the mouse to operate are used when using a wireless or Bluetooth mouse, but in this paper, the user uses his/her built-in camera or a webcam and uses hand gestures to control the computer mouse operations. In the proposed system, the web camera captures and then processes the captured frames, recognizing various hand gestures and hand tip gestures before performing the specific mouse function.

The Python programming language is used in the development of the AI virtual mouse system, as is OpenCV, a computer vision library. The model in the proposed AI virtual mouse system makes use of the Media Pipe package for tracking the hands and the tip of the hands, as well as the Pynput, Autopy, and PyAutoGUI packages for moving around the computer's window screen to perform functions such as left click, right click, and scrolling functions. The proposed model's results demonstrated very high accuracy, and the proposed model can work very well in real-world applications using a CPU rather than a GPU.

LITERATURE SURVEY:

This paper is an approach for Human Computer Interaction (HCI) to communicate between humans and computers. We attempted to control the mouse cursor movement and click events using hand gestures to perform some basic mouse operations based on hand moments. Color-detection cameras were used to capture hand gestures. This article primarily focuses on using a web camera to communicate with a device in a cost-effective manner.



3.WORKING:

1)Real-time video capture with a webcam:

In order for the system to function, we will need a sensor to detect the user's hand movements. The webcam on the computer is used as a sensor. The webcam captures real-time video at a fixed frame rate and resolution determined by the hardware of the camera. The system's frame rate and resolution can be changed if necessary.

2) Converting the captured video to HSV format:

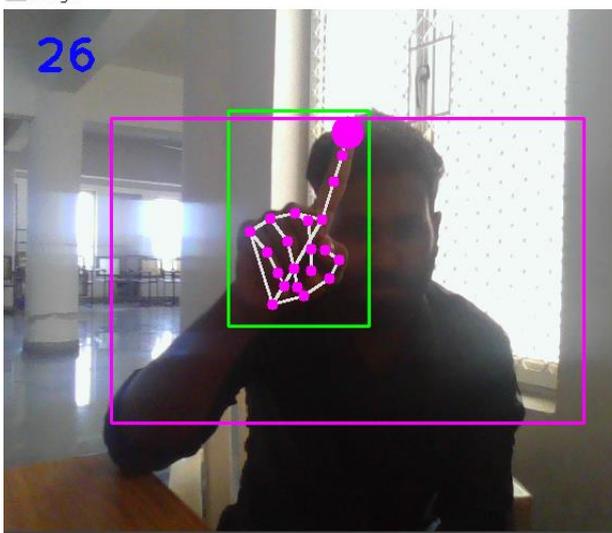
The video has also been converted to HSV (hue, saturation, meaning, also known as HSB), an alternative representation of the RGB colour model developed by computer graphics researchers to better reflect human vision's perception of colored characteristics.

3) Each image frame is processed independently:

After the video is captured, it goes through a brief pre-processing stage before being processed one frame at a time.

4) **Mouse pointer tracking:** The mouse driver is accessed after determining the coordinates, and the coordinates are sent to the cursor. Using these coordinates, the cursor moves to the required location. As a result, as the user moves his hands across the camera's field of view, the mouse moves proportionally across the screen.

5) **Simulating mouse movements;** In simulation mode, the user must make hand gestures to create control actions. The use of colour pointers shortens the computation time.



IMPLEMENTATION

1. **Camera Settings:** The webcam of the connected laptop or desktop manages the runtime operations. To record a video, we must first create a Video Capture object. It accepts either the device index or the name of a video file as an argument. The device index is simply a number that indicates

which camera is being used. We pass it as '0' because we only use one camera. We can add more cameras to the system and pass them as 1, 2, and so on. After that, you can capture frames one by one. But don't forget to release the capture at the end. We could also apply colour detection techniques to any image by making minor changes to the code.

2. **Capturing frames:** The infinite loop is used to ensure that the web camera captures frames in every instance and remains open throughout the programme. Frame by frame, we record the live feed stream. Then we convert each captured frame from RGB (default) to HSV colour space. OpenCV includes more than 150 color-space conversion methods. However, we will only look at two popular ones: BGR to Gray and BGR to HSV.

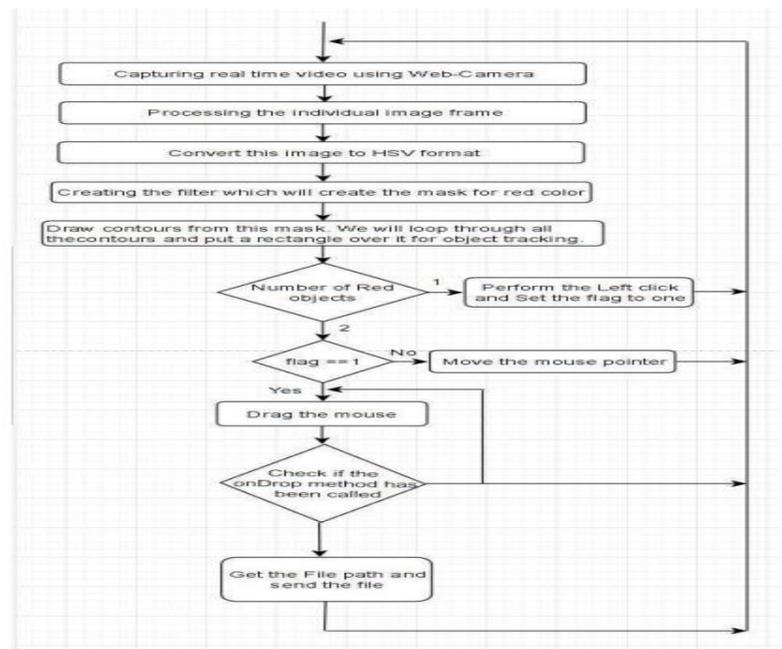


Fig -1: Flowchart

3. Making technique: A mask is used to create a specific region of an image by following certain rules. In this example, we're making a mask out of a red object. Following that, we perform a bitwise AND operation on the Input and Threshold images, highlighting only the red coloured objects. The outcome of the AND operation is saved in res. We then use the imshow() function to display the frame, res, and mask on three separate windows.

4. Display the frame: The imshow() function of High Gui is required to call the wait Key on a regular basis. The event loop of the imshow() function is processed by calling wait Key. wait Key() waits for a key event for a "delay" (here, 5 milliseconds). High Gui handles Windows events such as redraw, resizing, and input events. So, even with a 1ms delay, we call the wait Key function.

5. Mouse Movement: We must first calculate the center of both detected red objects, which we can easily do by averaging the maximum and minimum points of the bounding boxes. We now have two co-ordinates from the centers of the two objects, which we will average to get the red point shown in the image. The detected coordinate from the camera resolution is being converted to the actual screen resolution. The location was then set as the

mouse position. However, moving the mouse pointer will take some time. As a result, we must wait until the mouse pointer reaches that point. So we started a loop and are doing nothing there except waiting to see if the current mouse position is the same as the assigned mouse position, that's the open gesture.

6. Clicking: The following step is to use the close gesture. The operation is carried out by clicking and dragging the object. It is similar to the open gesture, but because we only have one object here, we only need to calculate its centre. And it will be placed on the spot where we will place our mouse pointer. We will perform a mouse press operation instead of a mouse release operation.

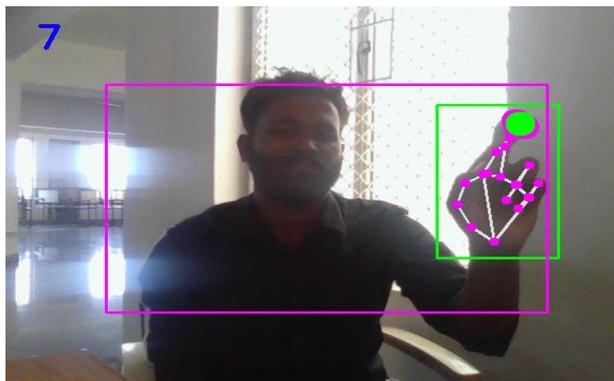
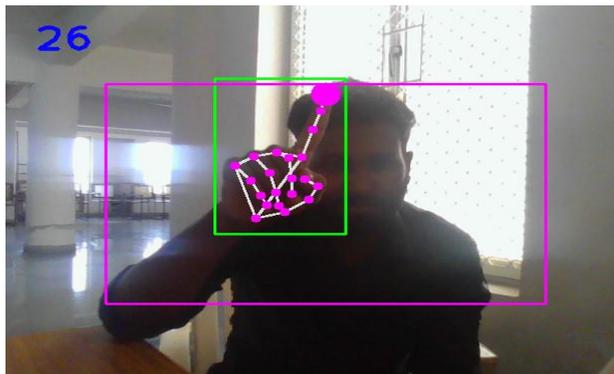
7. Drag: To implement dragging, we introduce the variable 'pinch flag'. It will be set to 1 if it was clicked earlier. So, whenever we find the open gesture, we check to see if the pinch flag is set to 1. If set to one, the drag operation is performed; otherwise, the mouse move operation is performed.

RESULTS AND TESTCASES

1. Only use "Forefinger" for Hand Tracking, which is only supposed to be raised for device monitoring.

2. To access "Clicking and Dragging Features," use both the "Middle Finger" and

the "Fore Finger" at the same time. Once after clicking on track button, the camera of your device will be automatically accessed by this application and a small window will pop up in which "Your hand will start getting monitored."



CONCLUSION

On the mouse cursor, a Web camera is running. This will also result in new levels of human-computer interaction (HCI) that do not necessitate physical contact with the device. This machine is capable of performing all mouse tasks involving colour recognition. This device has the potential to be useful for interacting with contactless input modes. It's useful for people who don't use a touchpad. The proposed device's architecture would be drastically altered.

Interactions between people and computers. Everyone is able to use the Webcam, microphone, and mouse. It would completely eliminate the need for a mouse. It can also be used in games or other standalone applications. Free movement, left-click, right-click, drag/select, scroll-up, and scroll-down are all operations that can be accomplished solely through gestures in this application Multi-Functional system, only gestures are used.

REFERENCE

- [1] ISSN: 2231-2803 International Journal of Computer Trends and Technology (IJCTT) - volume 9 number 1 - March 2014 www.internationaljournalsrg.org Page 15: Mouse Control with a Web Camera Using Color Detection.
- [2] K N. Shah, K R. Rathod, and S. J. Agravat, "A Survey on Human Computer Interaction Mechanisms Using Finger Tracking," 7(3), 2014, 174-177.
- [3] Tutorials point (n.d.). Agile Model for SDLC. [online] The tutorial is available at http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm.
- [4] Tkinter Python GUI Programming <https://realpython.com/python-gui-tkinter/>
- [5] NumPy in Python, <http://numpy.org/>