

Virtual Mouse Control Using Hand Gesture Recognition

G N Srinivas¹, S Sanjay Pratap², V S Subrahmanyam³, K G Nagapriya⁴, A Venkata Srinivasa Rao⁵

⁵ Department of ECE, Sasi Institute of Technology & Engineering, Tadepalligudem, W.G. Dist, India.

^{1,2,3,4}UG Students, Department of ECE, Sasi Institute of Technology & Engineering, Tadepalligudem, W.G. Dist, India

Abstract - The mouse is an excellent device for human-computer interaction. We currently use three types of mice: wired, wireless, and Bluetooth. We need power to connect a dongle to a PC in all of these scenarios. This work employs cutting-edge machine learning and computer vision algorithms to recognize hand gestures, which work flawlessly without the use of any hardware. It is compatible with CNN models implemented by mediapipe. In this paper, we work on S. Shriram's algorithm that [11] propose a method for controlling the cursor's location using only one's hands and no mouse. Some actions, like clicking and dragging items, will necessitate a variety of hand movements. The proposed system will only require the use of a single computer. A camera is used as an input device. The following programs will be used: The proposed system necessitates the use of Python and OpenCV. The output of the camera will be displayed on the system's display so that the end-user can fine-tune it.

Key Words: Camera, Machine Learning, CNN Model, Mediapipe, Virtual Mouse, etc

1. INTRODUCTION

Hand gestures are universally recognized as the most expressive and effective form of human communication. Hand signals, thumbs up, and thumbs down have always existed. Gestures are regarded as the most natural way for people to communicate with one another. It has a lot of personality. It was written in such a way that it could be understood by the deaf and dumb. So why not put it to use on our machines? In this work, we present actual hand gestures. The initial setup includes a low-cost USB web camera for system input.

This paper proposes a real-time hand gesture system. The experimental setup of the system makes use of a low-cost web camera with high-definition recording capability that is installed in a fixed position. A camera mounted on a computer monitor is used to photograph a laptop. This project proposes an effective hand gesture segmentation technique based on pre-processing, background subtraction, and edge detection approaches.

The Python programming language and OpenCV, a computer vision library, were used to create the AI virtual mouse system. The MediaPipe package is used to track hands and fingers, as well as the Pynput, Autopy, and PyAutoGUI packages for navigating the computer's window screen and

performing operations like left click, right click, and scrolling.

2. LITERATURE SURVEY

Chen-Chiung Hsieh et al. [1] proposed "A Real Time Hand Gesture Recognition System Using Motion History Image" to control the mouse cursor. The proposed method employs an adaptive skin colour detection model to reduce misclassifications. To develop these methodologies, they used a C++ software platform with the image processing library open cv installed. Kamran Niyazi et al [2] proposed "Mouse Simulation Using Two Colored Tapes," which used the Background Subtraction method, Skin Detection method, and HSV Color Model to control the cursor and perform clicking operations. The distance between the tape colours was used to guide the clicking operations. This model was created using Java software. Abhik Banerjee et al. [3] proposed a "Mouse Control Using a Web Camera Based on Color Detection" to control cursor movements and click events by detecting camera colour. Each colour represents a different cursor control, and clicking actions are performed by simultaneously detecting the colours. This method was created with the help of MATLAB software and the MATLAB image processing tool box. "Vision-based Computer Mouse Control Using Hand Gestures" [4] was proposed by Sandeep Thakur et al. To improve the efficiency and reliability of the interaction, this method employs a vision-based system to control various mouse activities such as left and right clicking with hand gestures. To improve the system's efficiency and performance, different colour caps are used on fingers to recognise hand gestures. The MATLAB environment was used to implement this method. To control the mouse cursor, Horatiu-stefan Grif et al [5] proposed "Mouse Cursor Control Based on Hand Gesture". They used an external camera attached to a hand pad and colour strips attached to the fingers in the proposed method. To implement this methodology, they used C programming software along with an image processing library called OpenCV. Pooja Kumari et al. [6] proposed "Cursor Control Using Hand Gestures" for controlling a mouse with camera-captured hand gestures. The camera acts as a sensor in this method, capturing and recognising colour tips attached to the hand. Because it requires the user to have colour tips on his hand in order to control the mouse, this method is also known as the marker-based approach method. To implement this methodology, they used the MATLAB environment, the MATLAB Image Processing Tool box, and

the OpenCV library. Using an Extreme Learning Machine Method, Danling Lu et al. [7] proposed a "Gesture Recognition Using Data Glove" method for controlling the mouse cursor and performing clicking operations. In this method, they collected data for gesture recognition using a novel data glove. This glove is inexpensive. Advanced machine learning was used to create this method. Alisha Pradhana et al. [8] proposed the "Design of Intangible Interface for Mouseless Computer Handling Using Hand Gestures" to control the mouse cursor and click operations. This method, which employs a convex hull algorithm, was implemented using Microsoft virtual studio, a Microsoft integrated development environment. To detect the user's hand and reduce noise, a red glove is used. Aashni Hariaa et al. [9] proposed "Hand Gesture Recognition for Human Computer Interaction," which uses a background extraction and contours detection system to improve user-computer interaction. Human interaction can be accomplished through a variety of sensor modes such as gesture, speech, facial and body expressions. In their paper, Horatiu-Stefan et al. proposed a "Human Hand Gesture Based System for Mouse Cursor Control". This model was created using a blue-colored hand pad, a webcam, and a computer. Specific operations were carried out based on hand pad postures. This model was created using Visual C++ 2008 and the OpenCV library. For controlling the mouse cursor and performing clicking operations, Abhilash et al. [10] proposed the "Virtual Mouse Using Hand Gesture" technique. They made a mask out of a red object in this case. This method was implemented using Python software, which included a variety of modules and functions.

3. METHODOLOGY

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.

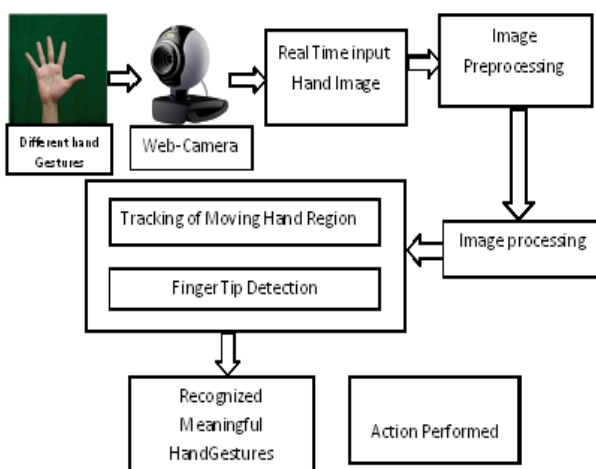


Fig-3.1 Block diagram for proposed system

Fig-3.1 depicts a functional block diagram of the proposed system, which demonstrates how the system operates. We must hold our hand up to the webcam. The webcam starts the video and captures the frames. The input image was subjected to pre-processing. The function of the image pre-primary processor is to standardize the image. The process of scaling and pre-processing an image to have similar heights and widths is known as standardization. To improve the quality of the standard image, it is now processed using an image processing technique. The camera moves its hand after image processing, and the finger tips are detected using MediaPipe and openCV. After recognizing the hand and finger tips, it begins to draw.

There are hand landmarks and a box around the hand on the screen. On the window pc, draw a rectangular box to hold the mouse. It will determine which of your fingers is up and which is down. Based on the finger detections, the mouse action is performed, and the program returns to the frames to perform the next operation. This is how the entire system operates.

3.1. ALGORITHM

Step 1: Start

Step 2: Start the webcam video capture and initialize the system.

Step 3: Frame capture with a webcam.

Step 4: Using Media Pipe and OpenCV, detect hands and hand tips and draw hand landmarks and a box around the hand.

Step 5: Draw a rectangle around the computer window area where we'll be using the mouse.

Step 6: Determine which finger is raised.

Step 6.1: The gesture is neutral if all five fingers are up, and the next step is taken.

Step 6.2: The cursor moves to step 2 if both the middle and index fingers are raised.

Step 6.3: A double click is performed when both index and middle fingers are joined side by side, and step 2 is performed.

Step 6.4: If both index and middle fingers are down, perform a left click and proceed to step 2.

Step 6.5: If the middle finger is down and the index finger is up, the right click is performed and the process proceeds to step 2.

Step 6.6: Volume up and down are accomplished by joining the thumb and index fingers and moving them up and down.

Step 7: To exit, press the EXIT key.

The preceding algorithm shows how to simulate virtual mouse control with hand gestures step by step. This makes it easier to swap a physical mouse for a virtual mouse. This will aid in the conversion of input hand gestures into mouse clicking functions.

The preceding procedure, i.e. the algorithm, is depicted diagrammatically in flow chart Fig-3.2.

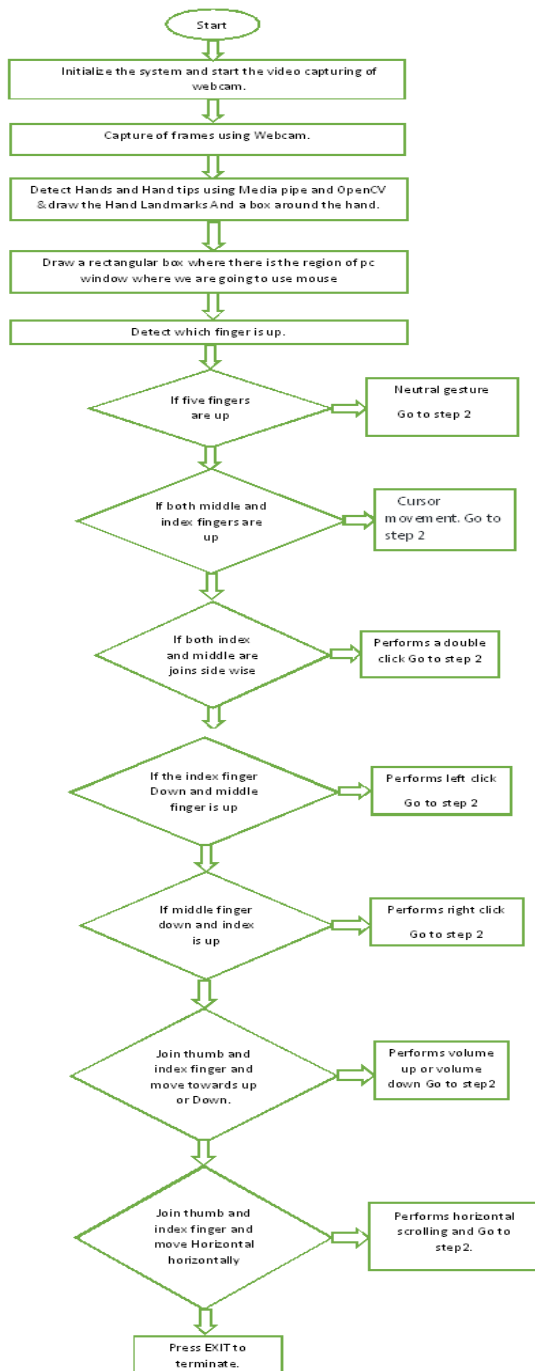


Fig-3.2 Flow chart for proposed system

- The AI Virtual Mouse System uses a camera. The proposed AI virtual mouse system is based on frames captured by a laptop or PC's webcam. The video capture object is created using the Python computer vision library OpenCV, as shown in Fig-3.1, and the web camera begins capturing video. The web camera captures images, which are then sent to the AI virtual system.

- The video is currently being recorded and processed. The AI virtual mouse system uses a webcam to capture each frame until the program is terminated.

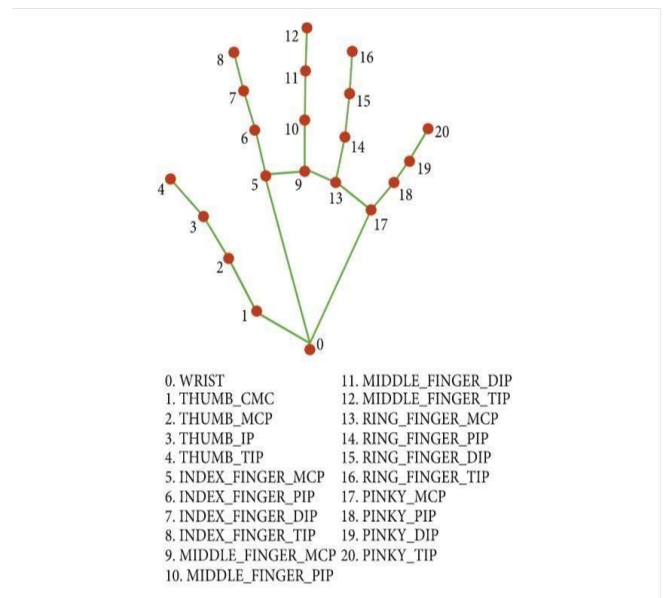


Fig-3.3 Model Graph of MediaPipe

- Detecting which finger is up and carrying out mouse functions. At this point, we use the tip Id of the respective finger discovered using the MediaPipe and the respective coordinates of the up fingers, as shown in Fig-3.3, to determine which finger is up, and then we perform the appropriate mouse function.

- Mouse functions based on hand gestures and hand tip detection using computer vision.

4. RESULTS & DISCUSSIONS

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.

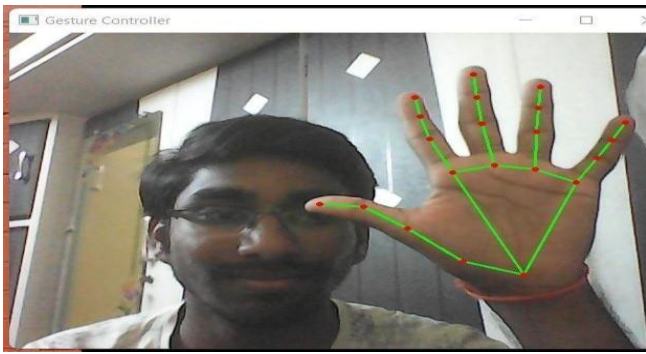


Fig.4.1 Neutral Gesture

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 Fig-4.2

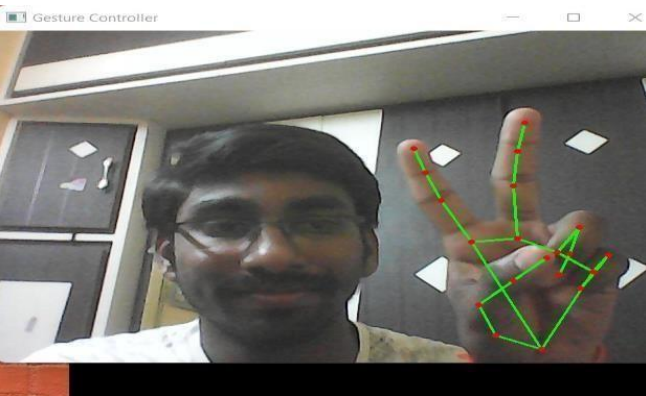


Fig-4.2 Cursor Control

- To perform a left-button click with the mouse. The computer is made to perform the left mouse button click using the pynput 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-4.3.



Fig-4.3 Left Click

- 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 Fig-4.4

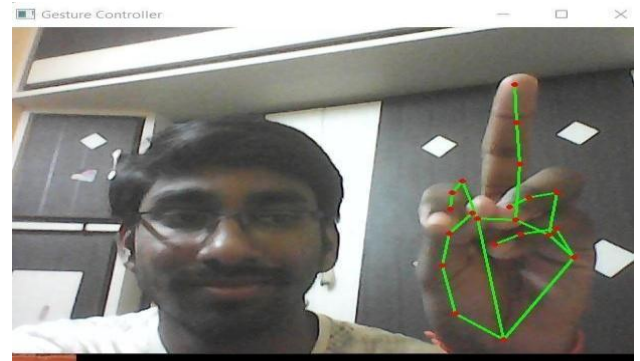


Fig-4.4 Right Click

- If both the index and middle fingers are closed, the mouse will perform a drag operation. As shown in Fig-4.5, the computer is programmed to perform the drag operation button click using the pynput Python package.



Fig-4.5 Drag Operation

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. CONCLUSIONS

The primary goal of the AI virtual mouse system is to control mouse cursor functions with hand gestures rather than a physical mouse. The proposed system can be realized by using a webcam or an in-built camera that detects hand gestures and hand tips and processes these frames to perform the specific mouse functions.

Based on the model's results, conclude that the proposed AI virtual mouse system performed well and has higher accuracy than existing models, and that the model

overcomes most of the limitations of existing systems. The AI virtual mouse can be used for real-world applications because the proposed model is more accurate and can be used virtually using hand gestures rather than the traditional physical mouse.

REFERENCES

- [1] Chen-Chiung Hsieh ,Dung-Hua Liou and David Lee “A real time hand gesture recognition system using motion history image” Proc.IEEE, 2010, pp. V2-394-V2-398 .
- [2] Kamran Niyazi, Vikram Kumar , Swapnil Mahe and Swapnil Vyawahare “Mouse Simulation Using Two Coloured Tapes”,IJIST 2012, Vol.2, No.2, DOI : 10.5121.
- [3] Abhik Banerjee , Abhirup Ghosh , Koustuvmoni Bharadwaj, Hemanta Saikia “MouseControl using a Web Camera based on Colour Detection”IJCTT ,March 2014, volume 9 number 1, ISSN: 2231-2803
- [4] S. Thakur, R. Mehra and B. Prakash, "Vision based computer mouse control using hand gestures," International Conference on Soft Computing Techniques and Implementations (ICSCTI), 2015, pp. 85-89.
- [5] Horatiu-stefan Grif, Train turc “Human hand gesture based system for mouse cursor control” INTER-ENG, October 2017.
- [6] Pooja Kumari ,Ghaziabad Saurabh Singh, Ghaziabad Vinay and Kr. Pasi “Cursor Control using Hand Gestures” International Journal of Computer Applications (0975 – 8887),2016.
- [7] Danling Lu, Yuanlong Yu, and Huaping Liu “Gesture Recognition Using Data Glove: An Extreme Learning Machine Method”Proc.IEEE,December 2016, pp. 1349-1354.
- [8] Alisha Pradhan , B.B.V.L. Deepak “Design of Intangible Interface for Mouseless Computer Handling using Hand Gestures” ICCCV(International Conference on Communication, Computing and Virtualization), 2016, oi: 10.1016/j.procs.2016.03.037.
- [9] Aashni Hariaa , Archanasri Subramaniana , Nivedhitha Asokkumara , Shristi Poddara and Jyothi S Nayak “Hand Gesture Recognition for Human Computer Interaction” Proc.ICACC(International Conference on Advances in Computing & Communications), 2017 August,pp367-374.
- [10] Abhilash S , Lisho Thomas, Naveen Wilson, and Chaithanya “VIRTUAL MOUSE USING HAND GESTURE” Proc. International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056 p-ISSN: 2395-0072, Apr-2018, Volume: 05 Issue: 04 .
- [11] S.Shriram, B.Nagaraj, J Jaya, S Shankar and P Ajay “Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to avoid Covid-19 Spread” Journal of Health Care Engineering, Volume:2021, Article ID:8133076.