GESTURE-BASED HUMAN-COMPUTER INTERACTION

N. Meghana[1], K. Sri Lakshmi[2], M. Naga Lakshmi Tejasree[3], K.Srujana[4], N.Ashok[5]
[1,2,3,4]Students, [5]Associate Professor
[1]Department of Information Technology
[1]Vasireddy Venkatadri Institute of Technology, Guntur, India.

Article DOI: https://doi.org/10.36713/epra14757
DOI No: 10.36713/epra14757

ABSTRACT

This project explores touchless interaction with computers using gesture recognition technology. By combining computer vision and gesture recognition, users can control their computers using hand gestures. The system uses a webcam for input, processes images with OpenCV, tracks hand movements with MediaPipe, and controls mouse and keyboard actions with PyAutoGUI. Users can perform various gestures like clicks, scrolls, and custom "Open Paint" gesture to control their computer. The project integrates with a paint application, allowing users to create digital art using their hand gestures, enhancing the overall user experience. Users can seamlessly control their computers by executing predefined gestures in the air, eliminating the limitations of conventional input device like mice.

KEYWORDS - *Gesture-based interaction, Human-Computer Interaction, MediaPipe, OpenCV, Paint Application, PyAutoGUI, Virtual mouse.*

I. INTRODUCTION

In 2018, a method for capturing shape and position information using color detection was introduced[1]. This development primarily focused on utilizing distinct colors to perform specific functions. Fast forward to 2022, a groundbreaking innovation emerged: the virtual mouse [2]. This virtual mouse eliminated the need for color-coded hand accessories and expanded its utility beyond cursor control. However, its functionality was primarily limited to cursor operations.

Recognizing this limitation, we have embarked on a journey to enhance the versatility and functionality of the virtual mouse system by integrating it with a paint application. This integration promises to elevate the project's utility to new heights. Our approach in implementing the virtual mouse involves the detection of a user's hand movements within an image, which are then translated into mouse pointer movements [3]. We employ a hand gesture recognition system and webcam input to identify motion in images, effectively replacing the conventional mouse system that has been in use for decades. The system is not confined to merely replicating cursor movements; it extends to controlling standard mouse functions, such as left-clicking, through the use of a real-time digital camera [4].

The integration of a paint application into our system represents a significant step forward in enhancing user interaction. This process involves capturing video frames, detecting a specific colored marker, tracking its movements, and using it to draw on a digital canvas. Furthermore, the paint application seamlessly interfaces with the virtual mouse. When the virtual mouse detects a custom gesture signifying the desire to access the paint application, it triggers the paint program to open, further expanding the scope and possibilities of our innovative project.

II. OBJECTIVES

- Improve user-computer interaction by recognizing a wide range of hand gestures, offering a more immersive experience
- Empower users to express their creativity digitally by integrating a paint application with intuitive hand gesture controls.
- Enable seamless and intuitive mouse control through hand gestures detected by a webcam, providing a user-friendly and efficient replacement for traditional mouse systems.

III. METHODOLOGY

The project starts by initializing the video interface through webcam. The system uses OpenCV which is open source computer vision library to process the video. We are using cv2.VideoCapture() function to get a video capture object for the camera and will read the data from that object using GestureController.cap.read() for further processing.

GestureController.cap = cv2.VideoCapture(0)

Once the video has been processed, the system uses MediaPipe to detect and identify landmarks on the user's hand. Using the process() function , the system can detect the hand landmarks.

results = hands.process(image)

These landmarks represent key points on the hand. These landmarks are processed to recognize the gesture. Once the gestures are recognized, the system translates these gestures into cursor operations. Those cursor operations are performed with the help of PyAutoGui.

- pyautogui.moveTo() for cursor movement.
- pyautogui.click() for left click.

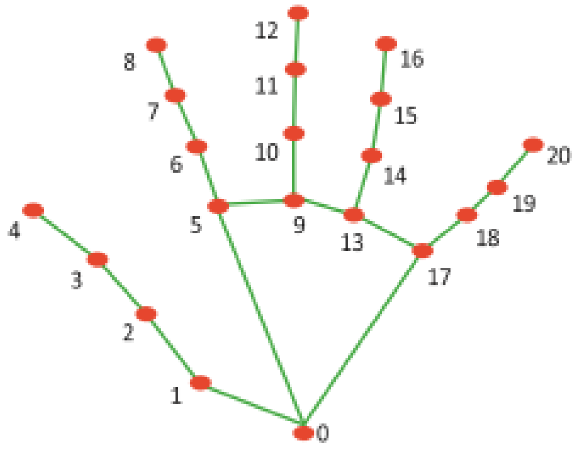

Fig.1: Landmark or Co-Ordinates in the Hand[6]

0.WRIST

1.THUMB_CMC

2.THUMB_MCP

3.THUMB_IP

4.THUMB_TIP

5.INDEX_FINGER_MCP

6.INDEX_FINGER_PIP

7.INDEX_FINGER_DIP

8.INDEX_FINGER_TIP

9. MIDDLE FINGER MCP

10.MIDDLE FINGER

11.MIDDLE_FINGER_DIP

12.MIDDLE_FINGER_TIP

13.RING_FINGER_MCP

14.RING_FINGER_PIP

15.RING_FINGER_DIP

16.RING_FINGER_TIP

17.PINKY_MCP

18.PINKY PIP

19.PINKY DIP

20.PINKY TIP

Those operations include mouse movement , right click , left click, double click ,drag and drop and scrolling. When the system recognizes pinch gesture, then it triggers to art mode where users can paint using their hands as brushes. Users can seamlessly make transition between virtual mouse and paint application.

The paint application has functions like clearing the canvas and provides color selections for drawing. Users can interact with the paint application by moving their hands to paint in different colors. The code maintains several arrays for storing points of various colors effectively and provides a user interface so that users can choose colors.

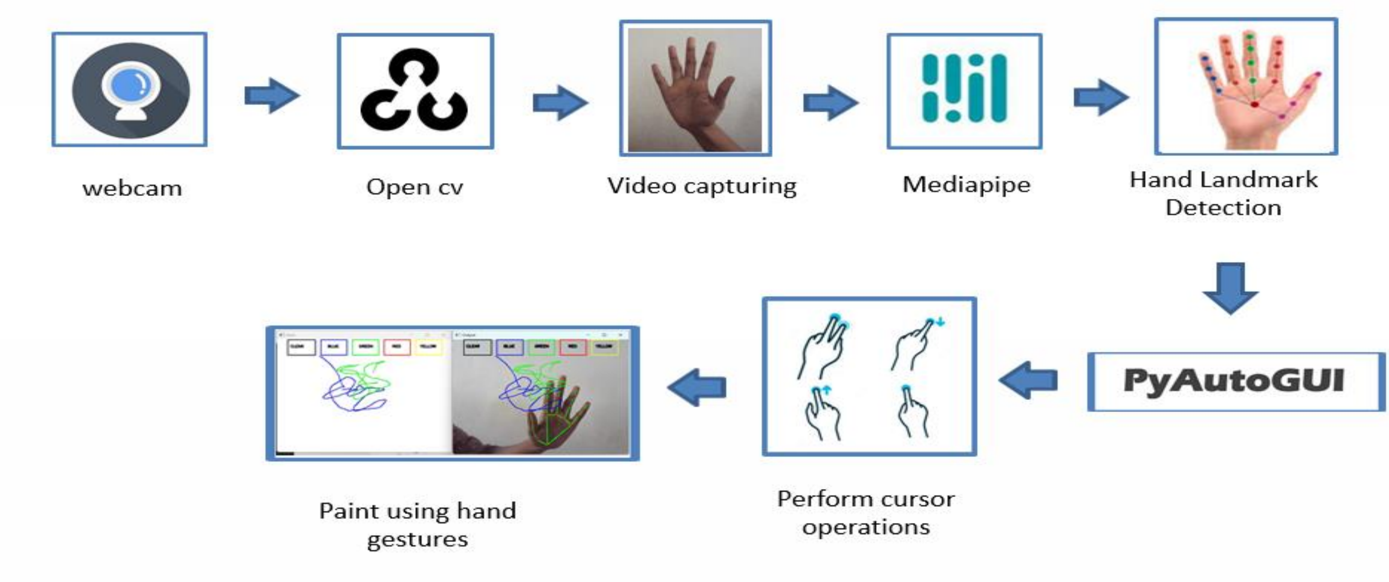

Fig.2: System Architecture

To develop the paint application, firstly the code create a canvas with different color options like blue, green, red and yellow for drawing. After that, the code will start the video capture through webcam and the video frames are captured. Those video frames are processed to detect the hand landmarks using the MediaPipe library. And then, the code can recognize the hand gesture for the detected hand landmarks. Then, the user can draw the different colors of lines on canvas based on their hand gestures. There will be specific gestures for selecting the colors and clearing the canvas. The user can exit the paint application by pressing the ‘q’ key.

IV. RESULTS

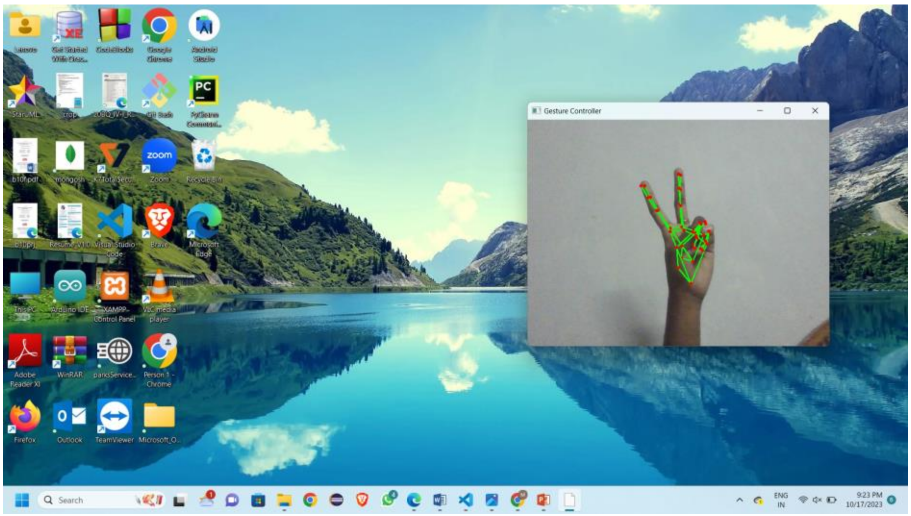

Fig.3: Mouse Movement

Figure 3 The system should precisely capture and respond to the hand gesture simulating mouse movement. Users should experience a smooth and accurate translation of hand movements into on-screen cursor actions, ensuring a fluid and intuitive interaction with the digitalenvironment.

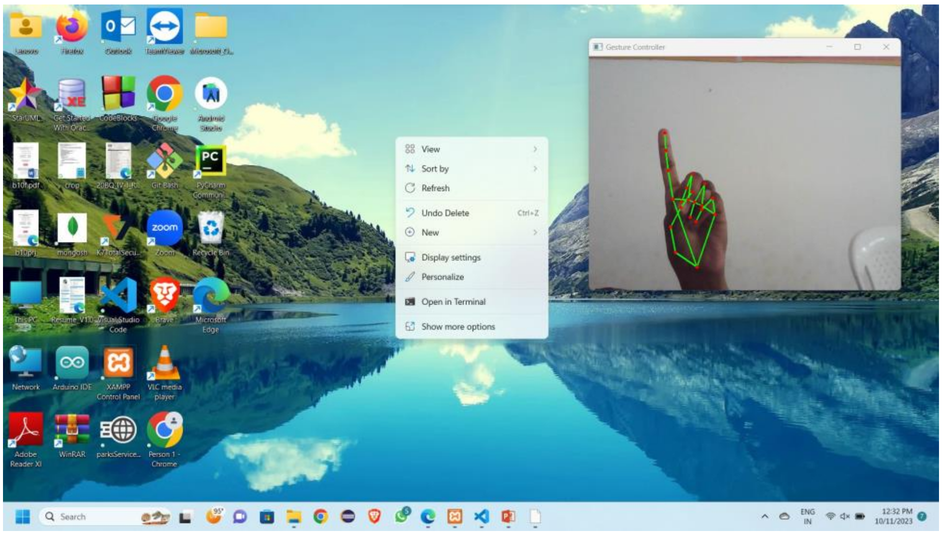

Fig.4: Right-Click

Figure 4 The system must reliably recognize the right-click hand gesture, enabling users to trigger context menus or other right-click functions within applications. The system should respond promptly and accurately to the gesture, providing an efficient alternative to traditional mouse interaction.

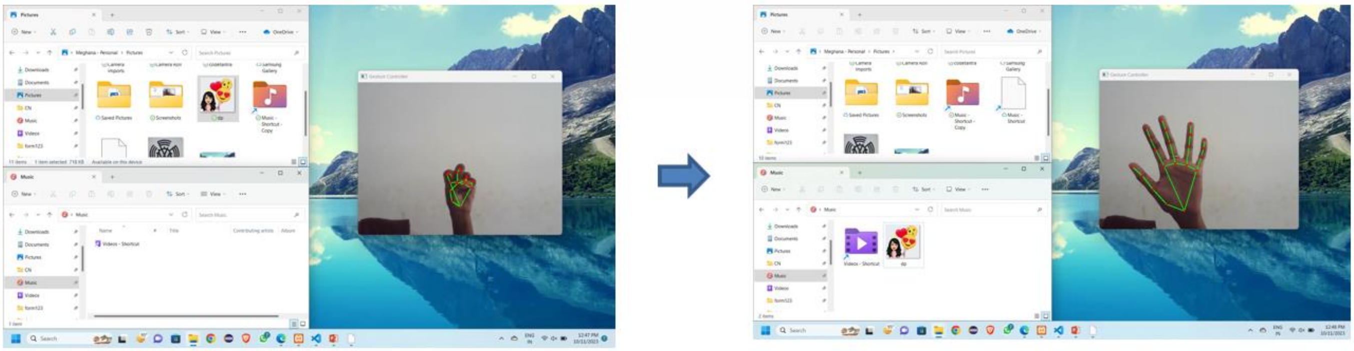

Fig.5: Drag And Drop

Figure 5 The system should accurately interpret the hand gesture to initiate and complete the drag operation. The object should follow the hand's movement during the drag, providing visual feedback. The release gesture should trigger the drop action, placing the object in the new location.

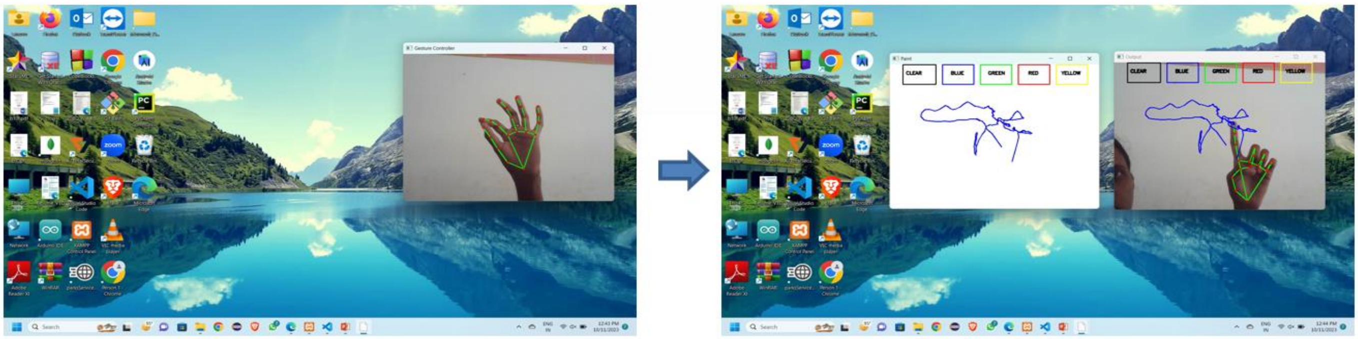

Fig.6: Paint

Figure 6 The system seamlessly interacts with the paint application and enables artistic expression through hand gestures, providing a user-friendly and engaging experience for creative tasks.

V. SUGGESTIONS

- Gesture Vocabulary: Enhance the system by adding more gestures to increase its functionality. Recognize a broader range of gestures for increased user convenience.
- Multimodal Integration: Explore the integration of voice commands to create a multimodal HCI system. This allows users to combine hand gestures with voice instructions for a richer interaction experience.
- User Customization: Develop features that enable users to customize gesture definitions, making the system more personalized.

VI. CONCLUSION

The Gesture-based Human-Computer Interaction (HCI) system introduced in this project has revolutionized the way people interact with computers. This system allows users to control digital interfaces using hand gestures, which is more natural compared to traditional input devices like a mouse. It overcomes the limitations of regular mouse interfaces and offers a smooth and effective solution for various tasks, including precise cursor movements and even creating digital art. The integration with paint applications takes this innovation a step further, enabling users to paint and be artistic using their hands. This project signifies a big advancement in the field of Human-Computer Interaction.

VII. REFERENCES

1. *Abhilash S S1, Lisho Thomas2, Naveen Wilson3, Chaithanya C4.(2018), "VIRTUAL MOUSE USING HAND GESTURE", International Research Journal of Engineering and Technology (IRJET).*
2. *Bharath Kumar Reddy Sandra, Katakam Harsha Vardhan, Ch. Uday, V Sai Surya, Bala Raju, Dr. Vipin Kumar . (2022), "GESTURE-CONTROL-VIRTUAL-MOUSE", International Research Journal of Modernization in Engineering Technology and Science.*
3. *Vijay Kumar Sharma, Vimal Kumar, Md. Iqbal, Sachin Tawara, Vishal Jayaswal.(2020), "Virtual Mouse Control Using Hand Class Gesture",ISSNNO:1869-9391. ,ResearchGate.*
4. *Alexander Masurovsky , Paul Chojecki, Detlef Runde , Mustafa Lafci, David Przewozny and Michael Gaebler.(2020), "Controller-Free Hand Tracking for Grab-and-Place Tasks in Immersive Virtual Reality: Design Elements and Their Empirical Study", Multimedia Technologies and Interaction.*
5. *Prachi Agarwal, Abhay Varshney, Harsh Gupta, Garvit Bhola, Harsh Beer Singh.(2022), "Gesture Controlled Virtual Mouse", View at google scholar.*
6. *Ranjith GC, Saritha Shetty (2023), "IMPLEMENTING A REAL TIME VIRTUAL MOUSE SYSTEM USING COMPUTER VISION", International Research Journal of Modernization in Engineering Technology and Science.*
7. *V. Bazarevsky and G. R. Fan Zhang. On-Device, "Media Pipe for Real-Time Hand Tracking", Google Research.*
8. *V. V. Reddy, T. Dhyanchand, G. V. Krishna and S. Maheshwaram (2020), "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition" IEEE-HYDCON, 2020, pp. 1-5*
9. *Israth Jahan, Mohammad Likhan, Md. Omar Faruk Hasan, Shanta Islam, Nurul Ahad Farhan (2023), "Artificial Intelligence Virtual Mouse", ResearchGate.*
10. *V. Tiwari, C. Pandey, A. Dwivedi and V. Yadav (2020), "Image Classification Using Deep Neural Network", International Conference on Advances in Computing, Communication Control and Networking (ICACCCN).*
11. *K. Bharath Reddy, Md. Fayazuddin, M. John Manohar(2022), "Artificial Intelligence Based Virtual Mouse", International Journal of Computer Science and Mobile Computing.*