International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 06 | Jun 2026 www.irjet.net p-ISSN: 2395-0072

Virtual Interaction System Using Open CV and Media pipe for Touch-Free Human-Computer Interaction

Dr. Manoj Kumar M[1], Dayananda J [2], Ravi[3], Chidananda G [4], Harsha H S [5]

[1]*Associate Professor, Dept. of CSE, Jyothy Institute of Technology, Karnataka, India*
[2345]*Student, Dept. of CSE, Jyothy Institute of Technology, Karnataka, India*

Abstract - *Touchless interaction is emerging as a vital component of human-computer interaction (HCI), especially in health- care, education, and public environments. This paper presents a real-time gesture-based system using OpenCV and Mediapipe for intuitive user interaction with digital systems. The developed solution incorporates modules like a virtual mouse, air canvas, background changer, and PowerPoint controller, offering natural interaction through hand gestures alone. The system eliminates the need for traditional peripherals and fosters accessibility, safety, and innovation. Testing and evaluation reveal its practicality, real- time responsiveness, and adaptability across different environments and user scenarios.*

Key Words: Gesture Recognition, OpenCV, Mediapipe, Virtual Mouse, Background Changer, Air Canvas, Python, HCI, Real-time Interaction

1. INTRODUCTION

Computer vision, a dynamic subfield of artificial intelli- gence, empowers machines to interpret and make decisions based on visual inputs. OpenCV, an open-source library, is extensively used for real-time image and video processing, object detection, and gesture recognition. This project lever- ages OpenCV and Python to create a virtual interaction system that enables touch-free computer control via hand gestures detected through a webcam. The motivation for this work stems from the need for hygienic, intuitive, and accessible interfaces, particularly in scenarios where traditional peripherals are impractical or undesirable.

2. LITERATURE REVIEW

2.1 OpenCV Libraries for Chemical Imaging

A OpenCV has been widely adopted for a variety of realtime image processing applications, including chemical imaging. Its open-source nature and extensive library of functions make it a flexible tool for tasks such as object detection, feature extraction, and image segmentation. The successful use of OpenCV in chemical imaging demonstrates its robustness and adaptability for diverse domains, laying a strong foundation for its application in gesture recognition systems.

2.2 Gesture Recognition using Image Processing

Computer vision techniques have been effectively used to recognize hand gestures, providing a natural and intuitive interface for human-computer interaction. By leveraging image processing methods such as skin color segmentation, contour detection, and feature tracking, researchers have developed systems capable of interpreting static and dynamic gestures. These advancements have paved the way for more sophisticated gesture-controlled applications.

2.3 Hierarchical HMMs for Sign Language

Hierarchical Hidden Markov Models (HMMs) have been employed to address the temporal complexity of sign language recognition. By structuring the recognition process into multiple layers, these models can capture both the spatial configuration of hand shapes and the sequential nature of gestures. This layered approach enhances the system's ability to accurately classify a wide range of sign language expressions.

2.4 3D CNNs for Gesture Detection

Hierarchical Hidden Markov Models (HMMs) have been employed to address the temporal complexity of sign language recognition. By structuring the recognition process into multiple layers, these models can capture both the spatial configuration of hand shapes and the sequential nature of gestures.

2.5 Vision-Based HCI Survey

Comprehensive surveys on vision-based human-computer interaction (HCI) have compared the effectiveness of static and dynamic gesture recognition techniques. These studies highlight the

strengths and limitations of various approaches, including rule-based, machine learning, and deep learning methods. The insights gained from such surveys inform the design of more robust and user-friendly gesture recognition systems.

3. RESEARCH GAP

Virtual interaction systems that we have now do not work well when the lighting is not good or when the background is messy. This makes it hard for these systems to detect what we are doing with our hands when we are using them. Many systems can only understand a hand movements and they do not have many features, like a virtual mouse or the ability to draw things using our hands. We need a system that's easy to use and works in real time without needing super expensive computers or complicated setups. We can use OpenCV and MediaPipe to make an interaction system that is cheap and works well.

Many gesture recognition systems today are only good for one or two things. They do not let users do things at once. We need a system that lets users control a mouse draw in the air and use gestures all in one place. These systems are important for making interactions easier. Advanced systems that let people interact in new ways need expensive equipment. They also
need a lot of computer power. This makes it hard for many people to use them. A system that is low-cost works in time and is easy to use would be much better. Using tools, like
OpenCV and MediaPipe can make this possible. Such a system would make it easier for more people to use. Would work well in real-life situations.

Virtual interaction systems that exist now need lighting and simple backgrounds to recognize gestures correctly. If the light is not right or the background is messy it can be harder to track hand movements. Some systems do not work well when people are using them in time. This makes it hard to use them in life. So making these systems better and more accurate is something that people are still trying to figure out. A lot of systems that use gestures are only made for one thing.

4. PROPOSED SYSTEM

The system I'm talking about lets people use their hands to control computers. Its made with OpenCV and MediaPipe. It works by using a webcam to track hand movements. The computer then figures out what the hand is doing. This way users do not need a mouse or keyboard. The system works in time which means it can keep up with fast hand movements. It uses special computer vision techniques to understand what the hand gestures mean. The OpenCV and MediaPipe tools are key, to making this work. The hand gestures let users interact with computers in a natural way.

The video capture module gets video from a webcam all the time. It looks at each picture it gets from the camera to see
what is happening. The video capture module uses OpenCV to handle the video and process each picture. This makes things work well. The pictures, from the webcam are changed so they can be used for things. The video capture module makes sure it gets each picture from the camera without stopping or slowing down. It keeps getting pictures from the camera so the video capture module can do its job with the video and the pictures.

The hand detection module looks at the users hand in video and follows it. We use MediaPipe to find the hand and fingers well. The module always watches how the hand is moving now. Finding landmarks, on the hand helps us see what the hand is doing. The system tracks the hand well and does it quickly.

The gesture recognition module looks at where your
fingersre how you move your hand to figure out what gesture you are making. Different gestures are linked to commands that your system can understand. For instance you can use hand movements to move the cursor on the screen click on things or scroll through pages. The distance, between your fingers can also tell the system what to do. The module works on understanding your gestures away so that it can respond quickly. When the system can accurately understand your gestures it makes it easier to interact with it. Reduces mistakes.

The new system has a feature that lets you control the mouse with your hands. You can move the cursor on the screen by moving your fingers. To do things you use different hand movements. For example you use one movement for the left click, another movement for the click and another movement for scrolling. The cursor, on the screen moves when you move your hand. This happens at the same time. The virtual mouse control feature is really cool because it uses hand gestures. The system tracks the movement of your fingers to move the cursor.

5. METHODOLOGY / SYSTEM ARCHITECTURE

A. System Workflow

Input Capture: Webcam collects real-time hand images.

Hand Detection: Mediapipe identifies landmarks.
Gesture Recognition: Recognizes gestures (click, drag, thumbs-up, etc.).

Command Mapping: Links gestures to system functions.

Module Execution: Activates features like air drawing or slide navigation.

B. Modules Overview

TABLE 1 SYSTEM MODULES AND DESCRIPTIONS

Module	Description
Virtual Mouse	Use fingers to move cursor, click or scroll.
Air Canvas	Draw on screen using finger as a brush.
Background Changer	Replaces background without a green screen.
PPT Controller	Slides are controlled by gestures like swipe left/right.

C. Module-wise Breakdown

Camera Module: Captures live feed and feeds frames into the pipeline.

Detection Module: Uses Haar Cascades and Mediapipe for filtering hand regions.

Feature Extraction Module: Identifies shape, direction, movement, and computes coordinate deltas.

The virtual interaction system is made up of a design that lets people talk to computers in real time using gestures. This system has a main parts: it gets information from the user it processes that information it figures out what gestures mean and it shows the user what is happening. The system uses a webcam to get information, from the user. The webcam takes pictures of the user. Sends them to the computer. The computer then looks at these pictures to understand what the user is doing. The virtual interaction system uses these pictures to figure out what the user wants to say.

The system has a part that works with OpenCV to do things to images like change the frame make it bigger or smaller add filters and change the colours. It then sends these changed
images to the part that finds hands. The hand tracking part helps the system see what the user is doing with their hands and understand the movements and gestures they are making.

The gesture recognition module looks at where the fingersre how they move after it finds the hand landmarks. It uses this information to figure out what gesture is being made. The gesture recognition module checks the finger positions and movement patterns to see if they match gestures.

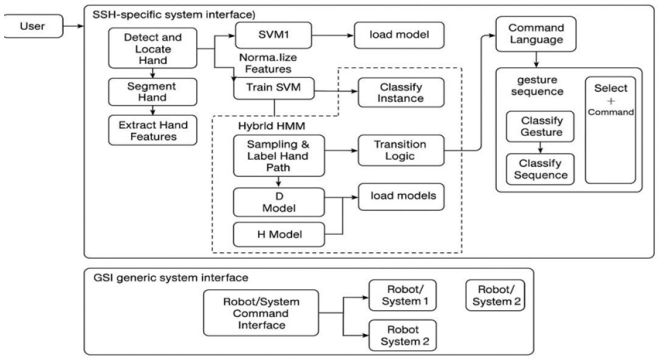

Fig -1: System Architecture

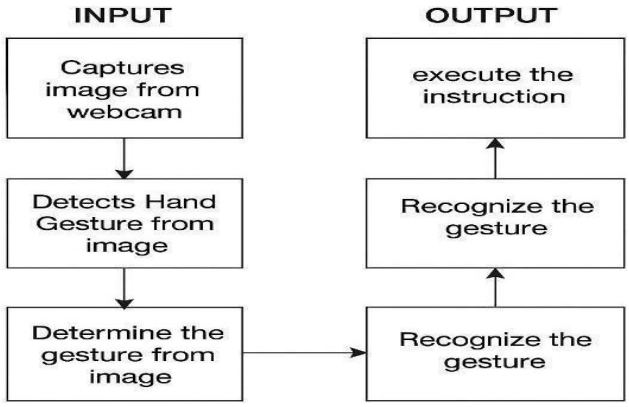

Fig-2: Data Flow Diagram – Level 1

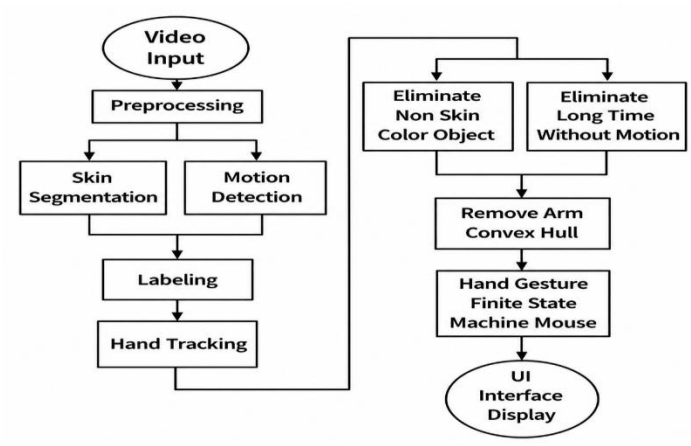

Fig -3: Data Flow Diagram – Level 2

6. IMPLEMENTATION

A. Tools and Technology Stack

- **Programming Language:** Python
- **Libraries:** OpenCV, Mediapipe, NumPy
- **Hardware:** Standard webcam, 4GB+ RAM

B. Implementation Steps

1) Setup webcam stream and capture frames.
2) Use Mediapipe for hand landmark detection.

3) Apply logic to identify gesture shape and motion.
4) Execute matching action (cursor movement, draw, slide change).
5) Display output via GUI or console.

C. Algorithm Snippet

```
if fingers == [ 0 , 1 , 0 , 0 , 0 ] :
# Index finger up
    cursor . move_to ( x , y )
elif fingers == [ 0 , 1 , 1 , 0 , 0 ] :
# Index + Middle
    Perform_click ( )
```

7. CONCLUSION

The Virtual Interaction System effectively recognizes hand gestures using a webcam and enables real-time virtual interaction without traditional input devices. It is intuitive, platform- independent, and easily extendable. With modules such as the Virtual Mouse and Air Canvas, it enhances both usability and accessibility for a wide range of users.

8. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Dr. Manoj Kumar M, Associate Professor, Department of Computer Science and Engineering, for his continuous guidance, valuable suggestions, and encouragement throughout the development of this work. His support played an important role in shaping the direction and quality of this study.
The authors also extend their thanks to the Department of Computer Science and Engineering, Jyothy Institute of Technology, for providing the necessary resources and environment to carry out this work successfully. Finally, the authors would like to acknowledge the support and cooperation of all team members, whose collective effort and coordination made the completion of this work possible.

9. REFERENCES

1) Rautaray & Agrawal, "HCI Gesture Recognition Survey," Springer, 2012.
2) Wachs et al., "Vision-Based Hand Gesture Applications," CACM 2011.
3) Lu & Picone, "Fingerspelling Recognition Using HMMs," ICIP 2013.
4) Molchanov et al., "3D CNNs for Gesture Recognition," CVPR 2015.
5) Zhang & Tian, "RGB-D for Activity Recognition," JRTIP, 2012.
6) Shan et al., "Facial Expression Detection Using LBP," Image and Vision Computing, 2009.
7) Kim & Kim, "Finger Segmentation for Virtual Mouse," IEEE TCE, 2016.
8) Singha & Das, "KLT for Gesture Recognition," IJCA, 2013.
9) C. B, K. S. Raja S, R. M, and R. S. Vignesh, "Eyeball Movement Cursor Control Using OpenCV," ECS Transactions, vol. 107, no. 1, pp. 10005–10011, Apr. 2022, doi: https://doi.org/10.1149/10701.10005ecst.
10) Murata, "Eye-gaze input versus mouse: Cursor control as a function of age," International Journal of Human-Computer Interaction, vol. 21, no. 1, pp. 1–14, Sep. 2006, doi: https://doi.org/10.1080/10447310609526168.
11) M. A. Azhari Halim, Mohd. F. I. Othman, A. Z. Z. Abidin, E. Hamid, N. Harum, and W. M. Shah, "Face Recognition-based Door Locking System with Two-Factor Authentication Using OpenCV," 2021 Sixth International Conference on Informatics and Computing (ICIC). IEEE, Nov. 03, 2021. doi: 10.1109/icic54025.2021.9632928.
12) R. TH. Hasan and A. Bibo Sallow, "Face Detection and Recognition Using OpenCV," Journal of Soft Computing and Data Mining, vol. 2, no. 2. Penerbit UTHM, Oct. 15, 2021. doi: 10.30880/jscdm.2021.02.02.008.