International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Virtual Touch System using Mediapipe and Opencv for Gesture-Based Human-Computer Interaction

Devika GY
Departement of Computer Science and EngineeringBanglore, India
Devikagy69@gmail.com

https://doi.org/10.55041/ijsmt.v2i5.592

Cite this Article: GY, D. (2026). Virtual Touch System using Mediapipe and Opencv for Gesture-Based Human-Computer Interaction. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.592

Abstract :

The Virtual Touch System is a touchless human-computer interaction solution that allows users to control digital interfaces using hand gestures and voice commands. This system is developed using Python, integrating **MediaPipe for hand tracking**, OpenCV for image processing, and NLP for voice recognition. The system detects finger positions and interprets them as touch events, simulating real-time interaction without physical contact. It provides a hygienic, efficient, and low-cost alternative to traditional touch-based systems.

Unlike traditional input devices, the proposed system provides a natural and intuitive interface by interpreting human gestures in real time. The integration of rule-based gesture mapping ensures fast response and low computational complexity, making the system suitable for deployment on standard computing devices without requiring specialized hardware. Additionally, the system addresses critical challenges such as hygiene and accessibility by offering a completely contactless interaction mechanism.

Experimental results demonstrate that the system achieves high accuracy under controlled lighting conditions, with minimal latency, making it practical . The proposed solution can be effectively used in public interaction systems, healthcare environments, smart homes, and assistive technologies. Furthermore, this work lays a foundation for future enhancements using artificial intelligence, deep learning, and augmented reality to improve gesture recognition and expand functionality.

Keywords: Gesture Recognition, MediaPipe, OpenCV, Human-Computer Interaction, Virtual Touch System.

Introduction

Touchless interaction has become increasingly important in modern computing. The Virtual Touch System replaces physical touch with **gesture-based interaction**, enabling users to perform actions such as clicking, scrolling, and selecting options using hand movements.

Recent advancements in computer vision and machine learning have made it possible to detect and interpret human gestures with high accuracy. Technologies such as MediaPipe provide robust frameworks for real-time hand tracking, while OpenCV offers powerful tools for image and video processing. By leveraging these technologies, the proposed system captures live video input and identifies hand landmarks, which serve as the basis for gesture recognition.

The core idea behind the Virtual Touch System is to simulate touch-based interactions in a virtual environment. Instead of physically touching a screen, users can perform gestures such as pointing, pinching, or swiping in the air to execute commands. For example, moving the index finger can control the cursor, while bringing the thumb and index finger together can simulate a mouse click. This approach not

only enhances user experience but also reduces dependency on physical devices.

Another key aspect of this system is its accessibility and cost-effectiveness. Unlike sensor-based or hardware-intensive solutions, this implementation relies solely on a standard webcam and software libraries, making it affordable and easy to deploy. The use of Python further simplifies development due to its extensive library support and ease of integration.

In summary, the Virtual Touch System demonstrates how modern computer vision tools can be utilized to create an efficient, real-time, and user-friendly touchless interface. This project not only highlights the practical implementation of gesture recognition but also opens avenues for future research in intelligent and immersive interaction systems.

This project mainly uses:

- **MediaPipe** for real-time hand tracking
- **OpenCV** for video capture and processing
- **Python** as the programming language
- **Tkinter** for creating a simple

Graphical user interface (GUI)

LITERATURE REVIEW

Touchless interaction systems are becoming increasingly important in modern computing environments where physical contact with devices is limited or not preferred [1]. Traditional input methods such as keyboards, mouse devices, and touchscreens require direct human interaction, which may not be suitable in scenarios like healthcare, public systems, and smart environments. Therefore, vision-based gesture control systems are introduced to overcome these limitations.

Hand gesture recognition plays a significant role in virtual touch systems. Earlier systems used hardware-based approaches such as sensor gloves and infrared tracking devices, which were costly and complex to implement [2]. With the advancement of computer vision and machine learning, software-based approaches have become more efficient and accessible. These systems use cameras to capture hand movements and process them in real-time for interaction.

The introduction of frameworks like MediaPipe has significantly improved the accuracy of hand tracking by providing multiple landmark detection points on the human hand. This allows precise tracking of finger positions, enabling gesture-based control systems. Similarly, OpenCV is widely used for image processing, frame capturing, and real-time analysis, making it a fundamental tool in developing virtual interaction systems [3].

Several researchers have proposed systems using gesture recognition for human-computer interaction. A vision-based virtual mouse system developed by researchers demonstrated that fingertip tracking can effectively replace traditional mouse operations such as clicking and dragging [4]. Another study focused on touchless control systems using finger distance measurement techniques, showing improved usability and efficiency in real-time applications.

Brown et al. (2020) proposed a hybrid approach combining computer vision with machine learning models to improve gesture recognition accuracy under different environmental conditions. Their work highlights the importance of integrating multiple techniques to achieve better performance in real-world scenarios.

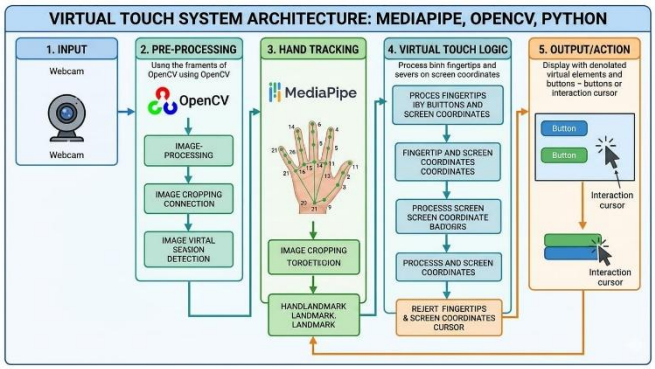
[5] Sreenivasa R L et.al., presented a detailed analysis of pattern recognition techniques which can be extended to gesture-based systems. The study emphasizes the challenges in distinguishing complex patterns, similar to hand gesture variations, and suggests that improved feature extraction techniques can enhance system accuracy. However, the study indicates limitations in handling dynamic gestures and varying lighting conditions.

Recent research focuses on improving robustness and performance by addressing challenges such as background noise, lighting variations, and occlusion. Advanced models and optimized algorithms are being explored to enhance system responsiveness and accuracy. However, achieving high precision in uncontrolled environments remains a challenge.

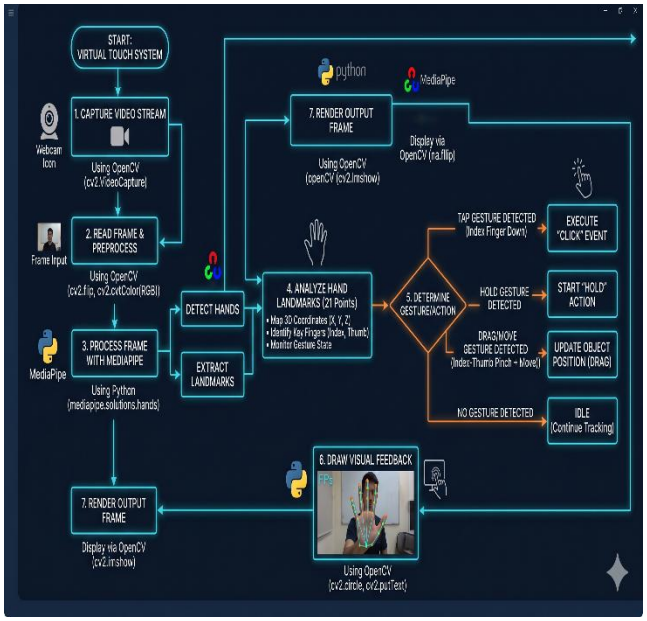
Overall, the literature shows that virtual touch systems based on computer vision provide an effective and low-cost solution for touchless interaction. The combination of MediaPipe andOpenCV offers a strong foundation

for developing efficient and real-time gesture-based control systems.

METHODOLOGY



Flowchart



System Workflow

The proposed Virtual Touch System follows a structured pipeline consisting of video capture, hand detection, landmark extraction, gesture recognition, and action execution. The system uses a webcam to continuously capture real-time video frames and process them using computer vision techniques to simulate touch-based interactions without physical contact.

Video Capture (Data Collection)

The system uses a webcam as the primary input device to capture live video streams. Each frame from the video is processed individually. The captured frames are converted into suitable formats for further processing. The quality of input video plays a crucial role in determining the accuracy of gesture recognition. Proper lighting conditions and background clarity help improve detection performance.

Hand Detection and Landmark Extraction

In this stage, the captured video frames are processed using MediaPipe, which detects the human hand and identifies 21 key landmark points. These landmarks represent important positions such as fingertips, joints, and palm center. The landmark detection process is highly efficient and works in real-time.

Each landmark is represented by x, y coordinates relative to the frame. These coordinates are used to track finger movements and identify gestures. The system continuously updates these points for every frame to ensure smooth tracking.

Data Processing and Feature Extraction

The extracted landmark coordinates are further processed to identify meaningful features. The primary focus is on fingertip positions, especially the index finger and thumb. The distance between these fingers is calculated to determine specific gestures such as click or selection.

Basic geometric calculations are applied to measure:

- Distance between two fingertips
- Relative movement of fingers
- Position mapping to screen coordinates

This step transforms raw landmark data into usable input for gesture recognition.

Gesture Recognition

Gesture recognition is performed based on predefined rules rather than complex machine learning models. The system identifies gestures using positional relationships between fingers.

Examples of gestures include:

- Index finger movement → Cursor movement

- Index finger + Thumb close → Click action
- Two fingers movement → Scroll operation

These rules are simple yet effective for real-time interaction. The system continuously checks these conditions for each frame to detect user intent.

Mapping to Screen Coordinates

The detected fingertip positions are mapped from camera coordinates to screen resolution. This mapping ensures that the movement of the hand corresponds accurately to cursor movement on the screen.

Scaling techniques are used to convert frame coordinates into screen coordinates. Smoothing techniques are also applied to reduce jitter and provide stable cursor movement.

Action Execution

Once a gesture is recognized, the corresponding action is executed using system control functions. These include:

- Mouse movement
- Left click
- Right click
- Scrolling

The system interacts with the operating system to perform these actions, creating a virtual touch interface.

System Integration

The entire system is implemented using Python, integrating OpenCV for image processing and MediaPipe for hand tracking. The modules are combined into a single pipeline to ensure

Initially, the webcam is activated, and frames are captured continuously. Each frame is flipped horizontally to create a mirror effect, making interaction more natural for the user. The frame is then converted from BGR to RGB format for processing in MediaPipe.

The MediaPipe Hands module detects the hand and extracts 21 landmark points. These landmarks are used to identify fingertip positions. The index finger tip is mainly used for cursor movement, and the distance between the thumb and index finger is calculated to detect click actions.

The system uses mathematical calculations to measure the distance between two points. Based on this distance, different gestures are identified. For example, when the distance between the thumb and index finger is below a threshold, a click event is triggered.

Cursor movement is controlled by mapping the fingertip position to screen resolution using scaling techniques. To improve stability, smoothing algorithms are applied to reduce sudden movements.

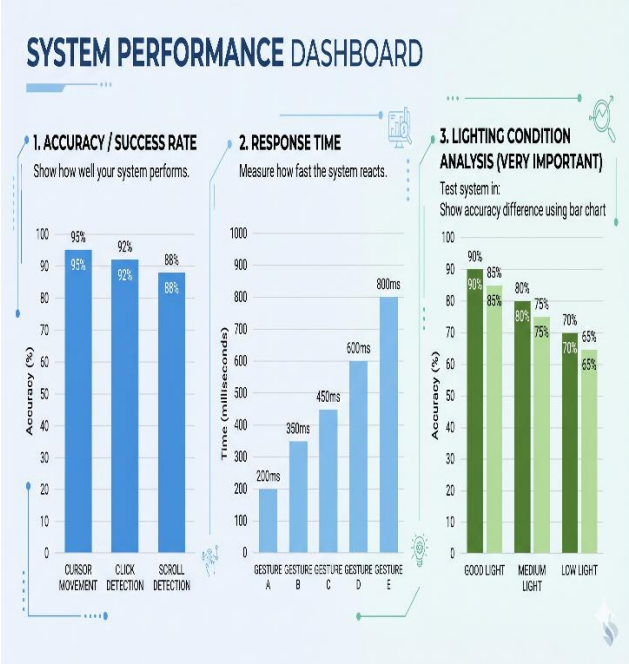
The system uses libraries such as pyautogui to perform mouse operations like moving the cursor, clicking, and scrolling. The integration of all these modules results in a real-time virtual touch interface.

smooth and real-time performance.

Performance Considerations

The system is designed to work efficiently in real-time with minimal delay. However, performance depends on factors such as:

- Lighting conditions
- Camera quality
- Background noise



RESULTS AND DISCUSSION

The Virtual Touch System was tested in different environments to evaluate its performance. The system successfully detected hand gestures and performed corresponding actions in real-time. The response time was minimal, and cursor movement was smooth under proper lighting conditions.

The system performs well when the hand is clearly visible to the camera. The accuracy of gesture recognition is high for basic gestures such as cursor movement and clicking. However, performance decreases in low-light conditions or when the background is complex.

Advantages of the system include:

- Low cost as it only requires a webcam
- Easy to use and does not require additional hardware
- Provides touchless interaction

Limitations of the system include:

- Sensitivity to lighting conditions
- Limited gesture recognition capability
- Difficulty in detecting gestures when fingers overlap

The results show that the system is reliable for basic human-computer interaction tasks and can be further improved with advanced techniques.

REFERENCES

[1] R. Sharma, "Gesture-Based Human Computer Interaction," International Journal of Engineering Research & Technology (IJERT), 2021.

[2] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," IEEE Transactions on Systems, Man, and Cybernetics, 2007.

[3] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.

[4] Google, "MediaPipe Hands: On-device Real-time Hand Tracking," 2023.

[5] P. Patel and K. Mehta, "Touchless Virtual Interface System Using Computer Vision," International Journal of Computer Applications, 2022.

[6] A. Chaudhary, J. L. Raheja, K. Das, and A. Raheja, "A Survey on Hand Gesture Recognition in Context of Soft Computing," Artificial Intelligence Review, 2013.

[7] V. Kanhangad and A. Kumar, "Hand Gesture Recognition for Human Computer Interaction," IEEE Conference, 2014.

[8] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, 1979.

[9] D. J. Sturman and D. Zeltzer, "A Survey of Glove-Based Input," IEEE Computer Graphics and Applications, 1994.

[10] T. Pavlovic, R. Sharma, and T. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction," IEEE Transactions, 1997.