



IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025 DOI: https://doi.org/10.22214/ijraset.2025.70798

www.ijraset.com

Call: 🕥 08813907089 🔰 E-mail ID: ijraset@gmail.com



International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 13 Issue V May 2025- Available at www.ijraset.com

Hand Tracking-Based Virtual Mouse System

Mr. Soham Patil, Mr. Somesh Chincholkar, Mr. Dhairyasheel Pawar, Mr.PratikMunde, Prof. Afrin Sheikh Savitribai Phule Pune University, Department of Engineering Science KJ College Of Engineering and Management Research, Pune, India

Abstract: This study presents a real-time hand gesture-based virtual mouse system leveraging computer vision technologies. The system utilizes a webcam, OpenCV, Mediapipe, PyAutoGUI, and Pynput to allow touchless cursor movement, left/right clicks, double click, and screenshot functionality. The goal is to improve human-computer interaction, especially in hygienic or accessibility-focused environments. This method replaces traditional physical mouse devices with intuitive gestures, enhancing ease of use and inclusivity.

I. INTRODUCTION

In the digital age, touchless interaction has become vital across domains including healthcare, gaming, AR/VR, and assistive technology. Hand tracking provides a natural interface between users and machines. This project aims to build a gesture-based virtual mouse system that uses hand landmarks detected through a webcam to simulate mouse events.

II. LITERATURE REVIEW

Hand tracking techniques have evolved from color segmentation and contour analysis to deep learning models. Libraries like Mediapipe provide fast, lightweight hand landmark detection. Previous systems faced challenges such as lighting sensitivity, gesture misinterpretation, and the need for wearable devices. Our system addresses these by using a simple webcam and machine-learned landmarks.

III. METHODOLOGY

The methodology consists of the following steps:

- 1) Webcam captures live video.
- 2) OpenCV processes frames.
- 3) Mediapipe detects 21 hand landmarks.
- 4) Angles and distances between landmarks are calculated.
- 5) Specific gestures are mapped to mouse actions using PyAutoGUI and Pynput.

Cursor movement is enabled when the index finger is extended. Click gestures are based on calculated angles between joints, ensuring reliable detection. Screenshot functionality is triggered using a unique hand pose.

IV. IMPLEMENTATION

The system was developed in Python using the following libraries:

- 1) OpenCV: For capturing and displaying frames.
- 2) Mediapipe: For real-time hand landmark tracking.
- *3)* PyAutoGUI: For executing screen actions like cursor move, click, and screenshot.
- 4) Pynput: For precise mouse button control.

The application tracks one hand, evaluates gestures in each frame, and simulates the corresponding mouse action. Landmark coordinates are scaled to screen resolution to ensure natural movement.

V. RESULTS & DISCUSSION

Testing was done under various lighting conditions and backgrounds. Results showed accurate cursor movement and gesture detection under proper lighting. The click and screenshot gestures worked reliably with minimal delay. Limitations include sensitivity to hand positioning and occlusions. The system performs best with a clear background and adequate lighting. Improvements could include adaptive thresholding and gesture training customization.



International Journal for Research in Applied Science & Engineering Technology (IJRASET)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 13 Issue V May 2025- Available at www.ijraset.com

C:\WINDOW5\system32\cmd. × + v			📃 🛛 👥 MB main 🗸						Close
INFO: Created TensorFlow Lite XNNPACK delegate for CPU. WARNING: All log messages before absl::InitializeLog() is ca	lled are written	۵	Project ~		📌 main.			¢,	
to STDERR W0000 00:00:1746142634.296807 22008 inference_feedback_mar	- k_manager.cc:114] Fee inference. Disabling on_calculator.cc:186]		~ D hand tracking	ng system C:\Users\			def ma 🗛	2 🗛 4 ^ ~	6
Webbe 09:00:1746114254.42968.79087 22000 Internet_reence.reentaxt_Bat Bdack manager requires a model with a single signature inter support for feedback tensors. 00000 00:00:17461142543.237222 22000 inference_feedback_man dback manager requires a model with a single signature infe support for feedback tensors. 00000 00:00:17461142447.050552 15228 landmark_projection_ct Using NORM_RECT without IMAGE_DIMENSIONS is on use PROJECTION_MATRIX.			📌 main.py						
				nshot_357.png					
			🔀 my_screen	nshot_525.png				frameRG	
			🔀 my_screen	nshot_621.png					
			my_screenshot_820.png						
			≡ run.bat						
			🔁 util.py						
			> Ilh External Libra	aries					
			> Scratches and	d Consoles					
I Fra	m.a.		_						
Hatelet and the second seco	ile .			U ^				detect_	
			Aller .					cv2.ims	
	0/20	1	(if cv2.	
		3	1						
		3							
	Welle		1000						
	F	123							
	1 / /		1						
			A REAL PROPERTY.						
	¥						🧣 main(
			2 2 3	-					
	×							_	

VI. CONCLUSION

This virtual mouse system showcases the practical potential of real-time hand tracking for touchless HCI. It demonstrates that standard webcams and lightweight libraries are sufficient for effective mouse control. The system can serve as an assistive tool for differently-abled users or be deployed in environments requiring hygiene. Future enhancements could involve multi-hand detection and custom gesture training for expanded control.

REFERENCES

- [1] Taylor, J. et al., 'Real-Time Hand Tracking with a Deep Sensor'
- [2] Zhang, Z., 'Hand Gesture Recognition with EMG Sensors'
- [3] Mueller, C., 'Low-Latency Hand Tracking Using Event Cameras'
- [4] Tompson, J., 'Real-Time Continuous Pose Recovery of Hands'
- [5] Zhang, F., 'End-to-End Hand Mesh Recovery' (ECCV 2020)
- [6] Sridhar, S., 'Real-Time Hand Tracking with a Color Glove'











45.98



IMPACT FACTOR: 7.129







INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 🕓 (24*7 Support on Whatsapp)