OKCL
Creating a Knowledge Lit World

ocac
Odisha Computer Application Centre

INTERNSHIP PROJECT REPORT

On

HAND GESTURE DETECTION

SUPERVISED BY :
SAMBIT SUBHASISH SAHU
Educator, OCKL

PRESENTED BY :
SATYAJIT PATRA
SUBHAKANKANTA MISHRA
BADAL KUMAR NAYAK

ABSTRACT

With the continuous advancement in human-computer interaction (HCI), traditional input devices like the mouse and keyboard are evolving toward more intuitive, touchless technologies. This project presents a real-time computer vision-based system using OpenCV and MediaPipe that allows users to interact with a computer using only hand gestures and finger movements—eliminating the need for physical devices.

The system comprises two core components:

- AI Virtual Mouse, which uses hand tracking to control cursor movement, perform left/right clicks, drag-and-drop, and scroll actions based on the distance between fingers.
- Hand Gesture Detection, which interprets custom gestures (e.g., Fist, Open Palm, Victory, Thumbs Up, SpiderMan, Gun, etc.) to recognize meaningful hand signs for HCI or command execution.

The application employs a webcam for real-time video feed, processes each frame using MediaPipe Hands, and triggers relevant desktop actions using PyAutoGUI. The GUI interface is built using ttkbootstrap (Tkinter) for easy access and launching of these features.

This innovative solution demonstrates how AI-powered vision systems can simulate physical hardware, enhance accessibility, and provide a more natural interaction with computers—paving the way for gesture-controlled applications in fields such as automation, gaming, accessibility tools, and smart environments.

TABLE OF CONTENTS

INTRODUCTION

In the digital era, the need for efficient and seamless interaction between humans and computers has grown rapidly. Conventional devices such as keyboards and mice have served as the primary tools for input. However, these physical interfaces often limit the scope of interaction, especially in dynamic or hands-free environments. This has led to the emergence of advanced technologies that support natural human gestures for system control and interaction.

This project introduces an AI-based computer vision solution that allows users to control their system using only hand gestures, without requiring physical contact with any hardware. Built using Python, OpenCV, and MediaPipe, the system utilizes a standard webcam to track hand movements, recognize gesture patterns, and simulate mouse functionality on the screen.

The system is divided into two main modules:

- AI Virtual Mouse – Enables the user to control the mouse cursor using index finger movement, perform left and right clicks using finger pinches, drag objects, and scroll vertically by using multi-finger gestures.
- Hand Gesture Detection – Identifies a range of predefined hand gestures (such as Fist, Victory, Thumbs Up, etc.) using geometric distances between landmarks, which can be mapped to specific functions or commands.

The GUI is designed using Tkinter with ttkbootstrap, offering a user-friendly interface to launch each module independently. This interface can be converted into a desktop application using PyInstaller, allowing for cross-platform deployment.

This project reflects the growing potential of AI-enhanced gesture recognition in creating smart, touchless systems that improve productivity, accessibility, and user experience. It demonstrates how intuitive control mechanisms can enhance traditional computing systems and set a foundation for future gesture-based interfaces.

OBJECTIVE

The primary objective of this project is to develop an AI-powered, touchless control system using computer vision and machine learning that enhances human-computer interaction through hand gesture recognition and virtual mouse control.

More specifically, the project aims to:

1. Eliminate the need for physical input devices like a mouse by replacing them with intuitive hand gestures tracked through a webcam.
2. Implement real-time hand tracking using OpenCV and MediaPipe to detect finger positions and gestures with high accuracy.
3. Map finger movements to mouse actions such as moving the cursor, left-clicking, right-clicking, dragging, and scrolling.
4. Recognize custom hand gestures (e.g., Victory , Thumbs Up , Fist , Spiderman, Gun , etc.) using distance-based landmark analysis and classification.
5. Provide a unified graphical interface (GUI) using Tkinter with ttkbootstrap for easy module access and interaction.
6. Convert the application into a deployable desktop app using PyInstaller, allowing it to be run without requiring any external Python setup.
7. Design a solution that is lightweight, efficient, and user-friendly, making it accessible for both technical and non-technical users.

This project is not only a technical exploration into real-time gesture detection but also a step toward natural, contactless user interfaces that promote hygiene, accessibility, and innovation across domains such as healthcare, gaming, and education.

TECHNICAL SPECIFICATION

1. Programming Language

   Python – Utilized for image processing, gesture recognition, GUI creation, and system automation.

2. Computer Vision & AI Frameworks

   Libraries Used:

   - MediaPipe – For accurate and fast real-time hand landmark detection.
   - OpenCV – For webcam capture and drawing annotations on video frames.
   - NumPy – For mathematical calculations such as Euclidean distances.
   - PyAutoGUI – To perform mouse movements, clicks, drag and scroll operations.

3. Gesture Logic & Detection

   Input Source:

   Live webcam feed captured via OpenCV.

   Detected Gestures:

   - Cursor Movement (via index finger)
   - Left Click (Index + Thumb)
   - Right Click (Middle + Thumb)
   - Drag & Drop (Index + Middle together)
   - Scroll (Index + Middle + Thumb closed)
   - Custom gestures like:
     - Fist, Victory, Call, Spiderman, Rock, Thumbs Up, etc.

   Detection Technique:

   - Euclidean distance between key finger landmarks.
   - Finger state analysis (open/closed) using y-coordinate comparison.

4. GUI Framework

   - Tkinter – Used to build a lightweight desktop interface.
   - ttkbootstrap – For modern styling and theming on top of Tkinter.

Features:

- Start AI Virtual Mouse
- Start Hand Gesture Detection
- Real-time status updates
- Prevents recursion and duplicate process launches

5. Deployment

Packaging Tool:

- PyInstaller – Used to create a single-file .exe application for Windows.

Flags Used:

- --noconsole – To hide background terminal
- --add-data – To bundle external .py files (gesture/mouse modules)

6. Environment Requirements

Hardware:

- HD Webcam (built-in or external)
- Minimum 4 GB RAM (8 GB recommended)
- Windows 10 or 11 (64-bit)

Software:

- Python 3.12
- Compatible library versions (NumPy < 2.0, protobuf < 5)

OUTCOME

The successful implementation of the AI Virtual Mouse and Hand Gesture Detection System demonstrates the powerful capabilities of computer vision in enhancing human-computer interaction. The system operates efficiently in real time using just a webcam, without any additional hardware or sensors.

Key Outcomes:

- Seamless Hand-Controlled Interface
  Users can control the mouse cursor, perform clicks, drag, scroll, and interact with applications through intuitive hand gestures.
- Real-Time Gesture Recognition
  Hand movements and gestures are detected with minimal delay using the MediaPipe hand landmark model, ensuring smooth and natural interaction.
- Increased Accessibility
  This system offers a hands-free alternative for users with motor disabilities or those in environments where traditional input devices are impractical.
- Modular & Extensible Design
  The system is built with modular Python scripts, allowing easy addition of new gestures or control actions in the future.
- User-Friendly Desktop GUI
  A clean and modern graphical interface allows users to start either the AI virtual mouse or gesture detection system with a single click, improving usability and accessibility.
- Performance Efficiency
  Despite being lightweight and running on standard hardware, the application achieves high responsiveness due to optimization in gesture processing and hand tracking.

REAL LIFE PROBLEM AND MY PROJECT

In today's digital age, human-computer interaction is predominantly dependent on physical input devices like the keyboard and mouse. However, several real-life challenges persist that limit their effectiveness, particularly in dynamic, constrained, or accessibility-sensitive environments.

Real-Life Challenges:

- Physical Limitations: Individuals with motor disabilities or hand impairments often struggle to use traditional input devices effectively.
- Touchless Interfaces in Hygiene-Sensitive Areas: In medical labs, clean rooms, or during pandemics (like COVID-19), users prefer non-contact interfaces to minimize surface transmission of pathogens.
- Immersive Interaction: Conventional devices do not offer immersive or intuitive control, especially for users working with AR/VR, robotics, or smart displays.
- Mobility Constraints: Professionals working in constrained spaces (e.g., field engineers, pilots, or technicians) might not have access to standard input devices at all times.

How This Project Solves the Problem:

This AI Virtual Mouse and Hand Gesture Detection System directly addresses the above challenges by introducing a touchless, gesture-based interface that requires only a webcam to function. Key benefits include:

- No External Hardware Needed: Uses just the computer's webcam and machine learning models for gesture detection, making it highly accessible.
- Gesture-Based Mouse Control: Allows users to move the cursor, click, drag, and scroll using natural hand gestures—eliminating the need for a physical mouse.
- Real-Time Feedback: The system processes gestures instantly and updates actions in real time, ensuring fluid interaction.
- Simple GUI for All Users: An intuitive interface enables users to launch either module (Virtual Mouse or Gesture Detection) without requiring technical expertise.
- Scalable for Broader Applications: Can be expanded for smart TVs, gaming, automotive dashboards, or smart home interfaces, enhancing interaction across domains.

FUTURE SCOPE

As technology evolves and user expectations shift toward more intuitive and immersive interaction models, this project holds immense potential for future development and expansion. The following points highlight how the current system can be enhanced and adapted to cater to a broader range of real-world applications:

- Enhanced Gesture Library
  Introduce more complex and customizable hand gestures for a wider range of commands, including multi-finger gestures, dynamic gestures (e.g., swipe, rotate), and dual-hand controls for multitasking.

- Voice + Gesture Hybrid Interface
  Integrate voice recognition alongside gesture detection to offer a multi-modal control system, making the interface more accessible, particularly for users with disabilities.

- AI Personalization
  Use machine learning to adapt to individual users' hand shapes, movement patterns, and preferences over time for improved accuracy and responsiveness.

- Cross-Platform Compatibility
  Package the system to support Linux, macOS, and mobile devices, extending its usability beyond Windows-based desktops.

- Web-Based Deployment
  Transform the application into a web-based tool using WebAssembly, TensorFlow.js, or MediaPipe in the browser—allowing usage on any device with a browser and webcam.

- AR/VR Integration
  Adapt the system for use in augmented and virtual reality environments to offer natural interaction in gaming, simulation, and virtual collaboration platforms.

- Advanced Computer Vision Models
  Incorporate deep learning-based hand pose estimation or 3D modeling for higher precision in recognizing complex gestures in varying lighting and background conditions.

CONCLUSION

The AI Vision Control Panel project, encompassing the AI Virtual Mouse and Hand Gesture Detection System, demonstrates how computer vision and machine learning can revolutionize human-computer interaction. By using only a webcam and Python-based libraries like OpenCV and MediaPipe, we have built an intuitive, touchless interface that replicates and extends traditional input methods like the mouse.

This project not only improves accessibility for users with physical limitations but also showcases the feasibility of gesture-based controls for general computing tasks. From moving the cursor with finger movements to executing clicks, scrolls, and detecting predefined gestures, the system reflects the power of real-time image processing and intelligent behavior mapping.

Throughout the development, we encountered and overcame several challenges related to accuracy, latency, and compatibility—especially while bundling the application as a desktop executable. These efforts have resulted in a functional, customizable, and engaging tool that opens the door to broader applications in assistive technology, gaming, smart environments, and AR/VR systems.

In conclusion, this project serves as a strong foundation for future innovations in gesture-controlled interfaces, emphasizing the growing relevance of AI-powered touchless interaction in modern computing.