

Virtual Mouse Project

Introduction

A **Virtual Mouse** is a software-based mouse that allows users to control the cursor without using a physical mouse. This project utilizes computer vision techniques to track hand gestures and movements, translating them into mouse operations such as moving the cursor, right-clicking, left-clicking, and double-clicking.

Objectives

- Develop a system that can replace a traditional mouse using hand gestures.
- Utilize computer vision and machine learning techniques for hand tracking.
- Ensure real-time performance with minimal latency.
- Implement basic mouse functionalities such as right-click, left-click, and double-click.

Technologies Used

- **Programming Language:** Python
- **Libraries:** OpenCV, Mediapipe, PyAutoGUI
- **Development Tools:** VS Code
- **Operating System:** Windows/Linux

Methodology

1. Capturing Hand Gestures

- Using **OpenCV** to access the webcam and capture real-time video frames.
- Applying **Mediapipe Hand Tracking** to detect key points of the hand.

2. Gesture Recognition and Mapping

- Recognizing hand landmarks and interpreting them for mouse actions.
- Mapping hand movements to control cursor position.

- Assigning specific gestures for left-click, right-click, and double-click.

3. Cursor Control

- Using **PyAutoGUI** to simulate mouse actions based on detected gestures.
- Implementing a smooth movement algorithm to enhance user experience.

4. Click Implementation

- **Left Click:** When the index finger tip and thumb tip come close within a predefined threshold.
- **Right Click:** When the middle finger tip and thumb tip come close within another threshold.
- **Double Click:** When the index finger tip and thumb tip come even closer than the left-click threshold.

Features

- **Cursor Movement:** Move the hand to control the cursor.
- **Left Click:** Pinch index finger and thumb together.
- **Right Click:** Pinch middle finger and thumb together.
- **Double Click:** Bring index finger and thumb even closer than the left-click threshold.

Challenges and Solutions

- **Hand Detection Accuracy:** Improved by using Mediapipe's advanced tracking.
- **Latency Issues:** Optimized by reducing unnecessary computations.
- **Gesture Recognition Errors:** Adjusted detection parameters for better accuracy.

Future Enhancements

- Add additional gestures for scrolling and dragging.
- Implement AI-based gesture learning for customization.
- Develop a mobile app version.

Source Code

GitHubLink - <https://github.com/aravinthan-126/Virtual-Mouse>

Conclusion

This Virtual Mouse project demonstrates how computer vision can replace traditional hardware peripherals, providing an innovative and interactive way to control a computer. By refining gesture recognition and optimizing processing speed, this system can become a practical alternative for users with accessibility needs or those looking for touchless control options.