PC Controller with Hand Gestures

Enrol. No. (s) - 9921103037, 9931103106, 9921103115
Name of Student (s) -Tanisha Bansal, Nikhil Raj, Shubham Mittal
Name of supervisor - Dr. Chetna Gupta

November 2024

Submitted in partial fulfillment of the Degree of Bachelor of Technology

in

Computer Science and Engineering

Department of Computer Science & Engineering and Information Technology
Jaypee Institute of Information Technology, Noida

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you our project, which is the result of a studied blend of both research and knowledge. We are grateful to Jaypee Institute of Information & Technology for this wonderful opportunity.

Furthermore, we express our earnest gratitude to our supervisor, Dr. Chetna Gupta, Professor, Department of CSE, our project guide, for her constant support, encouragement, and guidance. We are grateful for her cooperation and valuable suggestions.

Finally, we express our gratitude to all other members who are involved either directly or indirectly for the completion of this project.

Credentials Of Students :

Tanisha Bansal (9921103037)
Nikhil Raj (9921103106)
Shubham Mittal (9921103115)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Jaypee Institute of Information & Technology

Date: 25 November, 2024

Tanisha Bansal
9921103037

Nikhil Raj
9921103106

Shubham Mittal
9921103115

CERTIFICATE

This is to certify that the work titled "**PC Controller with Hand Gestures**" submitted by **Tanisha Bansal (9921103037), Nikhil Raj (9921103106)** and **Shubham Mittal (9921103115)** of B.Tech of Jaypee Institute of Information & Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Signature of the Supervisor: Dr. Chetna Gupta

Professor and Departmental Coordinator (CSE&IT)
25 November, 2024

ABSTRACT

The **PC Controller with hand gestures** is a touch-free interface that allows users to perform common computer actions like cursor movement, clicking, scrolling, and dragging through hand gestures alone. Designed as a hygienic and accessible alternative to traditional input devices, this system utilizes **MediaPipe** for detecting key hand landmarks and **OpenCV** for real-time image processing, enabling seamless gesture recognition and translation to on-screen actions. With the ability to interpret precise hand movements in real time, the virtual mouse offers a smooth, natural experience comparable to using a physical mouse.

This system is particularly valuable in settings where conventional devices may be impractical, such as healthcare environments or public spaces, and also provides a viable solution for individuals with physical disabilities, expanding digital accessibility. Unlike other gesture-recognition systems that rely on dedicated hardware like motion controllers or sensor gloves, this solution requires only a standard camera, making it both cost-effective and easy to deploy across different platforms.

By eliminating the need for physical peripherals, the PC Controller with Hand Gestures offers a flexible and forward-looking approach to human-computer interaction, with broad implications for improving accessibility and user experience in digital environments.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1 General Introduction

This project is an innovative advancement in human-computer interaction (HCI), which offers a new approach to control computers/laptops without physical contact. It enables users to perform various computer tasks—such as moving the cursor, clicking, scrolling, select, drag and drop eliminating the need for traditional input device like mouse. It is build using **MediaPipe** for gesture recognition and **OpenCV** for image processing, the system can detect and respond to user movements in real time, ensuring a seamless experience.

The concept of gesture control has broad applications, especially in environments where touch-free interactions are advantageous, such as in healthcare settings, public kiosks, and environments that prioritize hygiene. Additionally, for users with disabilities or limited mobility, this solution provides an alternative that enhances accessibility and usability without needing additional hardware, as it works with standard camera input.

This project addresses common limitations of traditional devices and expands the possibilities for a contactless, intuitive interface. It exemplifies a forward-thinking approach to digital interaction, leveraging the latest in computer vision to meet the growing demand for user-friendly, inclusive, and hygienic solutions in HCI.

1.2 Problem Statement

Traditional computer input device like mouse, is not always accessible or practical, especially in environments where touch poses hygiene risks or in cases where users have physical disabilities that limit their ability to interact with these peripherals. Recognizing these challenges, we aim to develop a touch-free solution that enables users to interact with computers seamlessly through hand gestures, eliminating the need for physical devices.

Our objective is to overcome the limitations of traditional input devices by offering a highly accessible, intuitive, and hygienic interface, built with widely accessible, affordable technology. By creating a hands-free interaction model that does not require specialized hardware beyond a standard camera, we're paving the way for versatile applications in various environments, from healthcare settings to public kiosks, where traditional devices may be impractical. This solution enhances both accessibility and user autonomy, expanding the possibilities of digital interaction.

1.3 Significance/ Novelty of the problem

Here are key points highlighting the significance of the problem:

1. **Enhanced Accessibility**: This solution provides an alternative input method that doesn't rely on physical peripherals, making digital interactions more accessible for people with physical disabilities, limited mobility, or injuries.

2. **Hygienic Interaction**: Particularly relevant in healthcare, cleanrooms, and public areas, the touch-free design reduces the risk of contamination and enhances cleanliness by eliminating the need for physical contact.

3. **Cost-Effectiveness and Accessibility**: Unlike existing gesture-based solutions that require dedicated hardware (like motion controllers), this project only requires a standard camera and open-source libraries (MediaPipe, OpenCV), making it affordable and accessible to a wider audience.

4. **Real-Time Responsiveness**: The system's reliance on real-time image processing ensures responsive feedback, making the interaction as close as possible to physical mouse input.

5. **Better User Experience**: Our project provides better user experience to users of operating the systems by making it more seamless and independent of physical devices.

6. **Adaptability to Different Environments**: The project's robustness in various lighting conditions and background settings makes it adaptable to diverse environments, from bright, open offices to dimly lit spaces.

7. **Scalability and Integration Potential**: Given its reliance on widely-used open-source libraries, this project can be easily integrated into various applications and scaled for use in different industries.

8. **Empowers Contactless User Interaction**: The project aligns with a growing trend towards contactless technology, where user demand is increasing for touch-free solutions in various fields, including retail, entertainment, and personal computing.

1.4 Empirical Study

The **Gesture Recognition** component of the project was evaluated through an experimental study to assess its accuracy and robustness across diverse settings. Using **MediaPipe's hand-tracking module**, the system identifies 21 key hand landmarks, which are processed to recognize gestures like clicks, drags, and swipes. To verify reliability, the gesture recognition was tested across various lighting conditions, hand sizes, and backgrounds, aiming to replicate real-world variability.

In this study, a diverse group of participants performed gestures under different lighting setups—from dim indoor lighting to bright natural light—and against varied backgrounds to test how well the model adapts. The system maintained a high recognition accuracy across different conditions, demonstrating its robustness. Notably, the system showed reliable performance in detecting gestures for users with a wide range of skin tones and hand shapes, reflecting its inclusivity and potential broad usability.

The **real-time processing** powered by OpenCV further enhances gesture accuracy, minimizing latency between gesture input and system response, which is critical for user experience. Overall, the study highlights the gesture recognition system's adaptability and accuracy, showing its effectiveness as a touch-free alternative for computer interaction in dynamic, real-world environments.

1.5 Brief Description of the Solution Approach

The project operates primarily through hand gesture recognition, providing an intuitive, touch-free way to control computer functions like cursor movement, clicking, scrolling, and dragging. The solution is built using **MediaPipe's hand-tracking module**, which detects 21 specific landmarks on the user's hand. These landmarks enable the system to interpret various hand movements and gestures accurately.

When a gesture is detected, **OpenCV** processes the input in real-time to track hand positioning and movements. The system can interpret gestures such as V-gesture, which simulates cursor movement, pinching, which simulates horizontal and vertical scroll depending on the movement of the hand, and open hand and closed fist to control the select and drag and drop feature. To achieve seamless interaction, the project integrates **computer vision techniques** to map gestures into actions, ensuring that users experience a response similar to a physical mouse.

This approach relies on standard hardware—a regular camera—making it widely accessible and cost-effective. The real-time processing capabilities allow for minimal lag, enhancing user experience and ensuring that each gesture is immediately translated into action on the screen. Overall, the solution leverages the synergy of **MediaPipe** and **OpenCV** to create a functional, affordable, and versatile alternative to traditional mouse-based input.

1.6 Comparison of existing approaches to the problem framed

When comparing existing approaches to contactless computer interaction, several solutions have been developed, each with varying limitations and requirements. Traditional gesture recognition systems often rely on dedicated hardware like **Leap Motion controllers** or **sensor-equipped gloves**, which can provide accurate hand tracking but come with drawbacks such as high costs, specialized setup, and dependency on additional equipment that may not be accessible to all users. These devices are also not universally compatible, making them less flexible across different platforms.

In contrast, camera-based gesture recognition systems, like the one implemented in this project, utilize standard webcams and open-source libraries, making them more accessible and affordable. Many camera-based systems, however, are limited to simple gestures, lacking precision and robustness when handling complex, real-time interactions such as clicking, dragging, or scrolling. Additionally, some approaches use basic image processing techniques, resulting in less accurate detection under variable lighting conditions or for users with diverse hand sizes and skin tones.

The PC Controller using Hand Gestures addresses these limitations by combining **MediaPipe's landmark-based tracking** with **OpenCV's image processing**, allowing for accurate, real-time gesture detection without specialized hardware. This approach stands out due to its adaptability, cost-effectiveness, and ability to function across a wide range of environments, making it a versatile and inclusive solution that meets the demands of various users and settings.

CHAPTER-2
LITERATURE SURVEY

2.1 Summary of papers studied

Research paper-1:

Renuka Annachhatre, Miti Tamakuwala, Prathamesh Shinde, Abhisha Jain, Prof. Pradnya V. Kulkarni, "Development of a Virtual Mouse Using Finger Tracking with Computer Vision," MIT World Peace University, Pune, India.

This paper discusses a virtual mouse system that uses finger tracking for hands-free computer control. The researchers used computer vision libraries, mainly OpenCV and MediaPipe, to detect hand gestures that allow users to move the cursor, click, and select items. They focused on handling challenges like background distractions and different lighting conditions, which can make gesture recognition less accurate. Through testing, they discovered that improving gesture detection in low-light settings and reducing interference from complex backgrounds could make the system more user-friendly. The study also suggests that this technology can make computers easier to use for people with physical limitations, enabling them to control computers more independently.

The researchers recommend further improvements in gesture accuracy and responsiveness to make such systems suitable for daily use. They highlight possible uses in hospitals, where touch-free controls can help with hygiene, and in classrooms, where alternative controls could support different learning needs.

Research paper-2:

Gayatri Jagnade, Mitesh Ikar, Nikita Chaudhari, Maithili Chaware, "Hand Gesture-based Virtual Mouse using OpenCV," Vishwakarma Institute of Technology, Pune, India.

This study develops a virtual mouse controlled by various hand gestures, not only for moving the cursor and clicking but also for scrolling and typing on a virtual keyboard. Using OpenCV and MediaPipe, the authors created fingertip detection for clicks, color tracking for scrolling, and eye and mouth tracking for extra functions. They added voice feedback for an interactive experience and to make it easier for people with disabilities to use. This feature is useful for reducing physical contact with devices, especially during the COVID-19 pandemic, for places like offices and libraries.

The research shows promising results in system accuracy, but it still faces challenges, especially with lighting changes and skin tone differences. The researchers suggest that using machine learning could help make gesture recognition more adaptable, making it less affected by environmental changes and suitable for real-world applications.

Research paper-3:

Tomasz Grzejszczak, M., et al., "Hand Landmarks Detection and Localization in Color Images," Multimedia Tools and Applications, 2016.

This paper looks into detecting hand landmarks in color images to support gesture-based systems. Instead of relying on depth sensors, it uses skin detection and distance transforms to recognize features like fingertips and joints in regular 2D images, making it a cost-effective approach. When compared to other methods, this technique showed high accuracy and speed, suggesting that it could be used for virtual mouse systems with regular webcams.

This method's main advantage is that it doesn't require depth sensors, making it affordable and accessible for places with limited budgets, like schools. This approach broadens the potential applications for virtual mice that can work with just basic webcam technology.

2.2 Integrated summary of the literature studied

The studies reviewed here show the development of virtual mouse systems controlled by hand gestures. These systems use advancements in computer vision, such as OpenCV and MediaPipe, to allow users to control a computer without touching a mouse, simply by using hand gestures. The papers demonstrate different approaches to gesture recognition, with most using fingertip and hand landmark detection for actions like cursor movement and clicking. Some studies even add extra features like eye and mouth tracking, color-based tracking, and voice feedback, making the systems more interactive and accessible for all users.

These systems have many practical uses, especially in healthcare, accessibility, and touch-free control areas. For example, gesture-based virtual mouse can help users with disabilities by allowing them to operate computers hands-free, which can increase their independence. The COVID-19 pandemic also highlighted the importance of touch-free controls in places like hospitals and public areas, where reducing contact is essential for hygiene. Since most of these systems work with simple webcams and OpenCV, they are also affordable and can be used in various settings.

However, there are still challenges to address. Environmental factors like lighting, background, and skin tone differences can affect how well gestures are detected. Improving these aspects is crucial for making gesture recognition more reliable. Machine learning and adaptive algorithms may help by making the system more adaptable to changing environments, which would make it more suitable for everyday use.

In summary, hand-gesture-based virtual mouse systems are an exciting development in how people can interact with computers. The improved accuracy of computer vision technology, along with the availability of OpenCV and MediaPipe, suggests that these systems could become common in the future. Future research should focus on handling environmental challenges and improving gesture recognition accuracy to make these virtual mouse systems more practical and user-friendly. These systems have the potential to help individuals with disabilities, reduce physical contact, and provide a more accessible computing experience for many users.

CHAPTER-3
REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1 Solution Approach

The solution for this project is a multi module system that utilizes a camera to detect hand gestures, which are then processed and mapped to control the PC in real time.

<u>Overall Solution Design</u>

The overall design can be visualized as a pipeline that begins with capturing a video feed, processes the captured frames to identify gestures, and maps those gestures to actions on the PC. The following steps summarize the process:

1. Video Capture and Preprocessing:

The system captures live video frames through a webcam using OpenCV, which are then preprocessed to ensure optimal conditions for gesture recognition.

2. Hand Gesture Recognition:

MediaPipe, a library developed by Google, is used for hand landmark detection. This library identifies the key points on the hand, such as the tips of the fingers and joints, which are crucial for recognizing gestures.

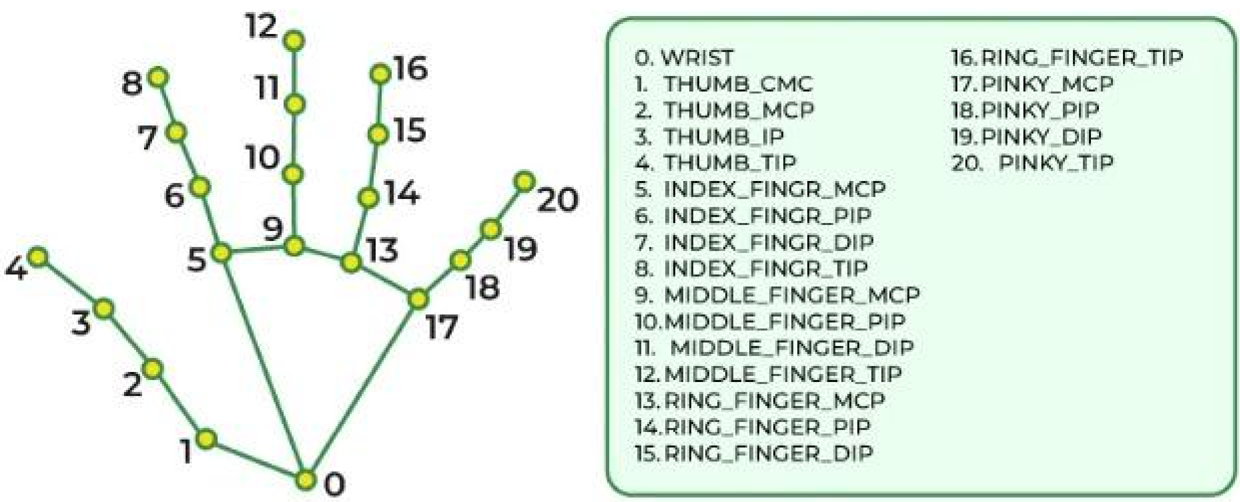

Fig 3.1 Hand Landmarks

3. Gesture-to-Action Mapping:

Once a gesture is identified, it's mapped to an action (e.g., mouse movement, click, scroll) using PyAutoGUI. This library allows the system to control the cursor, simulate clicks, and adjust system settings based on the detected gestures.

4. System Control and Feedback Mechanism:

A feedback mechanism is integrated to ensure real time responsiveness and adaptability, continuously monitoring for gestures and refining the system's response.

Module-Wise Solution Approach

1. Video Capture Module:

Library Used: OpenCV

Purpose: This module captures video frames from the user's webcam, creating a live video feed that forms the input for gesture recognition.

Key Functionalities:

- Frame Capture: Constantly captures frames from the video feed.
- Image Transformation: Mirrors the video feed so that hand movements appear more natural and intuitive.
- Resolution Adjustment: Ensures frames are of a suitable resolution for quick processing, balancing quality with speed.

Outcome:

Provides real time video input to the gesture recognition module, setting the stage for accurate and responsive interaction.

2. Hand Detection and Gesture Recognition Module:

Library Used: MediaPipe

Purpose:

This module is responsible for detecting the hand's presence and identifying specific landmarks on the hand, such as fingertips and joints, to recognize gestures accurately.

Key Functionalities:

- Hand Landmark Detection: MediaPipe identifies and tracks up to 21 key points on each hand, allowing detailed tracking of hand movements.
- Gesture Classification: Based on the relative positions of the landmarks, the system classifies gestures such as pointing, pinching, or forming a fist.

Challenges and Solutions:

- Lighting Conditions: Hand detection can be impacted by poor lighting. To counter this, the system uses dynamic thresholding to maintain accuracy under varying lighting conditions.
- Multiple Hands: The module distinguishes between left and right hands, assigning primary control to one hand while tracking both for more complex commands.

Outcome:

Delivers reliable gesture recognition by analyzing hand landmarks in real time, enabling accurate control over PC functions.

3. GesturetoAction Mapping Module:

Library Used: PyAutoGUI

Purpose:

This module translates recognized gestures into corresponding system actions, turning hand movements into tangible effects on the PC.

Key Functionalities:

- Cursor Control: Gestures such as V-gesture and moving the hand translate into cursor movement.
- Click Simulation: A pinching motion, detected by the relative proximity of the index finger and middle finger, is mapped to a mouse click.
- Scrolling: Vertical and horizontal scrolls are controlled by gestures where the hand moves up or down (or side to side).

Outcome:

The module provides a dynamic and robust interface for gesture based actions, translating hand movements into practical computer control functions.

4. System Workflow and Interaction Control:

Overview:

This module coordinates and manages the flow between video capture, gesture recognition, and action mapping.

Components:

- Primary Hand Detection: Identifies the user's dominant hand for primary control.
- Gesture Buffering: To prevent erratic behavior, this module buffers gesture recognition, ensuring that each gesture is consistently recognized over several frames before taking action.
- Adaptive Feedback: Continuously monitors the user's interaction to adapt the control response, providing smoother and more accurate adjustments over time.

Outcome:

Enhances the user experience by ensuring reliable and responsive interaction through efficient coordination of all system components.

CHAPTER-4
MODELING AND IMPLEMENTATION DETAILS

4.1 Implementation details and issues

The system implements several key gestures, each mapped to specific computer actions:

- Cursor Movement
  - Gesture: Raise both the index and middle fingers while keeping the rest of the fingers and thumb closed.
  - Action: Moves the on-screen cursor, with the cursor following the position of the midpoint of the extended index and middle fingertips.

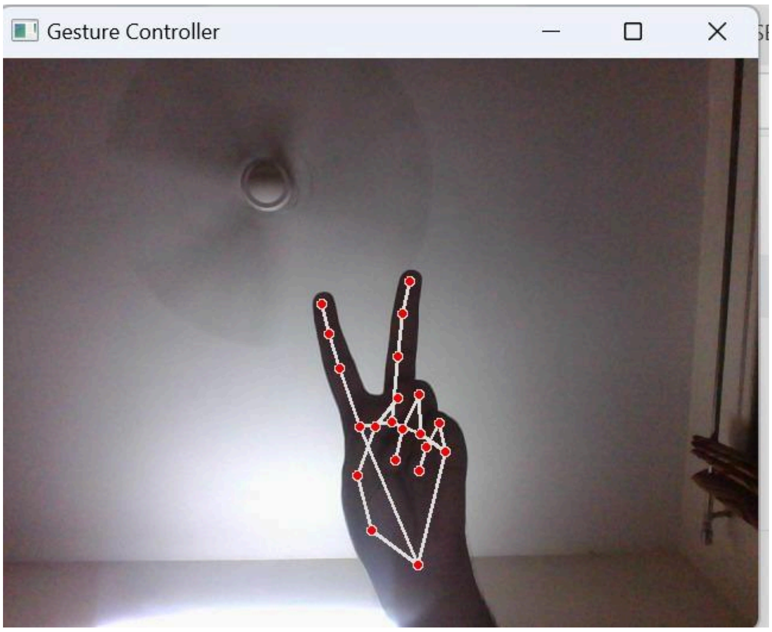

Fig 4.1 Cursor Movement

- Left Click
  - Gesture: Raise both the index and middle fingers, then click using the index finger.
  - Action : Simulates a left-click.

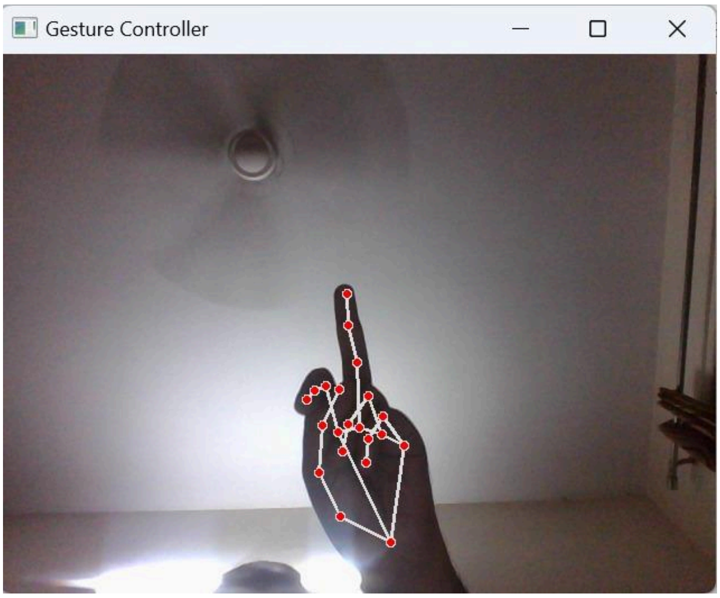

Fig 4.2 Left Click

- Right Click
    - Gesture: Raise both the index and middle fingers, then click using the middle finger.
    - Action: Simulates a right-click.

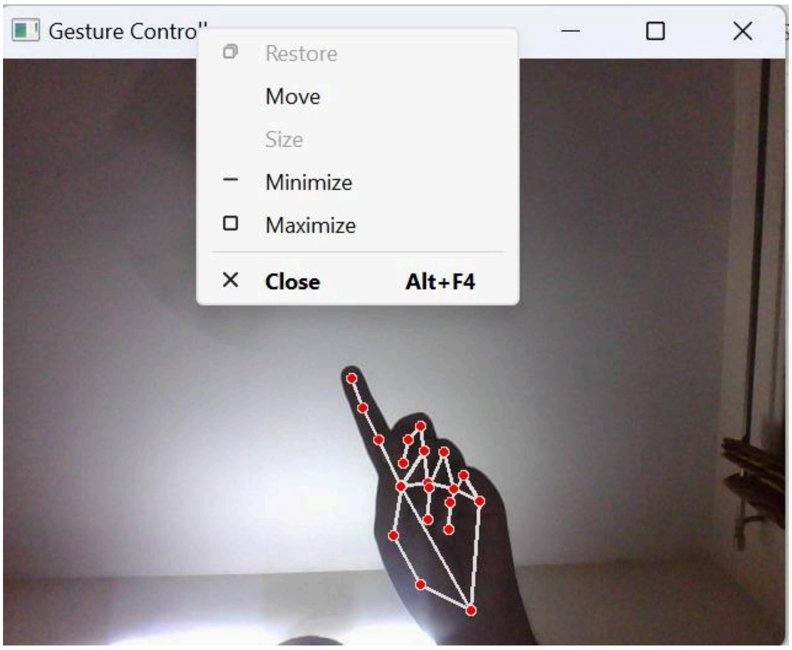

Fig 4.3 Right Click

- Double Click
    - Gesture: Double tap the index finger and middle finger together in rapid succession.
    - Action: Simulates a double-click, often used for opening files or applications.

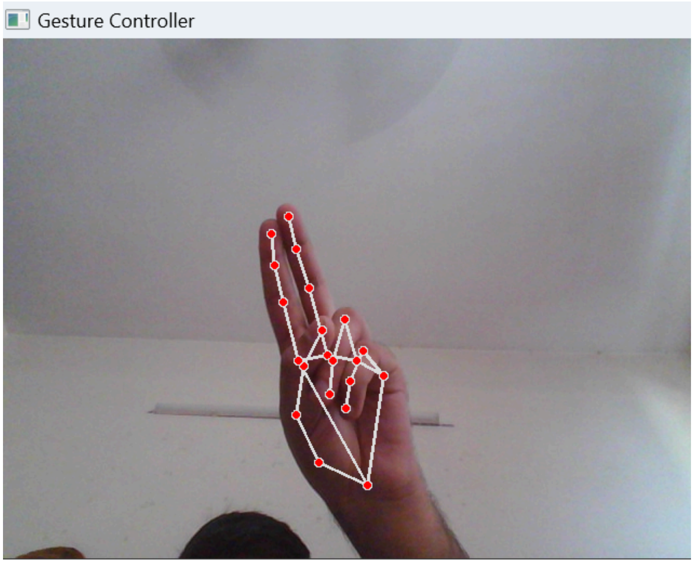

Fig 4.4 Double Click

- Vertical Scroll
    - Gesture: Pinch thumb and index finger and then move the hand up or down.
    - Action: Simulates vertical scrolling. The scrolling speed adjusts based on the speed of the hand's movement.

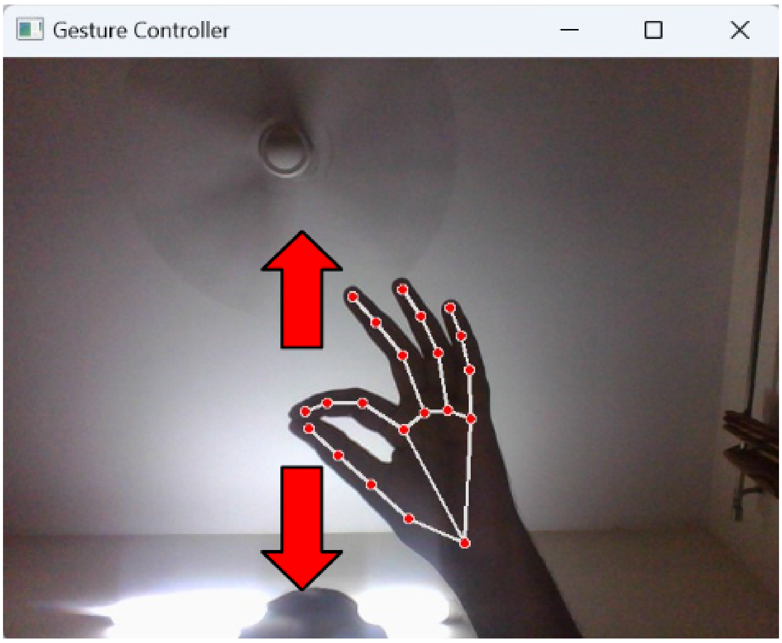

Fig 4.5 Vertical Scroll

- Horizontal Scroll
    - Gesture: Pinch thumb and index finger and then move the hand right or left.

○ Action: Simulates horizontal scrolling, which can be useful for navigating wide documents or web pages.

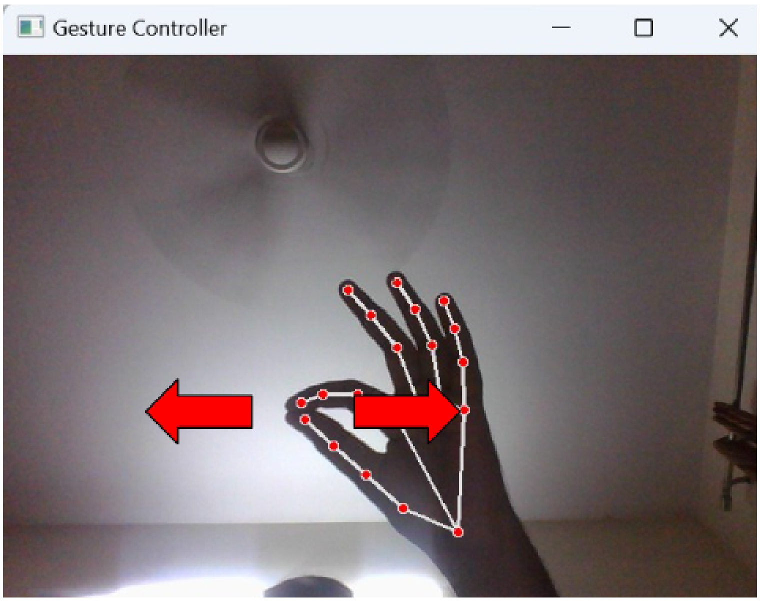

Fig 4.6 Horizontal Scroll

- Multiple Select

    ○ Gesture: Close all fingers into a fist, then move the hand to select multiple items.

    ○ Action: Enables multiple selections. Useful for selecting multiple items or lines of text.

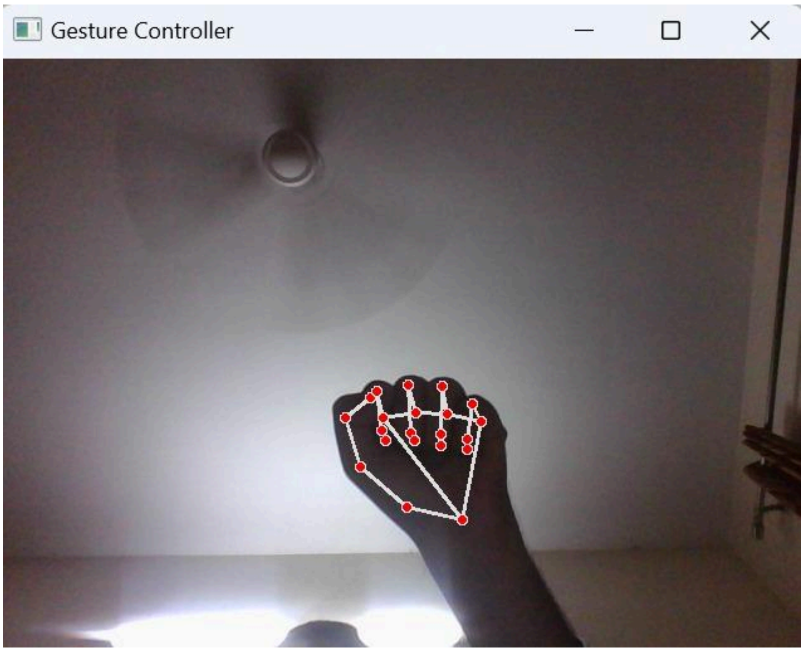

Fig 4.7 Multiple Select

- Drag and Drop
  - Gesture: Close all fingers into a fist and move the hand to drag the item, releasing the fist to drop it at the desired location.
  - Action: Simulates dragging an item across the screen and dropping it when the gesture is released.

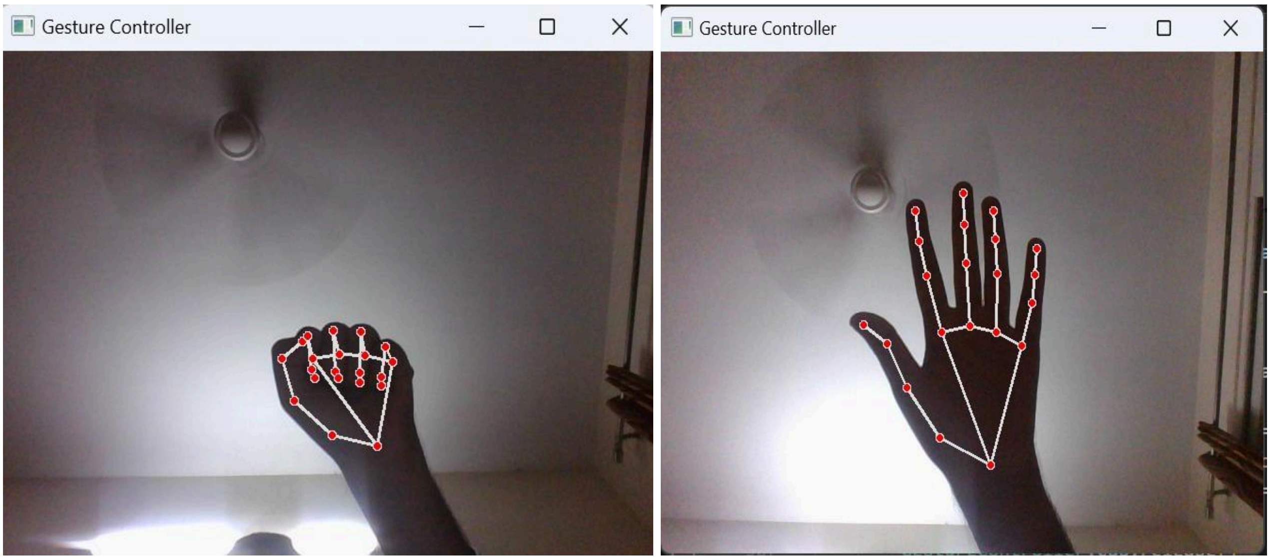

Fig 4.8 Drag and Drop

<u>Issues Faced :</u>

1. Lighting Conditions:

   Challenge: Hand detection accuracy is impacted by varying lighting conditions.

2. Real-Time Performance:

   Challenge: Ensuring the system responds in real-time without noticeable delay..

3. Multiple Hands Detection:

   Challenge: Differentiating between left and right hands and managing simultaneous gestures.

CONCLUSION

In Conclusion, the PC Controller with hand gestures project is an exciting new way for people to control their computer without touching a mouse or any other device. Instead, it uses hand gestures—like moving fingers in different ways—to perform actions like moving the cursor, clicking, scrolling, and dragging items on the screen. This project was made using two main tools: MediaPipe, which helps recognize hand gestures, and OpenCV, which processes images from the camera.

This technology can be very helpful in places where people prefer not to touch shared devices, like in hospitals, public kiosks, or even at home if you want a hands-free experience. It's also great for people who might have difficulty using traditional devices, as it just needs a regular camera, no special equipment, making it affordable and easy to use.

Unlike other systems that need expensive hardware like special controllers or gloves, this system only needs a standard webcam. It recognizes gestures quickly and works well in different lighting or background conditions, so it's reliable in many settings. Overall, the PC Controller with hand gestures is a creative, affordable, and easy-to-use solution for making computers more accessible, hygienic, and fun to use!

FUTURE SCOPE

- Advanced Gesture Library: Incorporate more complex gestures, including multi-finger and dual-hand gestures, for additional functionalities.
- Improved Accessibility Features: Combine with speech recognition to support individuals with severe disabilities.
- Incorporate voice feedback
- Commercial Applications:
    - Adapting the technology for gaming, where precise and natural gestures enhance the gaming experience.
    - Using gesture control in corporate presentations, education, and creative industries for interactive applications.

REFERENCES

[1] G. S. Mahitha, B. Revanth, G. Geetha, and R. Sirisha, "Hand Gesture Recognition to Implement Virtual Mouse Using Open Source Computer Vision Library: Python," in *Learning and Analytics in Intelligent Systems*, 2021, pp. 435–446, doi: https://doi.org/10.1007/978-981-15-9293-5_39.

[2] R. Sande, N. Marathe, N. Bhegade, A. Lugade, and S. S. Jogdand, "Virtual Mouse using Hand Gestures," *International Journal of Advanced Research in Science, Engineering & Technology*, vol. 8, no. 4, pp. 2350-0328, Apr. 2021.

[3] V. K. Sharma, V. Kumar, S. Tawara, and V. Jayaswal, "Virtual Mouse Control Using Hand Class Gesture," *GIS Science Journal*, vol. 7, 2020.