# GESTUREFLOW (VIRTUAL MOUSE)

**Submitted in partial fulfillment of the requirement for the Degree of Bachelors of Engineering in Computer Science & Engineering**

**Submitted To:**

**[PARUL UNIVERSITY, VADODARA, GUJARAT (INDIA)]**


**NIRANJAN PAL (210305124158)**
**ANKIT SRIVASTAVA (210305124125)**
**DIVYAAJEET PARMAAR (210305124141)**
**ASHISH KUMAR (210305124143)**

**Guide Name: AKASH PATIL**
**(ASST.PROF, PIT)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**PARUL INSTITUTE OF TECHNOLOGY VADODARA, GUJARAT**


**SESSION: 2023-2024**

# Parul University
# Parul Institute of Technology

**(Session: 2023 -2024)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

## <u>CERTIFICATE</u>

This is to certify that **Niranjan Pal, Ankit Srivastava, Divyajeet Parmar, Ashish Kumar,** Students of **CSE VI Semester** of **"Parul Institute of Technology, Vadodara"** has completed their **Minor Project** titled **"GestureFlow(virtual mouse)"**, as per the syllabus and has submitted a satisfactory report on this project as a partial fulfillment towards the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** under **Parul University, Vadodara, Gujarat (India)**.

**Akash Patil**
**(Project Guide)**
**(Asst. Prof )**
**(CSE / IT)**

**Prof. Sumitra Menaria**
**Head (CSE)**
**PIT, Vadodara**

**DR. Swapnil**
**Principal**
**PIT, Vadodara**

# DECLARATION

We the undersigned solemnly declare that the project report "**GESTUREFLOW(VIRTUAL MOUSE)"** is based on my own work carried out during the course of our study under the supervision of **AKASH PATEL.**

We assert the statements made and conclusions drawn are the outcomes of my own work. I further certify that

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree / diploma / certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.

Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

**NIRANJAN PAL (210305124158)** **SIGNATURE _____**

**ANKIT SRIVASTAVA (210305124125)** **SIGNATURE _____**

**DIVYAAJEET PARMAR (210305124141)** **SIGNATURE _____**

**ASHISH KUMAR (210305124143)** **SIGNATURE _____**

# ACKNOWLEDGEMENT

In this semester, we have completed our project on **“GESTUREFLOW (VIRTUAL MOUSE)”**. During this time, all the group members collaboratively worked on the project and learnt about the industry standards that how projects are being developed in IT Companies. We also understood the importance of teamwork while creating a project and got to learn the new technologies on which we are going to work in near future.

We gratefully acknowledge for the assistance, cooperation guidance and clarification provided by **“Akash Patel”** during the development of our project. We would also like to thank our Head of Department **Prof. Sumitra Menaria** and our Principal **Dr. Swapnil Parikh** Sir for giving us an opportunity to develop this project. Their continuous motivation and guidance helped us overcome the different obstacles for completing the Project.

We perceive this as an opportunity and a big milestone in our career development. We will strive to use gained skills and knowledge in our best possible way and we will work to improve them.

**NIRANJAN PAL (210305124158)** **SIGNATURE _____**

**ANKIT SRIVASTAVA (210305124125)** **SIGNATURE _____**

**DIVYAAJEET PARMAR (210305124141)** **SIGNATURE _____**

**ASHISH KUMAR (210305124143)** **SIGNATURE _____**

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| S. No. | Abbreviation | Full Form |
|---|---|---|
| 1 | SDLC | Software Development Life Cycle |
| 2 | HCI | Human Computer Interaction |
| 3 | DFD | Data Flow Diagram |
| 4 | VR | Virtual Reality |
| 5 | IDEs | Integrated Development Environments |

# Chapter 1

# Introduction

## 1.1 Overview

In The modern, technologically advanced environment is no longer suitable for using traditional mouse control approaches. Alternatively, people can operate their computers without a physical mouse or touchpad by using virtual mouse technologies. The purpose of this project is to develop an interactive virtual mouse control system that is based on dynamic hand gestures and that can give gesture control more accurately by utilizing more efficient working algorithms. Without a surface required, users can operate their devices while practicing yoga or working out. This is possible with virtual mouse technologies. The system shown in this study combines multiple technologies, such as computer vision, hand gesture detection, and Human-Computer Interaction (HCI), to achieve a high degree of accuracy and utility. Individuals with physical disabilities who find it difficult to get from one

## 1.2 Problem Statement

In today's digital era, the discovery and integration of essential web services like SOAP, UDDU, and WSDL into websites are pivotal for enhancing functionality and user experience. However, traditional methods of interaction, primarily reliant on physical mouse devices, pose challenges in terms of accessibility, usability, and hygiene. The need of the hour is to develop an innovative solution that addresses these limitations by introducing a Virtual Mouse utilizing hand gestures.

This aims to tackle the following challenges:

- **Accessibility**: Many individuals, including those with physical disabilities or discomfort using conventional mouse devices, face barriers in accessing and navigating through web services. A solution that enables interaction through intuitive hand gestures can significantly enhance accessibility.

- **Usability**: Traditional mouse devices may not always be convenient or practical, especially in scenarios where users are engaged in activities such as yoga or physical exercise. Developing a virtual mouse control system that allows users to interact with their devices without the need for a physical surface can enhance usability in various contexts.

- **Hygiene Concerns**: With increasing awareness of hygiene and the spread of infections, there is a growing need to minimize physical touchpoints, especially in shared computer environments. A virtual mouse control system based on hand gestures can mitigate these concerns by eliminating the necessity for direct physical contact with input devices.

- **Integration and Efficiency**: The proposed solution must seamlessly integrate with existing web service discovery processes while maintaining efficiency and accuracy. It should provide users with an intuitive and reliable means of navigating through a diverse range of web services.

## 1.3 Objective of Project

The objective of a virtual mouse system is to provide an alternative input method for controlling the computer cursor without the need for physical hardware such as a traditional mouse. This technology aims to enhance user experience by offering greater flexibility and accessibility, particularly for individuals with mobility impairments or those who prefer gesture-based interactions.

## 1.4 Application

Virtual mouse technology finds applications across various domains, catering to diverse needs and enhancing user experiences in numerous ways. Some key applications of virtual mouse systems are:

- **Accessibility**: Virtual mouse systems serve as assistive technologies for individuals with mobility impairments or physical disabilities, allowing them to control computers and interact with digital devices using hand gestures instead of traditional mice or keyboards.

- **Gaming:** Virtual mouse technology can be utilized in gaming environments to offer immersive and intuitive controls. It enables players to interact with games using natural hand movements, enhancing game play experiences in virtual reality (VR) environments or gesture-based gaming systems.

- **Human-Computer Interaction (HCI):** Virtual mouse technology plays a crucial role in HCI research and applications, enabling innovative interaction paradigms such as gesture-based interfaces, augmented reality (AR) interactions, and touchless interactions in smart environments.

- **Hygiene-Conscious Environments**: With growing concerns about hygiene and the spread of infections, particularly in shared computer environments, the system provides a hygienic alternative to physical mouse devices. By eliminating the need for direct physical contact, it helps reduce the risk of contamination and transmission of germs.

- **Virtual Reality (VR) and Augmented Reality (AR):** In VR and AR applications, virtual mouse technology enables users to interact with virtual objects and environments using hand gestures, enhancing immersion and enabling more natural interactions within virtual worlds.

- **Healthcare and Rehabilitation:** Virtual mouse systems are used in healthcare and rehabilitation settings to facilitate motor skill training and rehabilitation exercises. They provide patients with interactive and engaging tools for improving motor coordination and dexterity.

- **Industrial and Commercial Applications**: In industrial settings, the system can be used for hands-free control of machinery and equipment, improving worker safety and productivity. In commercial environments, it can enhance interactive displays and kiosks, allowing users to navigate through product catalogs or informational content using hand gestures.

- **Research and Development**: The development of the virtual mouse control system involves cutting-edge technologies such as computer vision, image processing, and machine learning. Researchers can explore new algorithms and techniques to improve gesture recognition accuracy and system performance.

# Chapter 2

## Literature Survey

A literature survey on virtual mouse technology encompasses:

1. **Historical Development of Virtual Mouse Technology:**
    - Explore the evolution of virtual mouse technology from its early conceptualizations to modern implementations.
    - Identify key milestones, breakthroughs, and influential research papers that have shaped the field.
2. **Gesture Recognition Techniques:**
    - Review different approaches to hand gesture recognition used in virtual mouse systems, including computer vision-based methods, machine learning algorithms, and sensor-based techniques.
    - Discuss the advantages, limitations, and performance characteristics of each approach.
3. **System Architectures and Components:**
    - Investigate the architectures and components of virtual mouse systems, including hardware sensors (e.g., cameras, depth sensors), software algorithms, and user interface design.
    - Analyze how these components interact to enable accurate and responsive cursor control.
4. **Applications in Human-Computer Interaction:**
    - Explore various applications of virtual mouse technology in human-computer interaction, such as accessibility tools for individuals with disabilities, gaming interfaces, presentation systems, and smart home controls.
    - Highlight case studies or real-world deployments of virtual mouse systems in different domains.
5. **Challenges and Limitations:**
    - Identify challenges and limitations associated with virtual mouse technology, such as accuracy issues in gesture recognition, sensitivity to environmental factors, and user fatigue.
    - Discuss ongoing research efforts to address these challenges and improve the performance of virtual mouse systems.
6. **User Experience and Usability Studies:**
    - Review studies that have evaluated the usability and user experience of virtual mouse systems, including user satisfaction, learning curves, and performance metrics.
    - Summarize findings related to user preferences, interface design principles, and recommendations for enhancing user interaction.
7. **Conclusion**
    - Summarize the key findings, trends, and insights from the literature survey.
    - Highlight gaps in the existing research and opportunities for future exploration in the field of virtual mouse technology.

# Chapter 3

# Research Methodology

## 3.1 Overview of Methodology

The methodology for developing a virtual mouse system typically involves several key steps, including hardware setup, software development, algorithm design, and evaluation. Here's an overview of the methodology:

**Hardware Setup**:

- Determine the hardware components needed for the virtual mouse system, such as cameras for capturing hand gestures.
- Set up the hardware configuration, ensuring proper placement and calibration for optimal performance.

**Software Development:**

- Develop software modules for capturing and processing input from the hardware sensors.
- Implement algorithms for hand detection and tracking to identify and locate the user's hand in the camera feed.
- Design algorithms for gesture recognition to interpret hand movements and translate them into cursor actions (e.g., movement, clicking, dragging).

**System Integration:**

- Integrate the hardware and software components to create a cohesive virtual mouse system.
- Ensure smooth communication between the sensors, gesture recognition algorithms, and cursor control mechanisms.
- Test the system for compatibility, stability, and performance across different hardware configurations and operating systems.

**Calibration and Optimization:**

- Calibrate the virtual mouse system to account for variations in lighting conditions, camera angles, and user preferences.
- Optimize the system parameters, such as gesture sensitivity and cursor speed, to enhance responsiveness and accuracy.
- Conduct iterative testing and refinement to fine-tune the system's performance and usability.

**User Testing and Evaluation:**

- Conduct user testing sessions to evaluate the effectiveness and usability of the virtual mouse system.
- Collect feedback from participants on their experience with the system, including ease of use, comfort, and accuracy.
- Use performance metrics such as gesture recognition accuracy, cursor responsiveness, and user satisfaction to assess the system's performance.

.

**Iterative Improvement:**

- Incorporate feedback from user testing and evaluation to make iterative improvements to the virtual mouse system.
- Address any issues or limitations identified during testing, such as false positives/negatives in gesture recognition or usability challenges.
- Continuously refine the system based on user feedback and emerging technologies to enhance its overall performance and user experience.

## 3.2 Project Platforms used in Project

Python serves as the primary programming language for the implementation of the virtual mouse system. Python is chosen for its simplicity, versatility, and extensive libraries for computer vision and machine learning.
Mediapipe:

**Mediapipe** is a framework developed by Google that provides tools for building perception pipelines. It offers pre-trained models and tools for various tasks, including hand tracking and gesture recognition. In this project, Mediapipe is utilized for identifying hand landmarks and tracking hand movements in real-time.

**OpenCV** is an open-source computer vision and machine learning software library. It provides a wide range of functionalities for image and video processing, including object detection, feature extraction, and motion tracking. OpenCV is employed in this project for real-time video processing, hand detection, and visualizations.
Autopy:

**Autopy** is a cross-platform Python library for automating mouse and keyboard inputs. It allows for programmatically controlling the mouse cursor's position and simulating mouse clicks and movements. Autopy is used in the virtual mouse system to move the cursor on the screen based on the detected hand gestures.

**Other Python Libraries**:

Various other Python libraries may be used for supporting functionalities, such as data processing, statistical analysis, and user interface development. These libraries may include **NumPy, Pandas, Matplotlib,** and among others.

**Development Environment:**

The project may be developed using integrated development environments (IDEs) such as PyCharm, **Visual Studio Code,** or **Jupyter Notebook**. These environments offer features like code debugging, syntax highlighting, and version control integration, facilitating the development process.

**Operating System:**

The virtual mouse system can be developed and deployed on different operating systems, including **Windows, macOS,** and **Linux**. Python and its associated libraries are cross-platform, ensuring compatibility across different environments.
By leveraging these platforms and tools, developers can effectively implement the virtual mouse system, integrating hand tracking, gesture recognition, and cursor control functionalities to create a seamless user experience.

## 3.3 Proposed Methodology

The gesture-based virtual mouse system uses the transformational approach to translate the coordinates of the fingertip from the camera screen to the full-screen computer window so that the mouse can be used. When the hands are identified and we ascertain which finger is capable of executing the designated mouse operation, a rectangular box is formed in connection to the computer window in the camera zone where we move the mouse pointer around the window.

### Modules Used:

- **OpenCV:**
  OpenCV is a computer vision package that includes methods for object detection in images. OpenCV is a Python computer vision toolkit that may be used to create real-time computer vision applications. Data analysis from pictures and videos, including object and face detection, is done with the OpenCV library. OpenCV is a free and open-source software library for machine learning and computer vision. Using OpenCV, a standard infrastructure for computer vision applications was developed to expedite the integration of AI into products.

- **Mediapipe:**
  A Google open-source framework called Media Pipe is used in a pipeline for machine learning. Architecture can be utilized for cross-platform development because it was developed using time series data. Because the Media Pipe design is multimodal, it supports a variety of audio and video formats. The developer uses the Media Pipe framework to build systems for application-related needs as well as to generate and analyze systems using graphs. The Media Pipe-using system's pipeline configuration is where the actions are executed. The pipeline's multiplatform execution flexibility enables scalability on desktops and mobile devices. The Media Pipe architecture consists of three

main parts: a mechanism for retrieving sensor data, a set of reusable parts called calculators, and performance evaluation. A pipeline is a graph composed of units called calculators that are joined by streams that allow packets of data to go through. Developers have the ability to add, remove, or redefine custom calculators anywhere in the graph to design their own applications.

- **Maths:**

  The Python math module is a vital component designed to handle mathematical operations. It comes with the standard Python version and has always been there. Most of the math module's mathematical functions are merely sparse wrappers for their equivalents on the C platform. Because the math module's basic functions are implemented in CPython, it operates efficiently and conforms to the C standard. You can carry out frequent and practical mathematical operations within your application by using the Python math module. There are several pre-set constants available in the Python math module. Having these constants available to you has several benefits. One benefit is that you may save a lot of time if you don't have to manually hardcode them into your software.

- **Pyautogui:**

  PyAutoGUI is essentially a Python program that can be used on Windows, Mac OS, and Linux that simulates mouse clicks and movements in addition to keyboard button presses. PyAutoGUI is a Python module enables individuals to automate cross-platform GUIs. The PyAutoGUI Python automation module can be used to move, scroll, click, and so on. You can use it to click exactly where you want to. utilized to automate mouse and keyboard control. In all three of the major operating systems (Windows, macOS, and Linux), there are multiple methods available for programmatically controlling the mouse and keyboard. This often has intricate, mysterious, and highly technological components. The purpose of PyAutoGUI is to hide all of this complexity behind a simple API.

## 3.4 Project Modules

The virtual mouse controller method employs **Python, Autopy,** and **Mediapipe** together with OpenCV to capture and analyse live video frames from a camera. The hand is identified and tracked using Mediapipe's hand tracking model. The hand landmarks are then used to identify the hand's position and motion.

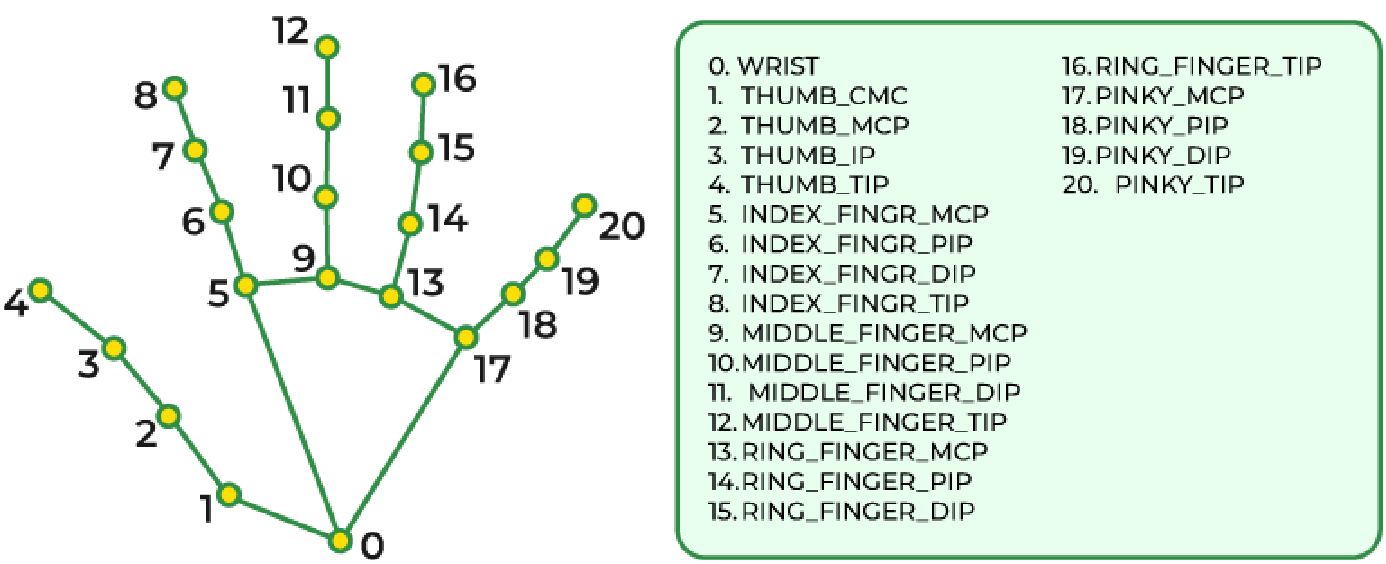
The mouse cursor on the screen is moved using Autopy in accordance with the user's hand gestures. To avoid unintentional cursor movements by hand tremors, a moving average filter is used to smooth the cursor's motion. Numerous technologies must be incorporated into the approach in order to provide hand gesture-based control of the mouse cursor on the screen.

The virtual mouse is implemented using Mediapipe, OpenCV, and Autopy software tools. Mediapipe provides pre-trained models for the purposes of recognising hand landmarks and tracking a user's hand location. Use the computer vision library OpenCV for real-time tracking and hand detection.

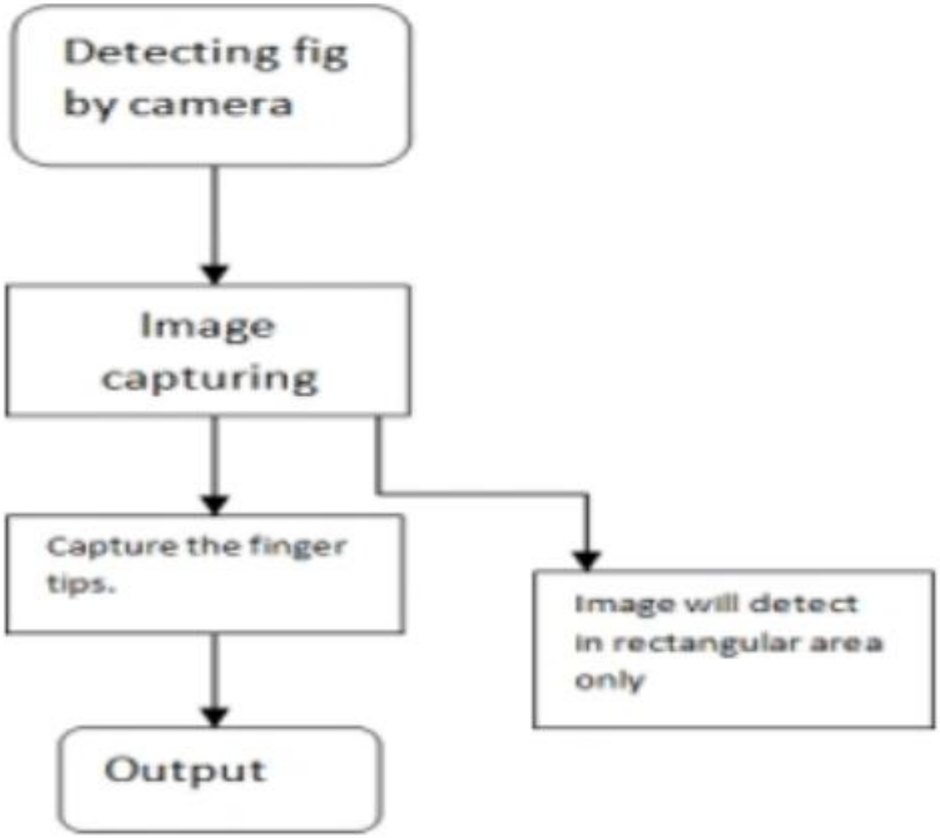
## 3.5 Diagrams

### 3.5.1 Block diagram of system

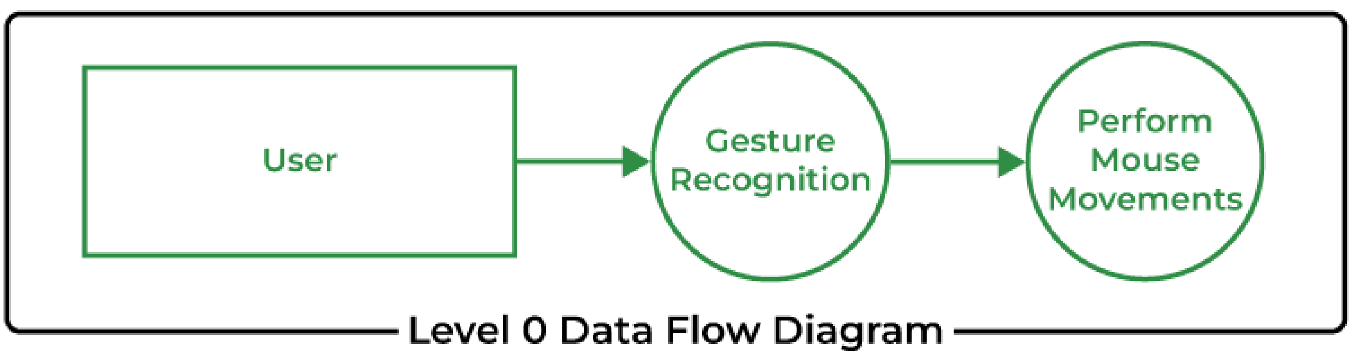

### 3.5.2 Co-ordinates or landmarks in the hand using Mediapipe



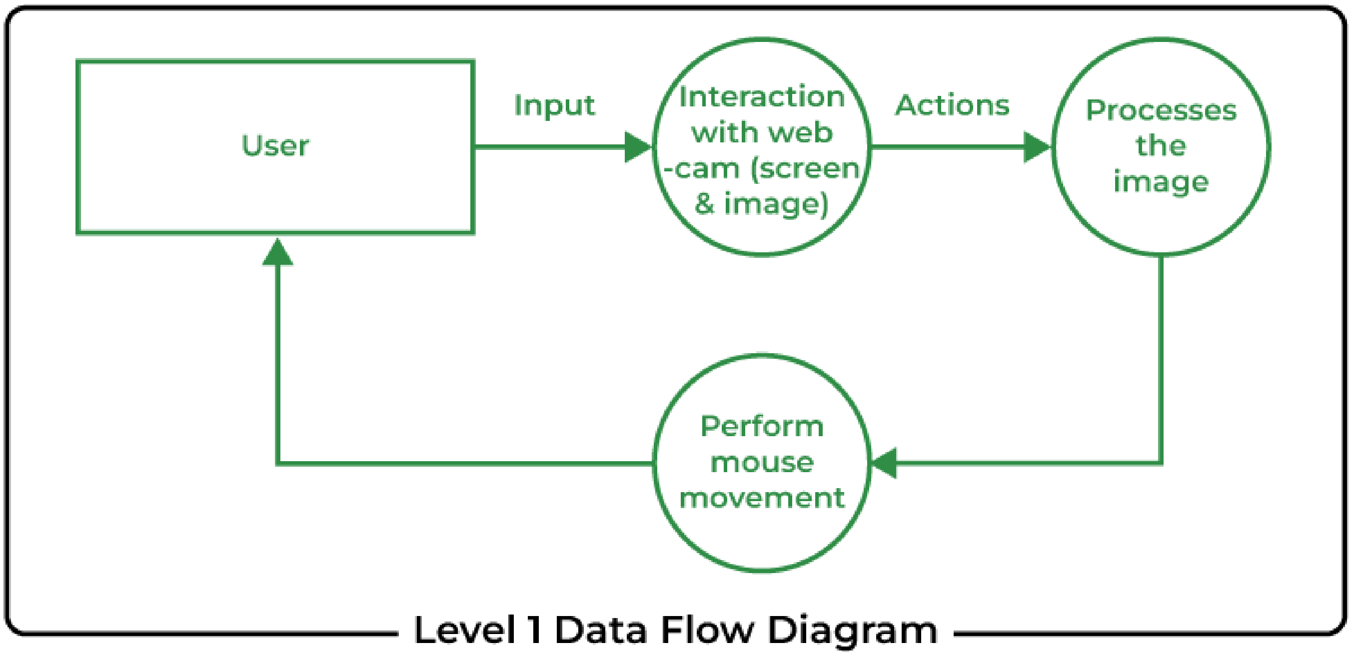
### 3.5.3 Mediapipe hand tip recognition layout



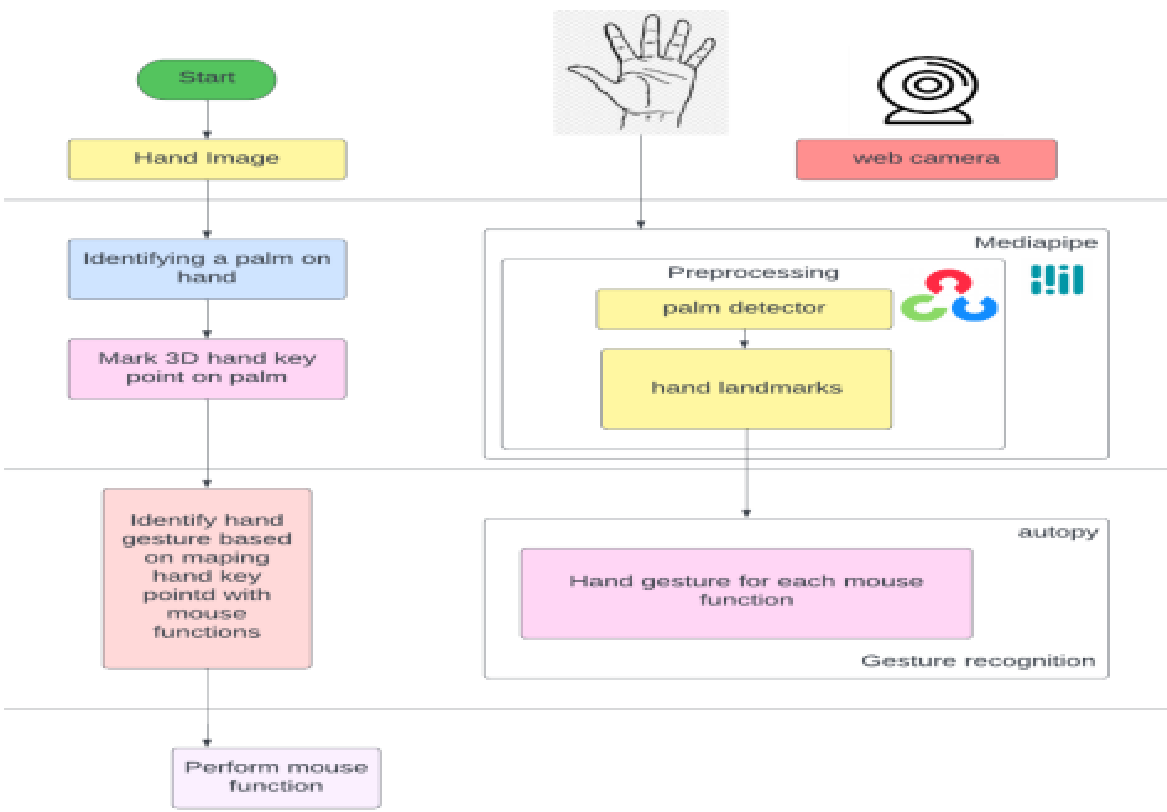

### 3.5.4 Level 0 DFD



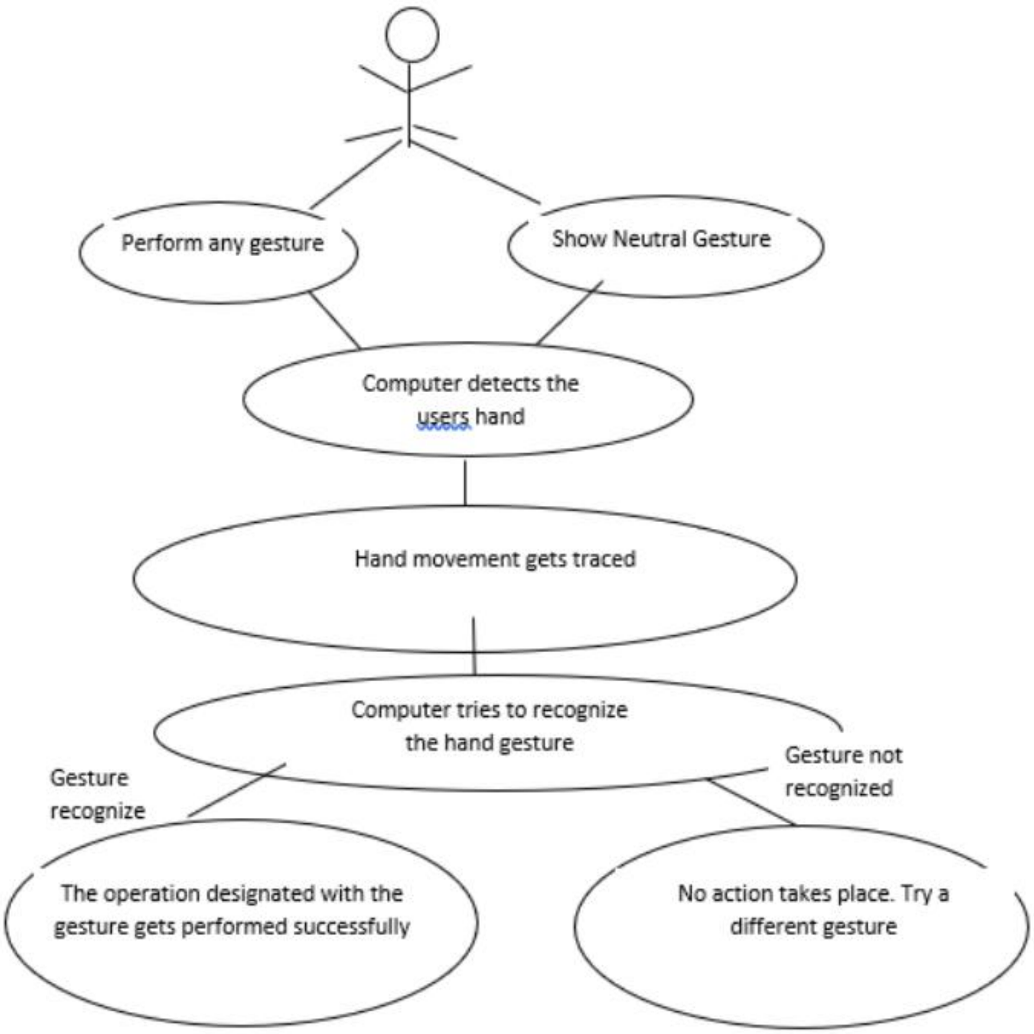
### 3.5.5 Level 1 DFD




### 3.5.6 Architecture Design



### 3.5.7 Use case Diagram



# Chapter 4

# System Requirements

## 4.1 Software Requirements:

- Operating System: Compatible with Windows, macOS, or Linux distributions.
- Python: Version 3.8.5 or Anaconda (allows you to create separate environments, each containing their own files, packages, and package dependencies).

**Python Libraries:**

- Mediapipe: Required for hand tracking and landmark detection.
- OpenCV: Necessary for real-time video processing and hand detection.
- Autopy: Used for controlling the mouse cursor based on hand gestures.
- NumPy: Often used for numerical computations and array manipulations.
- Other relevant libraries for data processing, visualization, and system integration (e.g., Pandas, Matplotlib).

**Development Environment:**

- Integrated Development Environment (IDE) such as PyCharm, Visual Studio Code, or Jupyter Notebook for code development and debugging.
- Version Control System (e.g., Git) for managing code revisions and collaboration.
- Package Management: Pip or Conda for installing and managing Python packages and dependencies.

**Additional Requirements:**

- Stable Internet connection for downloading necessary libraries, documentation, and resources.
- Access to relevant documentation, tutorials, and online resources for learning and troubleshooting.
- Permissions and credentials for accessing hardware components (e.g., camera) and system resources.
- Optional: Additional hardware peripherals for testing and validation (e.g., different types of cameras, input devices).

**Compatibility Considerations:**

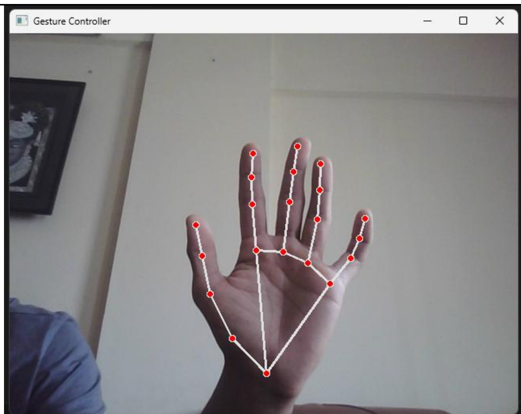
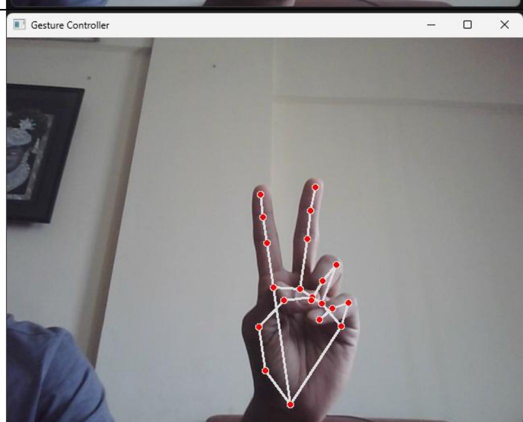
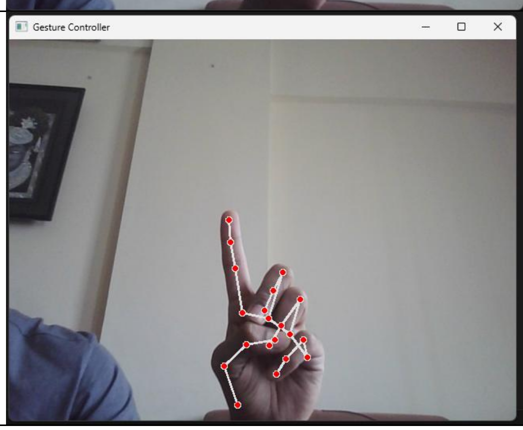
- Ensure compatibility of the virtual mouse system with different hardware configurations, operating systems, and Python versions.
- Test the system on a variety of devices and environments to identify and address compatibility issues.
- Provide clear instructions and guidance for users on setting up and configuring the virtual mouse system on their devices.

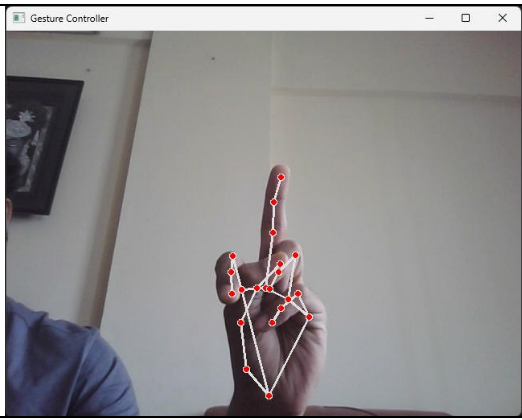
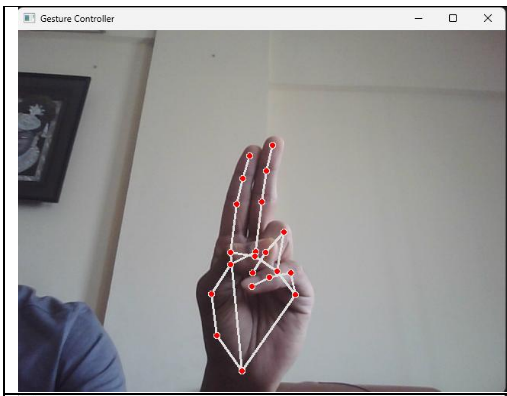
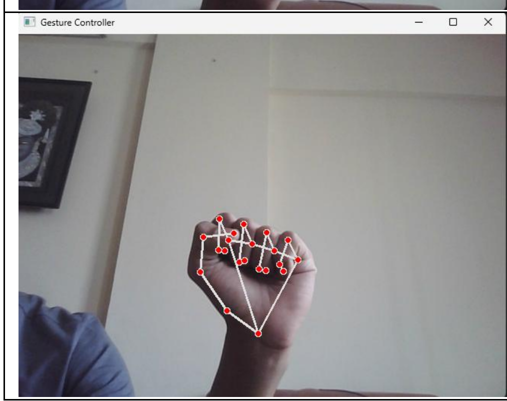
## 4.2 Hardware Requirements:

- Computer with a minimum of 2 GHz processor (multi-core recommended for better performance).
- Minimum of 4 GB RAM (8 GB or more recommended for smoother operation).
- Webcam or camera for capturing live video frames.
- Standard input devices such as a keyboard and mouse for system control during development.

# Chapter 5
# Expected Outcome

## 5.1 . IMPLEMENTATION

| Sample Input | Gesture Captured | Action Performed |
|---|---|---|
|  | **Palm** | **Neutral Gesture:**<br><br>Used for stopping the ongoing gesture. |
|  | **Peace Sign** | **Move Cursor:**<br><br>Using this gesture, the operator can navigate the cursor to the desired location. |
|  | **Index Finger** | **Left Click:**<br><br>Gesture to perform a single left-click |

| | | |
|---|---|---|
|  | **Middle finger** | **Right Click:**<br>Gesture to perform a single right-click. |
|  | **Two Fingertips (Index and middle)** | **Double Click:**<br>Gesture to perform a double-click. |
|  | **Fist** | **Multiple Item Selection Drag and Drop:**<br>It is used for selecting multiple items.<br>Gesture used for dragging and dropping files, icons, etc |

Parul® University NAAC A++

# Chapter 6

# CONCLUSION & FUTURE SCOPE

## 6.1 Conclusion

An effective and user-friendly technology that enables numerous mouse activities including clicking, dragging, and scrolling by simple hand movements is a virtual mouse operated by hand gestures.

Because it eliminates the need for a real mouse and allows for a hands-free experience, this technology is both convenient and accessible. People with mobility or disability may find it helpful. However, there are several restrictions that must be taken into account, such as the requirement for appropriate backdrop and lighting circumstances and the system's sensitivity to hand motion recognition, which can result in mistakes.

These restrictions imply that virtual mouse solutions might not be appropriate for all users and settings. However, with more study and development, computer vision techniques may be used to enhance the system's accuracy and resilience, making it a useful supplement to already-existing computer input devices. Thus, Virtual Mouse Control System is discovered through web services.

## 6.2 Future Scope

The virtual mouse we created has a very broad future potential. The incorporation of more sophisticated gesture detection methods might be one of the primary areas for improvement and might make it possible for even more accurate and effective mouse control.

Additionally, incorporating machine learning techniques may improve gesture identification precision and decrease false positives. Incorporating the usage of other input devices, such as voice or eye-tracking, might be another avenue for research to provide users even more accessibility and flexibility.

A virtual mouse system may offer certain advantages and breakthroughs, but it also has some drawbacks. The requirement for a camera or other sensing device to follow hand motions, which may only occasionally be practicable or accessible, is one of the key restrictions. Lighting circumstances, hand position, and movement might all have an impact on the system's accuracy and consistency.

Additionally, certain users, including those with physical limitations, may find it challenging to make particular movements. The virtual mouse system's overall usability and efficacy may be enhanced by addressing these drawbacks.

Now for the application part, many come to mind, some of them are listed below:

1. **Accessibility**: For those with impairments who are unable to operate a physical mouse, a virtual mouse may be a fantastic tool. They may use simple hand motions to operate a computer or other technological equipment.

2. In order to provide customers a more realistic and participatory **gaming experience**, virtual mice can be utilized in place of conventional gaming controllers.

3. Users can engage with the **virtual world** in virtual reality by making hand motions. In order to manage the user's interactions, a virtual mouse might be a helpful tool.

4. **Medical Industry**: By removing the need for direct physical touch with the computer or device, utilizing a virtual mouse in surgical procedures can assist maintain a clean atmosphere.

5. **Education**: A virtual mouse may be utilized in classrooms to improve hands-on learning activities like virtual laboratories and simulations.

6. **Design and Creative Industries**: To enable accurate command and control of digital art and graphics, virtual mice can be employed in the design and creative industries.

# Chapter 7

## REFERENCES


[1] Wu, H., Cai, X., Ma, J. and Zhang, X., 2022, September. Towards Natural Virtual Mouse with mm-Wave Radar. In 2022 19th European Radar Conference (EuRAD) (pp. 45-48), IEEE.

[2] Liang, Zifei, Choong H. Lee, Tanzil M. Arefin, Zijun Dong, Piotr Walczak, Song-Hai Shi, Florian Knoll, Yulin Ge, Leslie Ying, and Jiangyang Zhang. "Virtual mouse brain histology from multi-contrast MRI via deep learning." Elife 11 (2022): e72331.

[3] Joy Guha, Shreya Kumari, Prof. Shiv Kumar Verma, "AI Virtual Mouse Using Hand Gesture Recognition", Applied Science and Engineering Technology, Volume IJRASET41981, IJRASET, April 2022.

[4] S.Shriram, "Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread", Applied Sciences, Volume 2021, Article ID 8133076, 31 Jul 2021, Hindawi.

[5] J. Katona, "A review of human–computer interaction and virtual reality research fields in cognitive Info Communications," Applied Sciences, vol. 11, no. 6,2021.

[6] D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, "Real-time virtual mouse system using RGB-D images and fingertip detection," Multimedia Tools and Applications, vol. 80, no. 7, pp. 10473–10490, 2021.

[7] B. J. Boruah, A. K. Talukdar and K. K. Sarma, "Development of a Learning-aid tool using Hand Gesture Based Human Computer Interaction System," 2021 Advanced Communication Technologies and Signal Processing (ACTS), 2021, pp. 1-5.

[8] V. V. Reddy, T. Dhyanchand, G. V. Krishna and S. Maheshwaram, "Virtual Mouse Control Using Colored Finger Tips and Hand Gesture Recognition," 2020 IEEE-HYDCON, 2020, pp. 1-5.

[9] J. T. Camillo Lugaresi, "MediaPipe: A Framework for Building Perception Pipelines," 2019, https://arxiv.org/abs/1906.08172.

[10] Real-time Finger Tracking and Contour Detection for Gesture Recognition using OpenCV, Dr. Ruchi Manish Gurav; Dr. Premanand K. Kadbe [2015], International Conference on Industrial Instrumentation and Control (ICIC)

[11] S. U. Dudhane, "Cursor control system using hand gesture recognition," IJARCCE, vol. 2, no. 5, 2013.

[12] P. Nandhini, J. Jaya, and J. George, "Computer vision system for food quality evaluation—a review," in Proceedings of the 2013 International Conference on Current Trends in Engineering and Technology (ICCTET), pp. 85–87, Coimbatore, India, July 2013.

[13] A hidden Markov model-based dynamic hand gesture recognition system using OpenCV Published in 2013 3rd IEEE International Advance Computing Conference (IACC), Dr. Rajat Shrivastava.