International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 11 | November - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

A IMPLEMENTATION ON IMPLEMTATION OF VISION GLIDE TECHNIQUE FOR SMOOTH NAVIGATION WITH CAMERA BASED VIRTUAL MOUSE

Prof.S.S. Ganorkar[1]
Project Guide Dept.IT
KDKCE,Nagpur,India
s_ganorkar@kdkce.edu.in

Anagha D. Tembhurne[2]
Dept. Information Technology
KDKCE,Nagpur,India
anaghatembhurne516@gmail.com

Anjali P. Badole[3]
Dept.Information echnology
KDKCE,Nagpur,India
anjalibadole851@gmail.com

Dipti D. Dhore[4]
Dept. Information Technology
KDKCE,Nagpur,India
diptidhore14@gmail.com

Jayesh M. Tawade[5]
Dept.Information echnology
KDKCE,Nagpur,India
jayeshtawade3@gmail.com

Prajwal N. Neral[6]
Dept. Information Technology
KDKCE,Nagpur,India
prajwalnm11@gmail.com

Vitthal D Ghanwate[7]
Dept.Information Technology
KDKCE,Nagpur,India
vitthalghnwate24@gmail.com

ABSTRACT- As artificial intelligence technology has advanced, it has become commonplace to employ hand gesture detection to control virtual objects. The suggested system in this research is a hand gesture-controlled virtual mouse that uses AI algorithms to recognize hand gestures and transform them into mouse movements. People who have trouble using a conventional mouse or keyboard can use the system to provide an alternate interface. The suggested method takes pictures of the user's hand with a camera, which an AI program then utilizes to identify the motions the user is making. Since the development of computer technology, the method for constructing a process of human-computer interaction is advancing. The mouse is one of the best pieces of HCI (Human-Computer Interaction) technology ever created.

This article suggests an HCI-based virtual mouse system that makes use of hand movements and computer vision. webcam or built-in camera recordings of gestures that have been subjected to a color segmentation and detection procedure. This system uses a webcam or an integrated camera to capture frames. It then processes the frames to make them trackable, identifies various user motions, and executes mouse functions. In order to make the interaction more effective and dependable, this study suggests a vision-based system to control various mouse behaviors, such as left and right clicking, using hand gestures. This system uses a webcam or an integrated camera to capture frames. It then processes the frames to make them trackable, identifies various user motions, and executes mouse functions. Therefore, the suggested mouse solution removes the need for a device to use a mouse. Therefore, it can be seen that the development of HCI technology is advantageous.

Keywords- HCI (Human Computer Interaction), Hand Gesture, Gesture Recognition, OpenCV, Media-pipe, pyAutoGUI.

I. INTRODUCTION

There are many aspects of daily living that are influenced by technology. There are so many technologies available now, and computer technologies are advancing concurrently over the globe.. In an interactive setting, gestures are one of the most crucial forms of computer communication. There are so many technologies available now, and computer technologies are advancing concurrently over the globe. They are used to carry out a variety of duties that people are incapable of carrying out.

With the help of a webcam or built-in camera and a colored hat or piece of colored sticky note paper, it is quite easy to capture and track the fingertip of a hand using this gesture recognition system. The system will then monitor the color and movement of the hand and move the cursor along with it. In this research, a vision-based method for detecting hand motions and carrying out various mouse-like actions, like left and right clicking, is provided. Folding the first and middle fingers of the hand creates the baby-like mouse clicks on the left and right, respectively.

The AI virtual mouse framework is powered by the Python programming language. In addition, Open CV, a library for mobile PC vision, is used at various points in the AI virtual mouse framework. Many developers have made a variety of attempts to create models that can recognize human gestures. Artificial intelligence is one of the many developing technologies that is having a significant impact on every industry. Artificial intelligence makes life for humans easy and quick.

We are using the most recent artificial intelligence algorithms and tools to address the issues with the current methodologies. Artificial intelligence-based hand gesture control of a virtual mouse enables users to operate their computer mouse using hand motions without the usage of a real mouse. This technology tracks the user's hand motions and uses a camera vision-based method to carry out mouse functions on the computer screen. The centroid is also moved by hand motion, making this the fundamental sensor for changing the cursor on a computer screen.

II.PROBLEM DISCRIPTION & OVERVIEW

To track fingertips as a movable object, and to utilize it for mouse functions, the camera should be positioned in a way so that it can see the user's hands in the right positions. This can be used in space-saving situations, for those patients who don't have control over their limbs and for other similar cases. It's a virtual mouse instead of a physical mouse which will work only based on webcam captured frames & tracking colored fingertips.

III. DESIGN AND IMPLEMENTATION

Design

Designing a virtual mouse application that uses hand gestures to control the mouse typically involves multiple components: capturing hand gestures, processing them, and emulating mouse movements and clicks. Here's a high-level outline of how you can approach the development of such an application.

Creating a hand-gesture-controlled virtual mouse application is a multi-faceted endeavor encompassing two pivotal phases: design and implementation. The design phase constitutes the initial blueprint, where the architectural framework is mapped out, user interaction methods are defined, calibration processes are planned, and considerations for accessibility and robustness are addressed. On the other hand, the implementation phase involves the hands-on creation of the application based on the design specifications. This includes developing code to capture hand gestures, recognize predefined gestures, emulate mouse actions, and create the user interface. Each phase is critical, with design setting the direction and implementation realizing the envisioned concept.

Basic Components

Technology Stack Selection: In this stage, the choice of a programming language (e.g., Python, C++) and computer vision library (like OpenCV) is made, considering their compatibility and suitability for gesture recognition.

User Interaction Design: This phase involves designing the user interface elements, specifying the hand gestures to be recognized, and deciding on their corresponding actions, creating a user-friendly and intuitive interaction.

Calibration Process: Planning the calibration process includes defining how users will adjust settings based on their hand's size, positioning, and sensitivity preferences, ensuring accurate gesture recognition.

Create an Interface with tkinter Package: In this crucial step, you define the graphical user interface (GUI) using the tkinter package, crafting a visual layout with buttons, labels, and calibration options. The interface provides the means for users to interact with the application and configure settings.

Overall, the design phase sets the foundation for a user-friendly, accessible, and robust hand-gesture-controlled virtual mouse application, where the user interface, technology stack, calibration, and accessibility considerations work in harmony to provide an enhanced user experience.

Implementation

Capture Hand Gestures with MediaPipe and OpenCV

To capture hand gestures and enable real-time video feed processing, we rely on advanced computer vision techniques using the MediaPipe and OpenCV libraries. MediaPipe provides a robust framework for hand tracking, which complements OpenCV's capabilities for image processing.

The heart of this phase is creating algorithms that can accurately identify the user's hand, discern it from the background, and track its movements. MediaPipe's hand tracking module facilitates the detection of multiple hand landmarks, enhancing the precision and granularity of gesture recognition.

Gesture Recognition

At the core of the application lies the hand gesture recognition system. The algorithms devised in the design phase are implemented here to transform detected hand movements into meaningful gestures. These gestures are then mapped to specific actions, such as moving the mouse cursor, simulating clicks, or even scrolling.

MediaPipe's hand landmarks data, combined with OpenCV's image analysis, enables precise tracking and recognition of gestures. This information provides a wealth of data points that can be leveraged for accurate recognition of user intent.

Emulate Mouse Movements and Clicks

Once hand gestures are recognized, the application must emulate corresponding mouse movements and clicks. This functionality necessitates control over the mouse pointer's position and actions, such as left-clicks, right-clicks, and dragging.

To achieve this, libraries like PyAutoGUI (for Python) are often employed. PyAutoGUI offers the capability to programmatically move the mouse pointer to a specific location on the screen, simulate mouse clicks, and even drag objects. Through the seamless integration of gesture recognition and mouse emulation, users can control their virtual mouse cursor with their hand's movements and gestures.

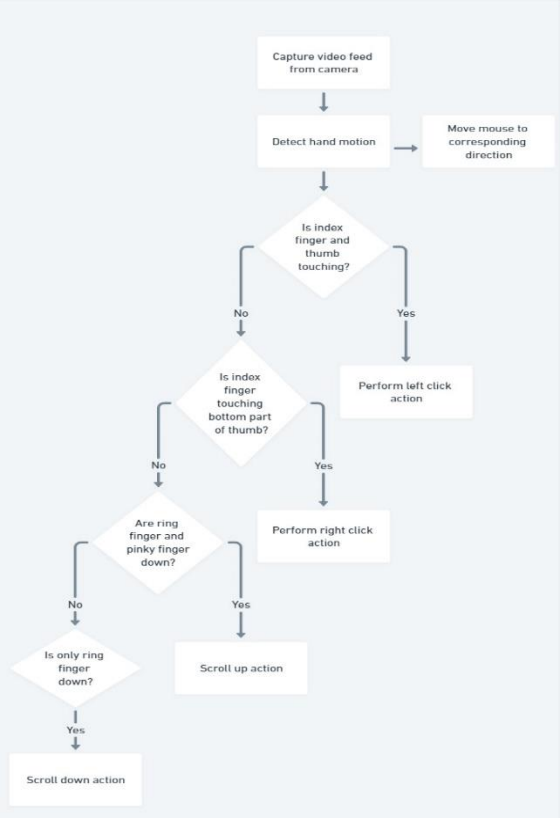
User Interface Implementation

The user interface, meticulously designed in the previous phase, now takes form through code. With the tkinter package in Python, buttons, labels, and calibration options are created, allowing users to start and stop the application, configure settings, and calibrate the system to their specific hand size, position, and sensitivity preferences.

The user interface plays a pivotal role in the application's overall usability. It's the bridge through which users interact with the application, and its design is translated into functional components that provide a smooth and intuitive experience.

Testing

Thorough testing is an integral part of the implementation phase. The application is subjected to various scenarios to ensure its reliability and responsiveness. This includes testing under different lighting conditions, hand sizes, and user interaction speeds. By simulating real-world usage, any issues, such as recognition inaccuracies or performance bottlenecks, can be identified and addressed.



In the development of a hand-gesture-controlled virtual mouse application, two integral modules, namely vmouse and controller, collaborate seamlessly to provide a rich user experience and precise control. These modules play distinct but complementary roles in enabling users to interact with their computers using hand gestures. Below, we delve into the functionalities and significance of each module in this innovative project.

THE FOLLOWING ARE THE DETAILS OF THE DIFFERENT MODULES IMPLEMENTED IN THE APPLICATION

vmouse Module

The vmouse module serves as the gateway to user interaction and the visual interface of the application. At its core, it accomplishes the following key tasks:

Graphical User Interface (GUI) Implementation: The vmouse module is responsible for creating and managing the graphical user interface (GUI) using the tkinter package. This interface serves as the bridge between the user and the application, offering a platform for user interaction. Through GUI elements such as buttons, labels, and calibration options, users can initiate and customize the application according to their preferences. The GUI design is a pivotal element, ensuring that users can comfortably and intuitively navigate the application.

Camera Access and Hand Tracking: To recognize and track the user's hand, the vmouse module accesses the computer's camera using the MediaPipe and OpenCV libraries. It captures a real-time video feed, which becomes the canvas for hand gesture recognition. By employing sophisticated image processing techniques, the module identifies the user's hand, distinguishes it from the background, and tracks its movements. This step is essential for providing real-time responsiveness to the user's hand gestures.

Gesture Recognition: Building upon the captured video feed and hand tracking, the vmouse module implements gesture recognition algorithms. These algorithms interpret hand movements and landmarks, recognizing predefined gestures that correspond to specific actions. For instance, the module can interpret hand motions as commands for moving the mouse cursor, simulating left-clicks, right-clicks, or even scrolling. The ability to accurately recognize these gestures ensures that users can control their computers effectively using their hands.

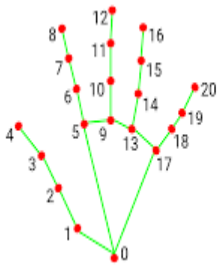
Call Controller Functions: Upon recognizing specific hand gestures, the vmouse module invokes functions from the controller module. These functions trigger actions such as left-clicks and right-clicks, enabling users to interact with applications, documents, and other elements on their computers seamlessly. The collaboration between vmouse and controller ensures that user gestures result in responsive and meaningful actions, enhancing the overall user experience.

controller Module

The controller module complements the vmouse module by providing the backend functionalities required for precise control of the virtual mouse. Here are the fundamental roles that the controller module fulfills:

Initialization: The controller module is responsible for initializing the system for mouse control. It sets up the necessary configurations and resources to monitor and respond to hand gestures detected by the vmouse module.

Mathematical Calculations: Utilizing Python's math package and mathematical calculations, the controller module determines the position of hand landmarks in the captured video feed. These calculations enable the module to accurately understand the hand's movements and translate them into precise mouse movements on the computer screen. The calculations are crucial for ensuring that the virtual mouse cursor follows the user's hand with precision.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Mouse Control: The heart of the controller module's functionality lies in its ability to control the mouse pointer's position and actions. It leverages the PyAutoGUI library to move the mouse pointer programmatically, simulate mouse clicks, and execute dragging actions. By doing so, the module translates the user's hand gestures, as recognized by the vmouse module, into tangible and responsive interactions with the computer's graphical interface.

The vmouse and controller modules work in concert to create a holistic hand-gesture-controlled virtual mouse application. The vmouse module focuses on user interaction, graphical representation, and gesture recognition, ensuring that users can seamlessly control their computers using hand movements. On the other hand, the controller module provides the technical underpinnings for precise mouse control, implementing mathematical calculations and mouse emulation. Together, these modules empower users to embrace a new dimension of computer interaction, where their hand gestures translate into efficient and intuitive control over their digital environment. The synergy between these modules is fundamental to the success of the application, offering an innovative and immersive user experience.

V. TECHNOLOGIES USED

Camera Hardware:
A high-quality camera or webcam is required for capturing real-time video. Consider a camera with high resolution and good low-light performance.
Computer:
A computer with sufficient processing power is essential for real-time video processing. This includes a capable CPU, GPU (optional but recommended for performance optimization), and an adequate amount of RAM.
Programming Language:

Python is a popular choice for implementing computer vision and user interface components due to its rich libraries and ease of use.
OpenCV:
A computer vision and ML software library called OpenCV is available for free download. Its objective is to aid programmers in the development of computer vision applications. Filtering, feature identification, object recognition, tracking, and other processing operations for images and videos are all available through OpenCV.

1. Loading and Preprocessing the Image/Video:
OpenCV can load images or videos from a variety of sources such as files, cameras, or network streams. Once the image or video is loaded, it can be preprocessed by applying filters or transforming the image to a different color space, such as converting a color image to grayscale.

2. Feature Detection and Description:
OpenCV can detect and extract features from an image or video, such as edges, corners, and blobs. These features can be used to identify objects or track their motion over time. OpenCV also provides algorithms for describing these features, which can be used to match them across multiple frames or images.

3. Object Detection and Recognition:
OpenCV can be used to detect and recognize objects in an image or video. This can be done using a variety of techniques, such as template matching, Haar cascades, or deep learning-based methods.

4. Tracking:
OpenCV can track objects in a video stream by estimating their position and motion over time. This can be done using a variety of algorithms, such as optical flow, mean-shift, or Kalman filtering.
5. Image and Video Output:
Finally, OpenCV can be used to display or save the processed images or videos. This can be done by showing the images in a window, writing the video frames to a file, or streaming the video over a network.
In general, OpenCV offers a large variety of tools and techniques for working with image and video data, making it a potent library for computer vision applications.

NumPy:
NumPy is a fundamental package for scientific computing with Python. It provides support for large, multi-dimensional arrays and matrices, along with an assortment of high-level mathematical functions to operate on these arrays. NumPy is often used as a foundational library for data manipulation, scientific computation, and machine learning in Python. Some of the key features and functions provided by NumPy include:

1. Arrays:

NumPy provides the `ndarray` data structure, which is a multi-dimensional array that can hold elements of the same data type. These arrays are more efficient and convenient for mathematical and numerical operations than Python's built-in lists.

2. Mathematical Functions:

NumPy includes a wide range of mathematical functions for performing operations on arrays, such as element-wise addition, subtraction, multiplication, division, and more. It also supports operations like matrix multiplication, transposition, and linear algebra functions.

3. Random Number Generation:

NumPy has a random module that allows for the generation of random numbers, which is useful for simulations and statistical analysis.

4. Indexing and Slicing:

NumPy supports advanced indexing and slicing techniques, making it easy to extract and manipulate specific elements of an array.

5. Performance:

NumPy is highly optimized for performance, making it suitable for large datasets and computationally intensive tasks.

6. Integration with Other Libraries:

NumPy is often used in conjunction with other libraries for data analysis and visualization, such as pandas, Matplotlib, and scikit-learn.

MediaPipe

Google created the open-source MediaPipe framework to enable the development of cross-platform, real-time computer vision applications. For processing and analyzing video and audio streams, it offers a number of pre-made tools and components, such as object detection, pose estimation, hand tracking, facial recognition, and more.

Developers can quickly construct intricate pipelines using MediaPipe that combine numerous algorithms and processes and execute in real-time on a variety of h/w platforms, like CPUs, GPUs, and specialized accelerators like Google's Edge TPU.

Additionally, the framework has interfaces helps us interacting with other well-liked machine learning libraries, including TensorFlow and PyTorch, and it supports several programming languages, like C++, Python, and Java. For computer vision and ML tasks, MediaPipe is a comprehensive library that offers a many of features.

Here are a few of the library's main attributes and features:

1. Video and Audio Processing:

MediaPipe provides tools for processing and analyzing video and audio streams in real-time. This includes functionalities such as video decoding, filtering, segmentation, and synchronization.

2. Facial Recognition:

MediaPipe can detect and track facial landmarks, including eyes, nose, mouth, and eyebrows, in real-time. This functionality is useful for applications such as facial recognition, emotion detection, and augmented reality.

3. Hand Tracking:

MediaPipe can track hand movements in real-time, allowing for hand gesture recognition and interaction with virtual objects.

4. Object Detection:

MediaPipe can detect and track objects in real-time using machine learning models. This functionality is useful for applications such as augmented reality, robotics, and surveillance.

5. Pose Estimation:

MediaPipe can estimate the poses of human bodies in real-time, allowing for applications such as fitness tracking, sports analysis, and augmented reality.

1) TKinter

Tkinter is a standard graphical user interface (GUI) library for Python that allows you to create windows, dialogs, buttons, labels, and other GUI elements for desktop applications. It is included with most Python installations, making it a convenient choice for developing cross-platform applications with a graphical interface.

Tkinter is based on the Tk GUI toolkit, which originated as part of the Tcl scripting language. Here are some key concepts and components of Tkinter:

Widgets:

Tkinter provides a variety of GUI widgets (controls) that you can use to build your applications, such as labels, buttons, text entry fields, checkbuttons, radio buttons, and more.

Geometry Managers: Tkinter supports several geometry managers to help you arrange and control the placement of widgets within your application window. The most commonly used managers are pack, grid, and place.

Event Handling:

You can bind functions to events that occur in your application, such as button clicks, keyboard input, and mouse actions. Tkinter provides mechanisms to handle these events.

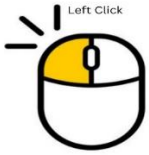
Top-level Windows:

You can create top-level windows for your application using the Tk() constructor. Each top-level window can have its own set of widgets and event handling.

Tkinter is a versatile library, and you can use it to build a wide range of desktop applications, from small utilities to more complex software. You can extend its functionality with third party libraries, and there are many tutorials and documentation resources available to help you get started with Tkinter GUI development in Python.

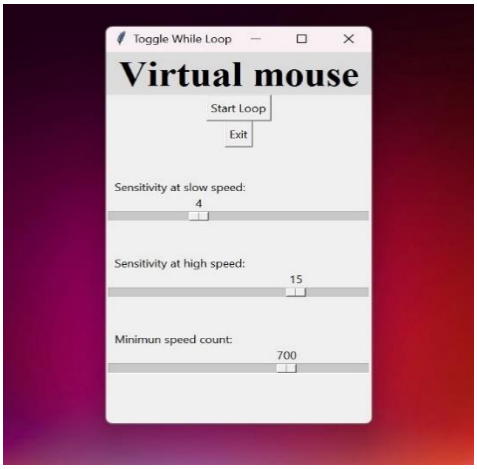
PyAutoGUI:
PyAutoGUI is a Python module that enables mouse and keyboard automation. It is used for controlling the virtual mouse and simulating mouse clicks and movements.
Depth Camera (Optional):
Depending on your project's requirements, you may consider using depth-sensing cameras like the Intel RealSense camera for 3D object tracking and more accurate positioning.
Machine Learning Frameworks (Optional): For advanced object recognition and gesture detection, you might utilize machine learning frameworks like TensorFlow or PyTorch.

VI. IMPLEMENTED GESTURES & OUTPUT







VII. FUTURE SCOPE

The future scope of work for the implementation of Vision on the Glide Technique with a camera-based virtual mouse offers exciting possibilities for further enhancing the system.
1. Advanced Gesture Recognition:
Explore cutting-edge gesture recognition technologies and machine learning algorithms to improve the accuracy and versatility of gesture-based interactions within Vision
2. AI Integration:

Integrate artificial intelligence (AI) features to automate tasks within Visio, such as optimizing diagram layouts and providing intelligent content suggestions based on user interactions.

3. Multi-Platform Compatibility:
Extend system compatibility to various platforms, including different operating systems, web-based Visio applications, and mobile devices, to reach a wider user base.

4. Collaborative Features:
Implement real-time collaborative features, allowing multiple users to work on Visio diagrams simultaneously and integrating with communication tools for discussions and project management.

5. 3D Visualization and AR:
Incorporate 3D visualization capabilities and augmented reality (AR) overlays, enabling users to work with 3D models and augmented content directly within Visio.

6. Voice and Natural Language Interaction:
Develop voice command and natural language processing capabilities, allowing users to control Vision and input data through spoken commands.

7. Enhanced Security and Privacy:
Continually improve security and privacy measures to protect user data and meet regulatory requirements.

8. Machine Learning Recommendations:
Implement machine learning algorithms to provide intelligent recommendations and suggestions to users as they work within Vision.

9. AR Headset Compatibility:
Consider compatibility with augmented reality headsets, enabling hands-free and immersive interaction with Visio diagrams.

10. Community and User-Generated Content:
Create a platform for users to share and access user-generated content, such as templates, stencils, and best practices, fostering a collaborative Visio ecosystem.

11. User Engagement Analytics:
Develop an analytics system to gather and analyze user engagement data, providing insights for system improvements and user-driven enhancements.

12. Continuous User Feedback Channels:
Maintain effective channels for user feedback and suggestions, enabling users to contribute to the system's development and direction.

These future scopes of work aim to push the boundaries of Visio's capabilities, making it more versatile, collaborative, and adaptable to emerging technologies. By focusing on these areas, the implementation of Visio on the Glide Technique with a camera-based virtual mouse can remain innovative and responsive to the evolving needs of users.

VIII. RESULT & CONCLUSION

The implementation of Visio on the Glide Technique with a camera-based virtual mouse is expected to yield a range of significant results. The system will offer users a seamless and highly intuitive navigation experience within the Visio environment. Users will be able to effortlessly interact with complex diagrams, charts, and visual data, thanks to the precise recognition of hand or controller gestures, which map directly to Visio's core navigation functions, including panning, zooming, and rotating.

This seamless interaction, paired with an enhanced user interface that adheres to the principles of user experience, will contribute to an improved overall user experience. The incorporation of visual and haptic feedback mechanisms will provide real-time confirmation of navigation actions, enhancing user confidence and understanding. Extensive usability testing and feedback collection, along with iterative refinements, will lead to a system that prioritizes user satisfaction and usability.

Accessibility features will ensure that the system is inclusive, catering to a broad range of users, including those with disabilities. Continuous performance optimization will minimize latency and maximize resource efficiency, guaranteeing that users experience a smooth and responsive navigation process. The system will also adhere to strict security and privacy measures to protect user data and meet regulatory requirements.

In conclusion, the implementation of Visio on the Glide Technique with a camera-based virtual mouse represents a transformative step toward improving the user experience within the Visio platform. Users can expect a powerful, accessible, and user-centric system that enhances their ability to work with Visio's diagramming and visualization tools. Ongoing maintenance and updates ensure that the system remains at the forefront of technological advancements, continually meeting user needs and adapting to emerging technologies. This project is poised to significantly enhance the way users interact with and navigate Visio diagrams, providing an immersive, intuitive, and efficient experience for a diverse range of users.

IX. REFERENCES:

[1] BALA HARSHITHAA B[1], LEELAVATHI R[2] "VIRTUAL MOUSE USING OPENCV" 2023 1Student, Department of Information Technology, Bannari Amman Institute of technology, Erode, Tamil Nadu, India [2023].

[2] CN Sujatha[1] , S.P.V Subbarao[2] , P.Preetham[3] , P.Surajvarma[4] , Y.Upendra[5] "VIRTUAL MOUSE USING HAND GESTURES" Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad 3,4,5 ECE, Sreenidhi Institute of Science and Technology, Ghatkesar,Hyderabad [2022].

[3] Thanrani, G.Gopikasri, R.,Hemapriya R., & Karthiga, M.(2022) Gym posture recognition and feedback generation using mediapipe and opencv.In international journal and advance research and innovative ideas in education (pp.2053-2057).

[4] Tharsanee, R.M., Soundariya, R.s., kumar, A.S., Karthing, M., & Sountharrajan, S.(2021). Deep convolutional neural network-based image classification for COVID-19 diagnosis.In Data Science for COVID-19 (pp. 117-145). Academic press.

[5] Roshnee Mtlani., Roshan Dadlani., Sharv Dumbre.,Shruti Mishra., & Abha Tewari. (2021). Virtual Mouse Hand Gesture. In international conference on technology advancements and innovations (pp.340-345).

[6] Neha Ramakrishnan[1] , Mrs. Jena Catherine Bel[2] "Virtual Mouse Using Hand Gesture" 2021 1 Student, Computer Science and Engineering, Velammal Engineering, college, Chennai, India 2Assistant Professor, Computer Science and Engineering, Velammal Engineering, college, Chennai, India [2021].

[7] Gubbala Durga Prasanth[1], P. Srinivasa Reddy[2] "Virtual Mouse Implementation using Open CV" SVKP & Dr K S Raju Arts & Science College, Penugonda, A.P, India [2020].

[8] Kabid Hassan Shibly[1], Samrat Kumar Dey[2], Md. Aminul Islam[3], Shahriar Iftekhar Showrav[4] "Design and Development of Hand Gesture Based Virtual Mouse" 2019 Dept. of Computer Science & Engineering Dhaka International University Dhaka, Bangladesh [2019].