International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Gesture Recognition Based Virtual Mouse and Keyboard

Snehal Shinde[1], Anurup Yemle[2], Shreepad Wadgave[3], Rohan Jadhav[4], Karan Anarase[5]

[1]Professor, Dept. of Information Technology, RMD Sinhgad School of Engineering ,Warje ,Pune, Maharashtra, India
[2,3,4,5]Student, Dept. of Information Technology, RMD Sinhgad School of Engineering, Warje, Pune, Maharashtra, India

Abstract
Computer vision has advanced significantly, allowing computers to identify their users through simple image processing programs. This technology is now widely applied in everyday life, including face recognition, color detection, autonomous vehicles, and more. In this project, computer vision is utilized to create an optical mouse and keyboard controlled by hand gestures. The computer's camera captures images of hand gestures, and based on their movements, the mouse cursor moves accordingly. Gestures can also simulate right and left clicks, while specific gestures enable keyboard functions, such as selecting alphabets or swiping left and right.
This system operates as a virtual mouse and keyboard, eliminating the need for wires or external devices. The project hardware requirement is limited to a webcam, with Python used for coding on the Anaconda platform. Convex hull defects are calculated, and an algorithm maps these defects to mouse and keyboard functions. By linking specific gestures with these functions, the computer interprets user gestures and executes corresponding actions.

Keywords: Computer Vision, Optical Mouse, Optical Keyboard, Hand Gestures, Python, Convex Hull Defects, Gesture Recognition, Virtual Input Devices

1. Introduction

The computer's webcam captures live video of the person sitting in front of it. A small green box appears in the center of the screen, where objects displayed within the box are processed by the program. If an object matches the predefined criteria, the green box changes to a red border, indicating that the object has been recognized. Once recognized, moving the object allows the mouse cursor to move accordingly. This feature enhances computer security and provides a virtual interaction experience.
Hand gestures are used in place of objects, with different gestures assigned to specific functions. For instance, one gesture controls the cursor movement, another performs a right-click, and a third triggers a left-click. Similarly, keyboard functions can also be performed using simple gestures, eliminating the need for a physical keyboard. If the gesture does not match any predefined patterns, the box remains green, while a recognized gesture changes the border to red.

2. Motivation

Gestures, commonly used in personal communication, play a significant role in enhancing interactions between humans and machines. They provide a natural and intuitive way to control devices, eliminating the need for physical interfaces. This opens up vast opportunities for developing unique methods of human-machine interaction, allowing for seamless integration of technology into daily life. By leveraging gestures, innovative applications can be created to enhance accessibility, efficiency, and user experience in various domains.

3. Problem Definition

Traditionally, computers and laptops rely on physical mice or touchpads, which were developed years ago. However, this project eliminates the need for external hardware by using human-computer interaction technology. It detects hand movements, gestures, and eye features to control mouse movements and trigger mouse events, offering a more innovative and intuitive approach

.

4. Functional Requirements

The system enables touchless computer control through hand gestures, supporting cursor movement, clicks, zoom, screenshots, and brightness adjustment. Built using Python (Anaconda/Spyder) with OpenCV and MediaPipe libraries, it processes real-time webcam input to detect hand landmarks and gestures. The architecture converts recognized gestures into OS-level commands through efficient input simulation, requiring only standard hardware. Development involved creating custom gesture datasets, implementing robust tracking algorithms, and mapping movements to system actions. Rigorous testing addressed challenges like lighting variations, occlusion, and false positives while optimizing latency. Performance depends on precise camera calibration and tuned parameters for reliable real-world operation. The modular design allows easy expansion of gesture vocabulary and supported commands. Special attention was given to ergonomic gestures that minimize user fatigue during extended use. Visual feedback confirms successful gesture recognition, improving usability. Security measures prevent spoofing attacks using recorded videos or images. The solution demonstrates practical computer vision applications for accessible human-computer interaction. Testing confirmed stable performance across different hardware configurations and environments. This approach offers an intuitive alternative to traditional input devices while maintaining system security. Future work could integrate multi-hand recognition and additional accessibility features. The project highlights Python's effectiveness for developing practical vision-based interfaces.

5. Activity Diagram

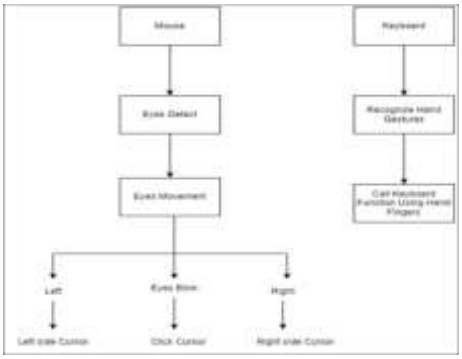

Fig -1: 1. Activity Diagram

Activity diagrams are visual representations that depict workflows through a sequence of activities and actions, incorporating choices, repetitions, and parallel processes. In Unified Modeling Language (UML), they are designed to represent both computational processes and organizational workflows, while also illustrating how data flows interact with the associated activities.
While activity diagrams mainly focus on illustrating the overall control flow, they can also incorporate elements that represent data movement between activities, often involving one or more data storage points.

6. System Architecture

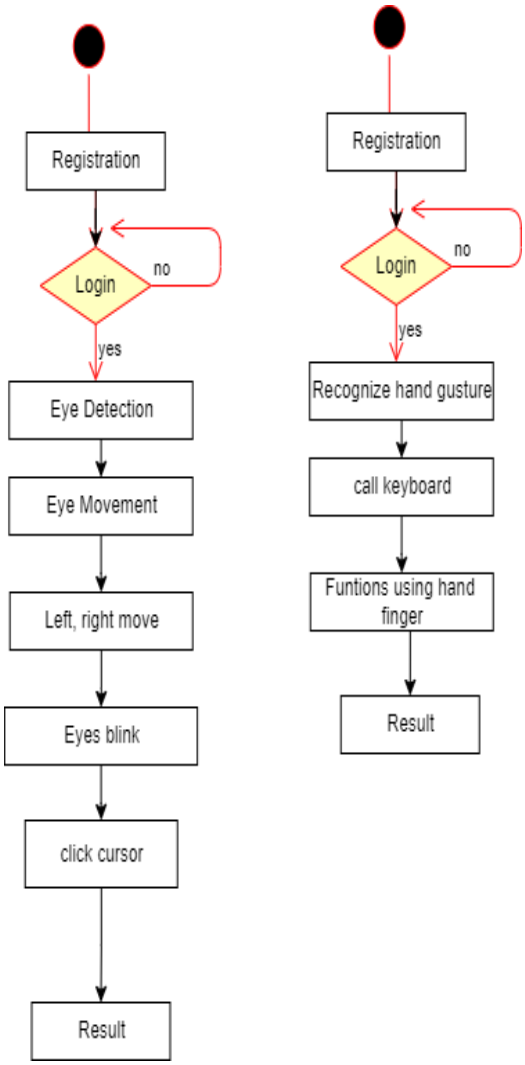

Fig -2: System Architecture

The architecture of this project integrates a webcam to capture real-time hand gestures for virtual mouse and keyboard functionality. The video frames undergo preprocessing using techniques like background subtraction, skin color detection, and contour analysis to isolate the hand. Gestures are recognized through convex hull defects and classified using algorithms like Haar cascade or CamShift. Recognized gestures are mapped to system actions, such as cursor movement or clicks, using libraries like PyAutoGUI. Visual feedback is provided with a green box turning red upon gesture recognition. This touchless system enhances user interaction, providing accessibility and security without external hardw-

are.

7. Advantages

Simple Implementation – Uses basic hand geometry for intuitive interaction
Cost-Effective – No expensive hardware required; works with a standard webcam Hardware-Free Solution – Eliminates the need for physical mouse/keyboard
Time-Saving – Enables quick and seamless computer control
High Productivity – Streamlines tasks like zoom, screenshots, and brightness control
Environmentally Resilient – Unaffected by dry weather (unlike fingerprint/scanners)
User-Friendly – Easy to learn and use for all age groups
Accessibility Support – Helps users with mobility challenges
Portable – Works on any system with a camera and Python setup
Future-Expandable – Can add more gestures and commands as needed

8. Applications

The system finds diverse applications across various sectors, enhancing efficiency and interaction. In colleges, it can be utilized for interactive learning and virtual lab setups, offering students a hands-on experience with innovative technology. In the government sector, it can streamline operations, improve accessibility, and provide secure, touchless interfaces for public services. Similarly, in the banking sector, it can enhance customer experience by enabling secure, gesture-based interactions, reducing the need for physical touch and ensuring hygiene. This versatile system proves beneficial in any domain requiring intuitive and contactless human-computer interaction.

9. Conclusion

The hand gesture recognition system presents a novel and efficient alternative to conventional mouse and keyboard interactions, enabling intuitive control through natural hand movements. By utilizing skin segmentation and arm-exclusion techniques, the system ensures accurate hand detection while minimizing background interference. Current capabilities include cursor movement, drag-and-click operations, and basic keyboard functions, with applications spanning 3D printing, architectural design, and remote medical procedures.

While the system performs reliably in static environments, future enhancements could expand its functionality to dynamic settings and integrate augmented reality (AR) for immersive 3D interactions. Implementing multidimensional cameras would further refine gesture precision, unlocking new possibilities in AR and real-world applications. These advancements promise to redefine human-computer interaction, making it more seamless, accessible, and adaptable to evolving technological demands.

This project lays a strong foundation for future innovations, bridging the gap between physical gestures and digital control systems. With continued development, gesture-based interfaces could revolutionize how users interact with technology across diverse fields.

References

1. S. S. Rao, "Sixth Sense Technology," in Proc. Int. Conf. Commun. Comput. Intell., 2010, pp. 336-339.

2. G. P. M. Game and A. R. Mahajan, "A gestural user interface to interact with computer system," Int. J. Sci. Technol., vol. 2, no. 1, pp. 018-027, Jan.-Mar. 2011.
3. Int. J. Latest Trends Eng. Technol., vol. 7, no. 4, pp. 055-062.
4. Imperial J. Interdiscip. Res., vol. 3, no. 4, 2017.
5. A. Christy, S. Vaithyasubramanian, V. A. Mary, and J. N. Renold, "Artificial intelligence based automatic decelerating vehicle control system to avoid misfortunes," Int. J. Adv. Trends Comput. Sci. Eng., vol. 8, no. 6, pp. 3129-3134, 2019.
6. G. M. Gandhi and Salvi, "Artificial Intelligence Integrated Blockchain for Training Autonomous Cars," in 2019 5th Int. Conf. Sci. Technol. Eng. Math. (ICONSTEM), Chennai, India, 2019, pp. 157-161.
7. "Online source," IOSR J. Comput. Eng., vol. 10. [Online]. Available: http://www.iosrjournals.org/iosrjce/papers/Vol10
8. S. S. Rao, "Sixth Sense Technology," in Proc. Int. Conf. Commun. Comput. Intell., 2010, pp. 336-339.
9. G. P. M. Game and A. R. Mahajan, "A gestural user interface to interact with computer system," Int. J. Sci. Technol., vol. 2, no. 1, pp. 018-027, Jan.-Mar. 2011.
10. Int. J. Latest Trends Eng. Technol., vol. 7, no. 4, pp. 055-062.
11. Imperial J. Interdiscip. Res., vol. 3, no. 4, 2017.
12. A. Christy et al., "Artificial intelligence based automatic decelerating vehicle control system to avoid misfortunes," Int. J. Adv. Trends Comput. Sci. Eng., vol. 8, no. 6, pp. 3129-3134, 2019.
13. G. M. Gandhi and Salvi, "Artificial Intelligence Integrated Blockchain for Training Autonomous Cars," in 2019 5th Int. Conf. Sci. Technol. Eng. Math. (ICONSTEM), Chennai, India, 2019, pp. 157-161.
14. A. Jesudoss and N. P. Subramaniam, "EAM: Architecting Efficient Authentication Model for Internet Security using Image-Based One Time Password Technique," Indian J. Sci. Technol., vol. 9, no. 7, pp. 1-6, Feb. 2016.
15. M. S. Roobini and M. Lakshmi, "Classification of Diabetes Mellitus using Soft Computing and Machine Learning Techniques," Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 6S4, 2019.