International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Gesturify: Hand Gesture Controlled Virtual Keyboard and Paint

Dr. Rajashekarappa[1], Varsha Rajput[2], Zhenkaar S.[3], Snehal M.[4], Qurrat Q.[5]
Department of Information Science and Engineering ,S D M College of Engineering and Technology, Dharwad, Karnataka

The shift toward touchless interfaces has accelerated with the demand for hygienic and intuitive computing. *Gesturify* is a gesture-based system that replaces physical input devices by enabling cursor actions and drawing functionalities through hand gestures. It integrates a virtual keyboard and a gesture-controlled paint application, powered by OpenCV, Media-Pipe, and Python's Tkinter for GUI. This solution offers a hands-free, accessible, and ergonomic interface ideal for virtual environments, physically challenged users, and applications where contactless interaction is essential.

Keywords: Hand Gesture Recognition, OpenCV, MediaPipe, Tkinter, Virtual Keyboard, Paint Tool, HCI

1. INTRODUCTION

Traditional input devices like keyboards and mice have long served as the primary means of interaction with computers. However, these devices present several challenges—especially in environments that demand hygienic, contactless, or inclusive solutions. Keyboards and mice are not designed with universal accessibility in mind, making them difficult to use for people with physical impairments. Moreover, in shared or public settings such as hospitals or kiosks, the use of shared input device scan pose hygiene risks due to the spread of germs through frequent physical contact.

In recent years, hand gesture recognition has emerged as a revolutionary approach to human-computer interaction (HCI), allowing users to control digital interfaces without the need for physical touch. Leveraging advancements in computer vision and machine learning, gesture-based systems enable real-time tracking of hand movements using standard webcams or depth sensors. This technology not only improves accessibility but also enhances the overall user experience by providing a more intuitive and immersive way of interacting with digital content.

Gesturify is a gesture-based system for typing and drawing using OpenCV, MediaPipe, and Tkinter. It includes a virtual keyboard and multi-mode Paint tool operated by hand gestures. The system is touchless, hygienic, and accessible for users with physical limitations. It demonstrates the potential of gesture control in enhancing user interaction.

2. LITERATURE SURVEY

Need for Research

As technology advances, the demand for more natural, inclusive, and hygienic ways of interacting with computers is growing rapidly. Traditional input methods pose several limitations, which has driven interest in gesture-based interfaces. The need for research in this area stems from:

1. **Accessibility Challenges:** Traditional input devices like keyboards and mice are not suitable for users with physical or motor impairments. Gesture-based systems provide an inclusive, hands-free alternative.

2. **Ergonomic Concerns**: Prolonged use of physical devices can lead to repetitive strain injuries and musculoskeletal discomfort. Gesture recognition offers a more natural interaction method, reducing physical strain.

3. **Hygiene Considerations:** Shared input devices in public or healthcare settings increase the risk of spreading infections. Touchless gesture-based systems improve hygiene by eliminating physical contact.
4. **Enhanced User Experience:** Gesture interfaces create more immersive and intuitive experiences, particularly in gaming, virtual reality, creative tools, and education.

Existing System

Most current systems for digital interaction rely heavily on physical input devices. Though effective, these devices have key drawbacks in modern use cases:

1. Limited Accessibility: People with physical impairments often find it hard to use keyboards and mice.

2. Ergonomic Risks: Continuous usage can cause fatigue, discomfort, or injury.

3. Poor Hygiene: Shared hardware can carry bacteria, especially in healthcare or public spaces.

4. Limited Interaction Modes: Physical devices do not support fluid, dynamic input like freehand drawing or mid-air typing.

Proposed System

The proposed **Gesturify** system addresses the limitations of traditional HCI methods through a gesture-controlled interface. Key features include:

1. **Gesture-Based Modules**: Integrates a virtual keyboard and a multi-mode paint tool, both operated using real-time hand gestures for typing, drawing, erasing, and switching modes.

2. **Computer Vision with MediaPipe & OpenCV**: Utilizes Python-based libraries to detect and track hand landmarks accurately, enabling intuitive and precise control without touch.

3. **Accessible and Touchless Interaction**: Enhances user accessibility and hygiene by allowing hands-free operation, beneficial in clinical, educational, and inclusive technology settings.

3. 4.UI Support with Tkinter: Offers a simple and interactive interface through Python's Tkinter library for easy mode switching

PROPOSEDMETHOD

Gesturify provides an integrated environment for typing and drawing through hand gestures using only a webcam. It includes:

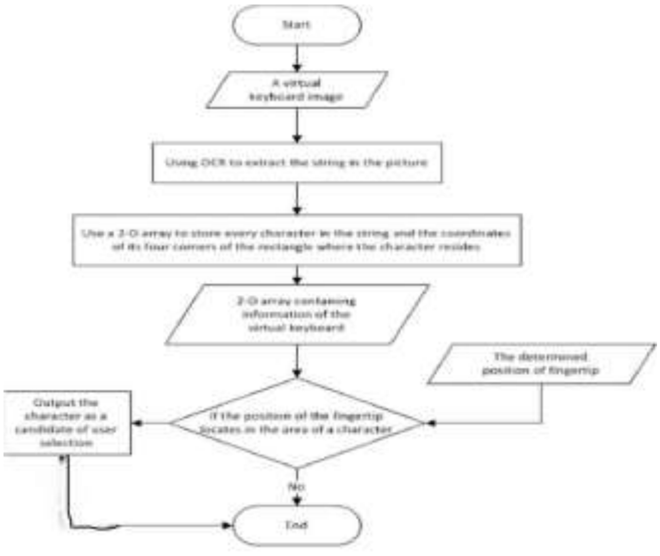


Fig1Flowchart

The following steps out line the methodology:

1. **Initialization:** The process begins at the "Start" node.

2. **Virtual Keyboard Generation:** A virtual keyboard image is created or displayed as the visual interface for user interaction.

3. **Text Recognition:** OCR (Optical Character Recognition) technology is used to extract all text characters present in the virtual keyboard image.

4. **Spatial Mapping:** The system creates a 2-D array data structure that stores each character from the virtual keyboard along with the precise coordinates of the four corners of the rectangle where each character resides**.**

5. **Data Structure Creation:** This results in a comprehensive 2-D array containing all information about the virtual keyboard layout, including character positions.

6. **Input Processing**: The system receives input about the determined position of the user's fingertip from external tracking.

7. **Position Analysis**: The system evaluates if the fingertip position overlaps with any character's defined area using the stored coordinate information.

8. **Character Selection**: If the fingertip is detected within a character's area, that character is output as a candidate for user selection.

9. **Process Completion**: If no characters selected or after character selection processing, the flow proceeds to the "End" node.

4. ALGORITHM USED FOR HAND DETECTION

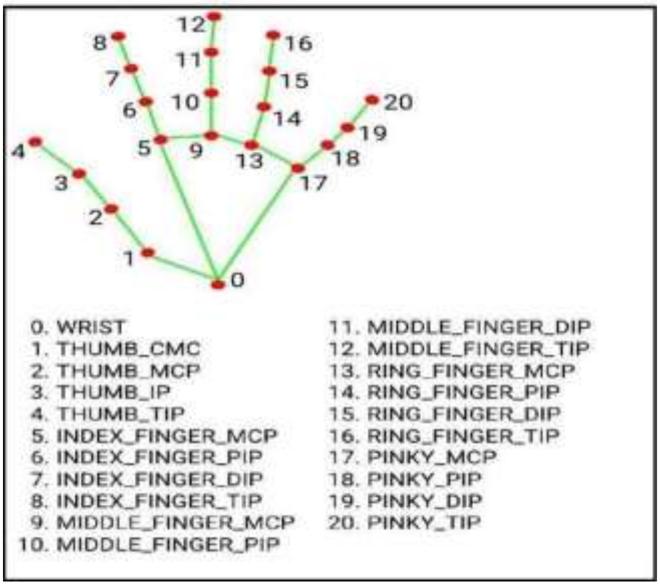
OPENCV

For hand and fingertip detection, the system employs Media-Pipe frame work and OpenCV, both of which are open-source libraries popular in computer vision tasks. MediaPipe is particularly effective in following hand gestures by employing machine learning algorithms that recognize finger positions and gestures

MEDIAPIPE

Media-Pipe is an open-source, flexible framework created by Google that enables developers to design and test machine learning pipelines for real-time computer vision applications. It enables the development of multiple models using a graph-based mechanism. Developers can build and interconnect processing elements such as graphs, nodes, streams, and calculators to create a pipeline specific to particular applications.

A MediaPipe pipeline typically involves several graphs that collaborate. For this project, the hand tracking module offered by Media-Pipe is utilized, which integrates palm detection and hand landmark estimation to effectively track hand movements. These modules run in real-time and are ideal for live input processing.



Some typical applications supported by Media Pipe are:

Selfie segmentation Face mesh detection Human pose estimationOpenCV "Open Source Computer Vision Library" is an open-source library in C++ and Python. It provides a large collection of over 2500 optimized algorithms for real-time image and video processing. The algorithms span a broad set of applications ranging from object detection to face recognition and gesture recognition.

In this project, OpenCV is utilized to process video frames and detect hands. The library accommodates various traditional and advanced computer vision algorithms. It facilitates video capture from webcams, image data processing, and embedding machine learning models. When combined with MediaPipe, OpenCV acts as the foundation for handling real-time image input and visualizing it.

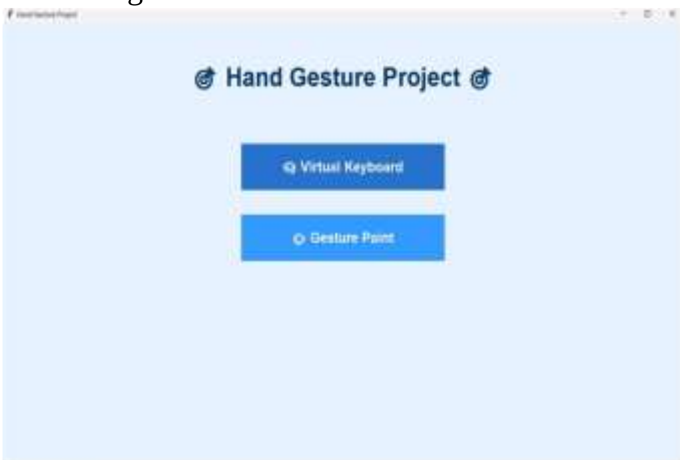


Fig3OpenCV

5. RESULTSANDDISCUSSIONS

The Virtual Keyboard Recognition system was tested in various environments to evaluate its performance. The results indicate the following key findings:

Accuracy: The system achieved high accuracy in character recognition and fingertip position detection using OCR and spatial mapping techniques. Proper lighting conditions were essential for optimal performance.

Responsiveness: The 2-D array implementation for character-coordinate mapping resulted in efficient processing and real-time character selection with minimal delay.

Accessibility: Users with mobility limitations found the touchless interface significantly more accessible than traditional keyboards. The virtual interface eliminated physical barriers to computer interaction.

Limitations: Performance decreased when fingertip tracking was compromised by environmental factors such as variable lighting or complex backgrounds. Character detection accuracy was affected by keyboard image quality.

Applications: The system showed promising results in contactless computing environments, educational settings, and public information kiosks where hygiene is a concern.

Img1:Frontpage

Img2:VirtualKeyboard

CONCLUSION

Gesturify successfully replaces conventional mouse and keyboard systems using a webcam-based gesture-controlled interface. With two primary modes—virtual typing and drawing— this project demonstrates a low-cost, hygienic, and inclusive solution for next-gen HCI systems. Future enhancements may include voice feedback, gesture training, or cloud integration.

REFERENCES

[1] Wang, Y., Liu, H., Ali, S., & Thompson, R., "Gesturify: A Framework for Virtual Keyboard Interaction Using Fingertip Detection and Character Recognition," IEEE Access, ISSN: 2169-3536,Volume11,pp.45302-45316, April 2023.

[2] Mehta, A., Patel, V., Rodrigues, J., &Kumar, N., "Computer Vision-Based Virtual Keyboard Using Spatial Mapping and Depth Sensing Technologies," International Journal of Human- Computer Interaction (IJHCI), ISSN: 1044-7318, Volume 39, Issue 7, May 2023.

[3] Zhang, L., Chen, R., & Williams, D., "Touchless Interaction Through Virtual Keyboard Recognition and Fingertip Tracking," IEEE Transactions on Human-Machine Systems, ISSN: 2168-2291, Vol-53, Issue-4, August 2023.

[4] Kim, J., Park, S., & Lee, H., "OCR-Enhanced Virtual Keyboard Systems for Touchless Computing Environments," Journal of Advanced Computer Vision and Applications, ISSN: 2167-7964, Volume 14, Issue 2, September 2023.