Smart Glasses for Hand Gesture Recognition for Screen Control of Digital Devices

Naquiya Juzar
Department of Computer Engineering
Vishwakarma Institute of Information Technology
Pune, India

Kaustubh Kale
Department of Computer Engineering
Vishwakarma Institute of Information Technology
Pune, India

Pravin Kumavat
Department of Computer Engineering
Vishwakarma Institute of Information Technology
Pune, India

Swapnali Kulkarni
Department of Computer Engineering
Vishwakarma Institute of Information Technology
Pune, India

Pranali Chavhan
Department of Computer Engineering
Vishwakarma Institute of Information Technology
Pune, India

***Abstract*—Hand gesture recognition has evolved from some basic deployments that depended on gloves and colored markers to low end vision-based configurations to the more recent high-end configurations such as Kinect and Leap Motion devices. Traditional systems depend on having controlled environment dependencies, suffer from a problem of lighting, and are not able to give continuous recognition capabilities. Using machine learning and depth-sensing technologies, some recent developments have managed to fabricate bulky hardware or inflexible setups that severely limit the portability and user experience.**

This paper proposes a novel solution: smart glasses with IR cameras and power of Mediapipe, OpenCV, PyAutoGui and machine learning algorithms for marker-free real-time hand gesture detection. Differently from previous systems which could not adapt to the dynamic nature of environments and involved external accessories, the proposed system is expected to achieve accurate gesture recognition in changing settings without requiring markers. For example, it supports proximity-based detection such that only gestures from inside a certain range belonging to the user will be captured while eliminating irrelevant signals. This development ensures seamless hands-free interaction with digital devices with applications in smart education, public kiosks, and touchless computing, overcoming some key limitations of previous systems.

***Index Terms*—Hand Gesture Recognition, Smart Glasses, IR camera, MediaPipe, OpenCV, Machine Learning, Real Time gesture recognition, Proximity-Based detection, Human-Computer interaction, Wireless screen control**

I. INTRODUCTION

Hands-free interaction and gesture recognition are fast becoming the new paradigms in **Human-Computer Interaction (HCI)**, opening up avenues for more intuitive and natural ways with which users would interact with any form of technology. In **smart devices**, gestures provide perfect control free from the need for human contact. Gesture control recognizes these gestures in virtual reality, giving the user a more immersive experience about engaging with virtual worlds. Gesture control further enhances accessibility and functionality with wearables while its applications stretch to industries such as education and healthcare. The above technologies are thus revolutionizing the way people interact with digital systems with their more fluid and intuitive interfaces across the sectors.

Gesture control enables people to communicate with the machine in a non-contact manner, mainly the body movements without physically touching the device, especially hands or eyes. Sensors or cameras utilized in swiping or pointing detect hand gestures. This is more flexible and accurate.

However, when it comes to eye gestures, the slight eye movements raise questions in terms of accuracy and precision because various commands are rather subtle, which thus may often make it hard to reliably distinguish between them and hence leads to quite many misinterpretations of inputs. This problem is compounded by the small range of motion in the gestures of the eyes, especially compared to the hand gestures, which are much more explicit and easily recognizable. In addition, frequent use of eye movements would quickly result in significant eye strain and therefore make the entire system less comfortable and even less functional to use for extended periods. This is further due to the fact that eye movements are limited in terms of the scope of commands applicable

Identify applicable funding agency here. If none, delete this.

as it provides relatively fewer and less diverse commands than is generally the case with hand gestures that are also to be condensed to the functionality of the system. Moreover, the systems are sensitive to environmental factors such as illumination, which skews the correctness and performance of the eye gesture recognition.[1][2][3][4].

To address these challenges, previous researches has made significant strides in exploring hand gesture recognition for human-computer interaction. It began with iMEMS technology, to develop the use of a 3-axis accelerometer in order to track movements of the hand and translate such movements into digital signals. Such data are, in turn, processed by an ATmega8L microcontroller, which digitizes the data before sending it wirelessly to a computer utilizing XBEE modules. Gesture recognition algorithms map specific hand movement with actions such as moving the cursor and manipulating 3D objects. When the data is processed and interpreted in terms of gestures, it comes out to be working on software written in Visual Basic, which then triggers some action in response to the gesture[5].

MediaPipe Hands is used as the basis for real-time tracking of hand skeletons on a single RGB camera. Authors improved keypoint detection efficiency by introducing new datasets and refined the rotation and scaling estimation of the hand. Two classifier approaches have been employed for gesture classification: a heuristics-based classifier based on predefined geometric relations among 3D keypoints of the hand and a much more accurate neural network-based classifier, trained on a large handshape data set with the features extracted from 3D keypoints. Recent advances make possible this system to recognize static gestures in an improved preciseness, running in real-time with high efficiency on mobile devices[6].

OpenCV was used for executing real-time computer vision tasks like hand detection and image processing. In the case of MediaPipe, a hand landmark detection is employed, whereby the system uses machine learning to identify and track 21 knuckles points in the hand. Finally, PyAutoGUI equips the system with the ability to control the GUI of a computer through automation functions like changing the cursor's position by hand movements, clicking or scrolling, among others. This system works based on capturing the hand movement through a webcam, and with background subtraction, skin detection, and edge detection methods, it analyzes in real time the hand gestures. This way, the system avoids any necessity of using external hardware since there is reliance on only the webcam used by the user in conjunction with these methods to ensure smooth virtual mouse control[7].

Using Mediapipe and DenseNet201 results in hand gesture recognition much more advanced compared to the earlier methods. Most of the earlier systems applied relatively simple methods which either were based on edge detection or colored markers sensitive to the lighting condition and calibrated. Mediapipe is designed for tracking a marker-free hand, and this feature allows the real-time, high accuracy system even in varying environments. Then, DenseNet201 improves the classification through solving the vanishing gradient problem and enhancing the reuse of features. This hybrid model integrates Mediapipe's benefits to the tracking ability with Deep Learning of DenseNet201 and also much more accuracy and robustness against earlier approaches[8].

II. Related Work

Noraini et al. [9] reviewed the advancement in hand gesture recognition technology from wearable devices to vision-based systems that eliminate the need for such wearables use single cameras and active techniques to improve reliability, machine learning models, particularly CNNs, to improve the precision in uncontrolled environments. Some of the challenges persist with regard to continued gesture recognition, signer independence, and adopting real-world conditions.

Nusrat et al. [10] developed a wearable hand gesture recognition device using iMEMS technology. The glove mounted with 3-axis accelerometer reveals hand movement and translates it into digital signals, to be fed into an ATmega8L microcontroller that sends all the data wirelessly into a computer using XBEE modules. The system can identify specific gestures and communicate these as different controls such as cursor movement, 3D object movements, and gaming control. The device also enables real-time interaction of many applications by defining certain gestures and commands.

Chidvika et al. [11] developed a system that enables users to display text without necessarily having physical interaction with hardware devices like keyboards or mice. The system uses the MediaPipe for real-time hand detection and gesture recognition. Hand movements are picked up by certain hand landmarks on the palm of the hand. This movement is processed to let the user write or draw on the screen using an OpenCV canvas. This system functions through a relatively simple webcam, with gestures having the possibility of controlling options like changing the color of text or erasing content.

Shaurya et al. [12] developed a virtual mouse system that can be navigated and controlled with a simple hand gesture without using any of the external hardware. The system captures the hand gestures of the user using a camera and integrates technologies such as OpenCV for video frame capture, and MediaPipe Hands, which detects and recognizes gestures from 21 landmarks of the hand, along with PyAutoGui, which controls mouse and keyboard functions according to the recognized gestures. These technologies work in concert with each other to create an interface by which one could manipulate the cursor and then left-click, right-click, drag-and-drop, and all those actions that may only require an arm movement.

Rijul et al. [13] developed a system to correct body posture using MediaPipe and Federated Learning technology. For the corrective technology of body posture, 33 coordinates from the human body are extracted through MediaPipe Pose to identify wrong postures in yoga and sitting positions and correct them. In hand gesture detection, MediaPipe Hands tracks 21 landmark points on the hand and feeds this information into the classification neural network. Federated Learning decentralizes model training on users' devices for privacy protection and personalization through the application of the Flower API. The feedback is provided in real time, is lightweight, and does not store user data in the cloud.

Sajin et al. [14] have designed a real-time hand gesture recognition system that lays special emphasis on the recognition of American Sign Language (ASL) gestures. MediaPipe, a machine learning framework, is used to detect and track 21 landmarks on the hand including the fingertips and knuckles. An ANN is used with these landmarks as input features that have been split into four hidden layers. The ANN employs ReLU activation functions and softmax for the output layer in the classification of gestures from 27 classes. The coupling of MediaPipe for accurate hand tracking and ANN for classification makes the system very efficient for real-time applications such as ASL recognition.

Vijaya et al. [15] developed system that interprets hand gestures after computer vision. The system was built using Python and applied OpenCV for image and video processing and MediaPipe to detect hand gestures. The system further used a Convolutional Neural Network in classifying and recognizing gestures. With real-time tracking of gestures, this system is therefore offering an intuitive, touchless interface approach that has incorporated the accessibility to interface with computers for people who are physically disabled.

III. Proposed System

A. System Overview

This proposed system is a pair of smart glasses that allows users to control digital screens wirelessly through hand gestures. The system captures, interprets, and executes user commands based on predefined hand movements using built-in infrared (IR) cameras and gesture recognition software. These smart glasses use real-time image processing techniques and proximity-based detection to enable intuitive, marker-free interaction with digital devices such as laptops, projectors, and smartboards. In architecture design, this allows for low-latency operation so that user gestures can be quickly mapped to on-screen actions for a fluid hands-free experience. It is best suited for smart education, public displays, and touchless computing because it is a reliable and portable solution for interactive control.

B. Hardware Components

The smart glass hardware architecture is streamlined into two main parts: the infrared camera and the wireless transmission unit. These pieces work together in achieving hand gestures and sending control signals to the digital screen almost in real time.

a) Infrared (IR) Camera: The infrared camera is placed at the front frame of the glasses, which does the capturing in real time all the hand gestures. The infrared camera works on the principle of emitting infrared light whose reflection off the user's hands it collects through the sensor. The infrared technology use therefore guarantees that the camera can pick up hand gestures under low lighting or changing environmental conditions, and it is reliable for various uses. The camera is configured to capture everything in front of the user since hand gestures are most likely to happen in that area.

b) Wireless Transmission Unit: The smart glasses are connected to the target digital device, be it a laptop, projector, or smartboard, through the wireless transmission unit. The unit supports either Bluetooth or Wi-Fi so that the data might transmit with low latency for providing a very smooth user experience. The units transmit the detected gesture commands wirelessly to an apparatus to which a device is connected so that they can be translated into actions over the screen, such as scrolling and zooming or selecting.

C. Software Architecture

The software architecture will involve the processing of real-time hand gesture inputs, mapped to corresponding screen control actions. The architecture is based on computer vision algorithms and techniques through image processing using the Mediapipe and OpenCV framework.

a) Image Capture and Processing: An infrared camera captures a view of the hand gestures of the user continuously. The image therefore has to be preprocessed to remove noise, enhance contrast, and ensure that the hands are well defined in gesture detection. OpenCV is used in handling basic tasks of image processing, filtering, and segmentation in identifying what part of the image is the hand and isolate from the background.

b) Gesture recognition with Mediapipe: This process of gesture recognition relies on a real-time framework by Mediapipe, whereby it tracks the skeletal structure of the hand by detected landmarks in the images preprocessed. Mediapipe is used specifically to outline the fingers' joints and orientation of the palm of the hand so that the establishment of a skeleton model becomes possible, which is helpful in identifying complex gestures such as swiping, pinching, or zooming and maps these to certain control commands.

Fig. 1. Hand Landmarks of MediaPipe

c) Gesture-to-Command Mapping: Once their hand gestures have been identified and recognized, they are mapped to the predefined commands that correspond to screen control actions. For example, a swipe can scroll through content, a pinch will zoom in or out of it, while a tap simulates a click. As such, the mapping of gestures onto commands ensures that the user's hand movements are intuitively translated to the screen, thus moving towards a more natural interaction with the interface.

D. Gesture Detection and Processing

The gesture detection and processing system captures and interprets the hand gestures of the user in real time to provide fast and accurate interaction with a digital device. The core steps involved in the process are as follows:

a) Gesture Capture: Infrared camera scans the space in front of the glasses for hand movements. Real-time capture by the camera of hand gestures by the user is transmitted to the connected digital device for processing.

b) Image Processing: It is received by a digital device and, using OpenCV, separates the hand from its background through filtering and segmentation. Thus, the system would process only those movements induced by the hand of interest while ignoring other nonmeaningful elements within the background.

c) Gesture Recognition: On the digital device, Mediapipe identifies key landmarks like finger joints and palm orientation to establish a skeletal model of the hand. The system detects specific gestures like swiping, pinching, and zooming and relates these to specific commands.

d) Proximity Filtering: The glasses use proximity-based filtering. This will ensure that gestures beyond a certain distance range from the glasses do not enter the system. The system also utilizes gesture recognition based on one hand at a time for presentation as other hands may interfere in crowded spaces. Thus, the interaction is focused on intended gestures of the user.

e) Command Execution: Once the digital device has processed and recognized the gesture, it maps the gesture into a control command such as scrolling, zooming, or selecting. The corresponding action gets executed on the digital screen in real-time, giving the user the ability to interact seamlessly with the device.

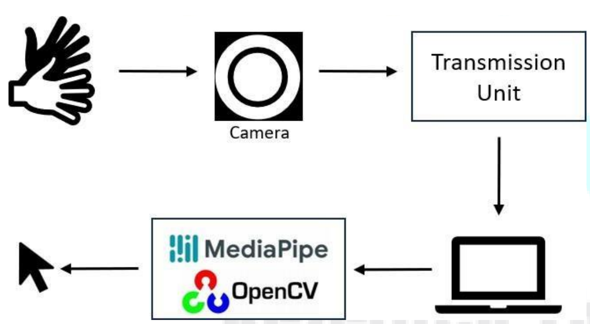

Fig. 2. Flowchart

IV. Results

The developed system passed the test by being successful with gesture-controlled mouse navigation using hand landmarks without needing any additional hardware setup. It enabled camera video frames to capture real-time feeds and suitably differentiated hand gestures through the MediaPipe Hands framework, which recognizes 21 key landmarks of the hand. It gave precise and fast feedback in real time.

Fig. 3. Scroll Up

Fig. 4. Scroll down

Fig. 5. Zoom In

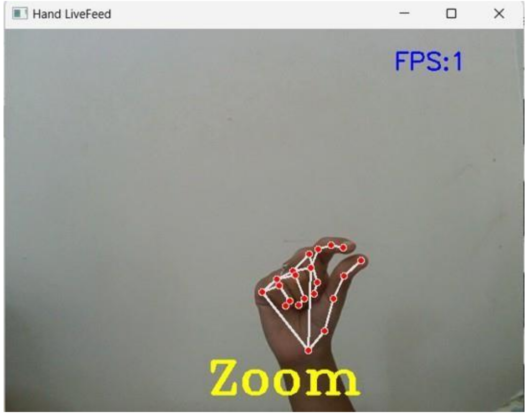

Fig. 6. Zoom Out

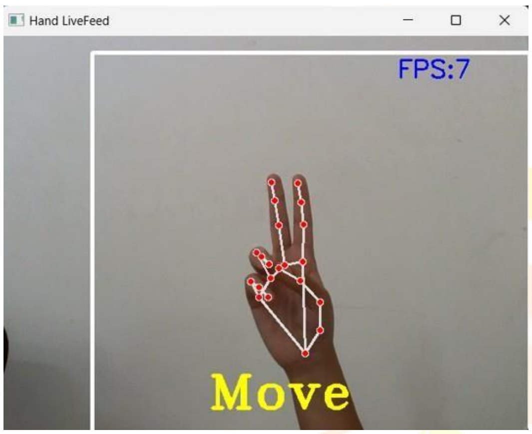

Fig. 7. Drag/Select/Move

V. Future Scope

The scope of further research and applications of future smart glasses with hand gesture recognition is enormous. Some other integration with additional sensors, like depth cameras, might achieve better accuracy in gesture detection in more complex environments. Such sensors would make the system more capable of recognizing subtle gestures as a result of a better understanding of spatial movements. The range of the system can also be extended to enable long-distance interaction. This will enable the glasses to work beyond the digital device at larger distances. This would prove especially useful for use in large environments such as conference halls where the presenter has to control a display from a distance. For example, if high-power Bluetooth or long-range Wi-Fi capabilities are added to the range of the wireless transmission unit, the system should then remain in complete control from other parts of the room or the stage. The reason for this is that in applications such as corporate presentations, educational seminars, and public speaking events, flexibility and liberation are paramount. Another area with great potential for development in this field is the refinement of machine learning models in gesture recognition. Better algorithms may make this system even more adaptive and able to learn from a much larger set of gestures as well as adapt the behaviors of users. This can further be developed into a transmission unit capable of supporting more than just Bluetooth and Wi-Fi connectivity, thus enabling the system to interface with a greater variety of devices. The system, therefore, has high prospects for use in healthcare and assistive technology applications. For example, people with reduced mobility and other physiologic disabilities will be able to interact with digital devices using hand gesture commands, thus making the process more practical and user-friendly in terms of accessibility as compared to conventional ways of inputting commands. In the health sector, for example, the technology can be adapted for use in sterile operating rooms where touchless interaction is fundamental, such as during surgeries.

VI. Conclusion

The latest innovations in human-computer interaction include smart glasses that recognize hand gestures. This system makes use of IR cameras, computer vision technologies such as Mediapipe, or even OpenCV to provide markerless and hands-free control over digital screens in simple manners. Proximity-based filtering along the edge can combine proximity detection to count for the possibility of crowded environments by diminishing interference from other hands. Moreover, this system's ability to transmit gesture data in real time wirelessly to any device that may be connected gives it a high degree of practicality for education, public displays, and hands-free computing applications. Other potential directions for further development include more precise gesture recognition via advanced machine learning models, multiple wireless protocols to increase connectivity options, and coupling the system with a much broader range of devices and platforms. These enhancements will allow the smart glasses system to fundamentally change user interaction with digital interfaces, making operations more intuitive, flexible, and accessible for users in various sectors. In conclusion, the proposed smart glasses system represents a robust and innovative solution for gesture-based screen control that will pave the way for further developments of wearable technology and gesture-driven interfaces in the future.

References

[1] Y. G. Pame and V. G. Kottawar, "A Novel Approach to Improve User Experience of Mouse Control using CNN Based Hand Gesture Recognition," 2023

[2] K. S. Varun, I. Puneeth and T. P. Jacob, "Virtual Mouse Implementation using Open CV," 2019

[3] A. S. Rama Chandra Murthy, D. Anitha, N. M. Chowdary and S. Siddhartha Sai Kumar, "Mouse Cursor Control with Eye Gestures," 2024 .

[4] N. L. Reddy, R. Murugeswari, M. Imran, N. Subhash, N. V. K. Reddy and N. B. Adarsh, "Virtual Mouse using Hand and Eye Gestures," 2023

[5] A. Pandit, D. Dand, S. Mehta, S. Sabesan and A. Daftery, "A Simple Wearable Hand Gesture Recognition Device Using iMEMS," 2009

[6] George Sung, Kanstantsin Sokal, Esha Uboweja, Valentin Bazarevsky, Jonathan Baccash, Eduard Gabriel Bazavan, Chuo-Ling Chang, Matthias Grundmann, "On-device Real-time Hand Gesture Recognition," 2021

[7] K. Naithani, D. Malik, P. Kumar, M. Bilal and M. Singh, "Computer Control with Hand Gestures using Machine Learning (ML) and Computer Vision," 2023

[8] P. Padhi and M. Das, ”Hand Gesture Recognition using DenseNet201-Mediapipe Hybrid Modelling,” 2022

[9] N. Mohamed, M. B. Mustafa and N. Jomhari, ”A Review of the Hand Gesture Recognition System: Current Progress and Future Directions,” 2021

[10] N. Ansari, S. Awari, S. H. Movva, A. Parthasarathy and S. Poriwade, ”GesSpy: ML Driven Real Time Sign Language Detection,” 2024

[11] C. Gunda, M. Maddelabanda and H. Shanmugasundaram, ”Free Hand Text Displaying Through Hand Gestures Using MediaPipe,” 2022

[12] S. Guliani, R. Nagpal, R. Sehgal and A. Singhal, ”Gesture Controlled Mouse Navigation : Hand Landmark Approach,” 2023

[13] R. Singhal, H. Modi, S. Srihari, A. Gandhi, C. O. Prakash and S. Eswaran, ”Body Posture Correction and Hand Gesture Detection Using Federated Learning and Mediapipe,” 2023

[14] S. Xavier, V. B and M. L. Pai, ”Real-time Hand Gesture Recognition Using MediaPipe and Artificial Neural Networks,” 2023

[15] V. L. Adluri, P. Kadiyala, S. Gopu and S. Jangiti, ”Virtual Mouse with Hand Gestures Using Machine Learning,” 2023