Archives available at journals.mriindia.com

International Journal on Advanced Electrical and Computer Engineering

ISSN: 2349-9338
Volume 14 Issue 01, 2025

Virtual Mouse Using Hand Gesture And Voice Recognition

Mrs. A. A. Bamanikar[1], Rhushikesh Kale[2], Hitesh Pagar[3], Rushikesh Wakode[4], Manmath Sajjanshette[5]

[1]Assistant Professor, Dept. of CSE, PDEA's College of Engineering, Pune.
[2,3,4,5]Department of Computer Science and Engineering (CSE), PDEA's College of Engineering, Pune.
E-mails: kalerushikesh964@gmail.com[2], hiteshpagar10@gmail.com[3], rushikeshwakode408@gmail.com[4], manmathsajjan@gmail.com[5]

Peer Review Information

Submission: 08 Jan 2025
Revision: 11 Feb 2025
Acceptance: 04 March 2025

Keywords

Hand Gesture Recognition
Machine Learning
Computer Vision
Voice Assistant Integration
Virtual Mouse

Abstract

One of the most widely used tools for human-computer interaction is the mouse, which comes in three main varieties: wired, wireless, and Bluetooth. All of these, meanwhile, necessitate a physical connection—typically through a dongle—which makes device setup more difficult. We suggest a brand-new hand gesture- based cursor control method that does away with conventional hardware. Rather, it uses computer vision and machine learning. Convolutional neural networks (CNNs) implemented with MediaPipe are used in the suggested system to provide a reliable, hardware-independent gesture recognition technique. This technique expands on S. Shriram's approach by allowing users to manipulate clicks, drag objects, and move the pointer using a variety of hand gestures. Python and OpenCV provide the foundation of the system, which simply needs a computer and a camera.

JEL Classification Number: C88, O33

Introduction

The need for user-friendly, contact-free interfaces that improve accessibility and usability is growing as human-computer interaction (HCI) advances quickly. People with speech or hearing difficulties can benefit greatly from hand gestures, which are a generally known form of communication that enhances interactions with a natural, nonverbal layer. By combining voice help and hand gestures, a potent dual-modal interface is created that offers consumers a touchless, accessible experience, revolutionizing how they interact with Using The system captures and analyzes movements without any extra equipment except a camera. It identifies and tracks hand movements using pre-processing, background subtraction, and edge detection through computer vision techniques to deliver the final gesture recognition. This system was developed in Python and utilizes Pynput, Autopy, and PyAutoGUI to navigate through and manipulate the screen. MediaPipe is used to track the movement of hands and fingers, while OpenCV is used to process images. Voice assistants can initiate or deactivate gesture recognition. They allow users to switch the system on or off depending on the need of a user. They can search Google using only their voice to easily find any information needed without hassle and hands-on involvement. Moreover, users may locate places with Google Maps just by giving voice commands. The file navigation is supported by the assistant, and people can navigate through their directories and file management without the need to even touch their keyboards or mice. It can, at a request, announce what the current date and time is, thus turning out to be a quick source of reference for users. Basic copy and paste commands may also be used through voice and this makes for easy multitasking and productivity in work. in addition, it is possible for the user to put the assistant into sleep mode (wake it up), thereby disabling voice interaction temporarily to save on interactions when not working. lastly, the system provides an exit command that, through voice, will close down the virtual mouse system smoothly. It would enhance accessibility since users can access their computer with hand gestures and voice commands to enable hands-free use, making it more user-friendly. The combination of gesture recognition and voice control delivers a robust, interactive solution that has proven to be particularly useful in hands-free applications, and hence the virtual mouse system is innovative, cost-effective, and accessible for HCI.

LITERATURE SURVEY

Munir Oudah et al., proposed a paper that goes by the title "Hand Gesture Recognition Based on Computer Vision," which was published in the year 23 July 2020 by "Journal of imaging" This paper described Hand gestures are used as nonverbal communication in many applications such as, medical application, human-computer interface, robot control and deaf-mute communication. The use of hand gesture research papers has applied various different methods, whether computer vision or instrumented sensor technology. Moreover, it sums up the effectiveness of these approaches that focus on computer vision techniques dealing with points of similarity and dissimilarity used dataset, detection distance (distance), techniques for classification, number and type of movements hand segmentation method & camera type. Along with a brief overview of some possible use cases. A common weakness in most interaction systems is that they only recognize and respond to specific types of hand used. It is more natural, easier, more flexible, and cheaper to manually control things rather than having to fix bug problems Static Hand Gesture Recognition using CNN Md. Zahirul Islam et al, So we have proposed this paper to illustrate what can still be done Research gate24 April 2019AbstractComputer is a part and parcel in our day to day life and used in various fields. The interaction between human and computer is done through traditional input devices such as mouse, keyboard etc. So hand gestures can be a good medium of interaction between human-computer system which will ease the process of interaction. Hand gestures differ from individual to individual in orientation and form. And hence there is non-linearity in this problem. Convolutional Neural Network has recently been proved to be the most powerful approach for representing as well as classifying images. Rescaling, zooming, shearing, rotation, width and height shift to create data augmentations. Three major applications performed to manipulate computer are mouse operation using hand gesture, controlling media player & third is creating shortcuts using static hand gesture. In static gesture recognition every gesture is mapped with specific application, for example opening word file or opening control panel. For static gesture recognition PCA is applied as the primary component. In any way, we had to design the Personal Assistant possessing brilliant powers of deductions together with the capacity to interact through the surroundings, by just one of the materialistic forms that human interaction encompasses. The capture of the request in audio via microphone and after processing the same request, making the device get a response of the individual that made the same request. using inbuilt speaker module. Gesture recognition and voice assistance technologies have significantly advanced the field of human-computer interaction (HCI), with each approach offering unique benefits for creating intuitive, touchless interfaces. Hand gestures, as natural and universally understood forms of communication, have been widely explored in HCI systems. Sivakumar et al. In the paper "Virtual Mouse Control Using Hand Gesture Recognition," Miziara et.al. For hand detection the authors used OpenCV and the gesture recognition by mapping contours. Their study revealed advantages of using webcams for gesture-based interaction, despite difficulties in different lighting conditions and a complex background. Similarly, Patel et al. As per Waggoner et al. (2019), in their Research study "Real-Time Gesture Recognition Using Convolutional Neural Networks," showed that CNNs can be used to identify complex hand gestures with high accuracy. They made a conclusion that deep learning algorithms can alleviate the gesture recognition robustness against such real-world scenarios. Gupta et al. Based on the work of (2021), the article "Applications of Multimodal Interfaces in Assistive Technologies" addressed the increase of the usability of virtual mouse systems in accessibility. Their study noted the ability of the system to enable even physically disabled users to easily interact with a computer without touch. Jain et al. (2019), EU project "Gesture and Voice Recognition in Gaming and AR/VR" Gesture-based controls improve the user experience in AR/VR systems, and voice commands offer extra flexibility. Many of these early studies were based on very primary image processing and straightforward pattern recognition in the detection of gestures, basing such detection on subtraction of the background from an image and edges in static environments. Deep learning and CNN now has enabled gesture recognition systems to be extremely responsive and accurate, as has been seen in Zhang et al. (2016) and in the frameworks implemented in Media Pipe. These technologies allow real-time hand tracking, making gesture-based systems viable for practical applications, such as virtual mice, where users control a cursor by simple hand movements. This line of research reflects how gesture recognition has become an established input modality for HCI systems, though significant challenges remain regarding real-time accuracy and gesture tracking. For voice assistance, with speech recognition reaching advanced systems of NLP, complex commands can be accepted according to commercial versions like Siri and Google Assistant. Sharma et al. (2019) as well as Part et al. (2021) studied this possibility of gesture interaction and voice interaction. They presented that combining them provides a better, more complex user experience because some users cannot experience mobility. This dual approach to HCI inspires the design of a virtual mouse system that integrates real-time gesture recognition with voice assistance, enabling users to perform tasks such as Google searches, navigation, and file management hands-free.

METHODOLOGY:

1. **Methodology For Hand Gesture:** In this section, we present the block diagram and flow chart of Virtual Mouse Control Using Hand Gesture Recognition. We briefly describe how the system works and what it comprises

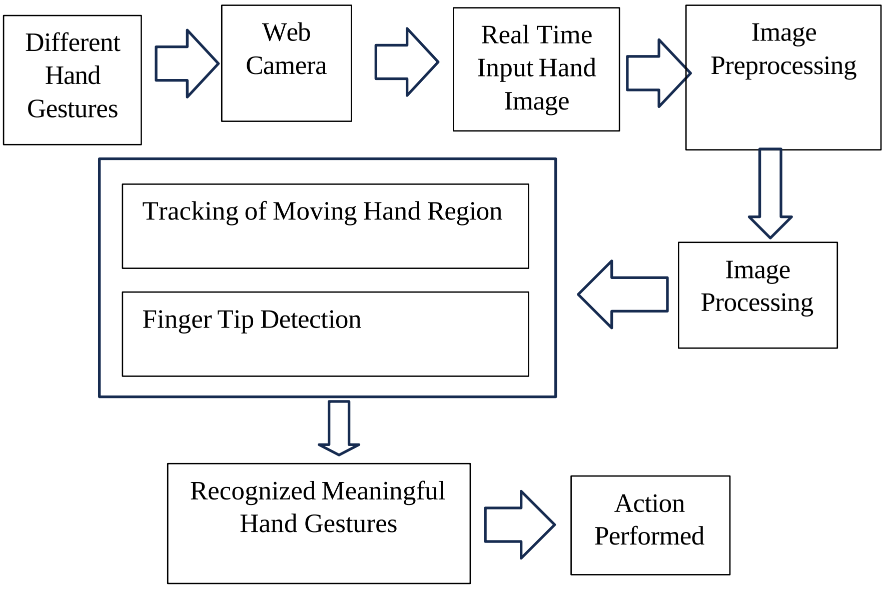


Fig 3.1 Block Diagram For Hand Gesture

Fig-3.1 shows the functional block diagram of the proposed system, illustrating how the system works. We are supposed to lift our hand towards the webcam. The webcam records the video, which captures frames. The pre-processing was carried out on the input image. The role of the image pre- primary processor is that it sets up a standard in the image. The process of resizing and pre-processing an image, so that two images may get similar heights and width, is standardization. Nowadays, for obtaining a good standard image, processing is done in the form of an image-processing technique. In this process of image processing, it moves after its hand touches the surface. Further, by employing MediaPipe and openCV, fingertip can easily be detected. Once it finds a hand and finger tips, it starts to draw. Hand landmarks and a box around the hand appear on the screen. A rectangular box for holding the mouse is drawn on the window pc. It decides which one of your fingers is up and which is down. Depending upon the detections of fingers on the mouse action is done by the program in order to work on the further frames and continues its working. It is the way that works the whole system.

We can use the following process for detecting and understanding hand movements. It starts with data recognition, using devices such as cameras, webcams, or Leap Motion to record the gesture. Next comes data preparation, in which the system strips away irrelevant information, sharpens the image and separates the hand from the background. Feature extraction and identification of the hand's features like its shape, edges, or key points, like fingertips. Then those are used to train a model using machine learning or deep learning techs (Support Vector Machines (SVM), Convolutional Neural Networks (CNN)

During the training phase, the model can identify gestures as they occur and relate them to specific actions or commands. The system gets properly tested using metrics

2. Methodology For Voice Assistance: Figure 3.2 It will be a voice-based information retrieval system proposed with a user-friendly, hands-free information access approach. It will first begin its workflow from the Speech Recognition Module which would convert the spoken queries into text and then forwarded to the Voice Assist Module, which acts as an intermediary The Voice Assist Module, therefore, acts as a mediator between the user and the backend of the system. It calls the Content Extraction Module and Text- to-Speech Module through API calls to the Python Backend. The Content Extraction Module analyzes the text query with utmost care and retrieves information from a pre-defined database or knowledge base. The Text-to- Speech Module then converts the retrieved information into speech that can be heard by the user, thus giving the user a voice response. This module integration allows users to interact with the system through voice commands only, thereby not requiring the use of traditional text-based input methods.

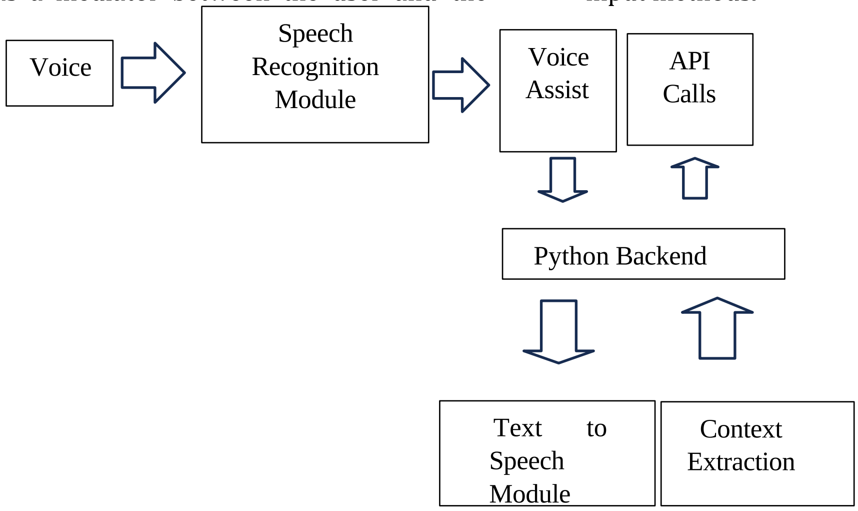


Fig 3.2 Block Diagram For Voice Assistance

The full-blown process of building a voice assistant consists of a series of defined stages that help the system recognize, interpret, and respond to user instructions appropriately. The process starts with acquiring the speech input; wherein the assistant captures the user's voice via a microphone. The details of sampling and quantization techniques are beyond the level of this input, however this input is then digitized into a series of digital data. Speech Preprocessing removing noise, filtering out unwanted sounds, and enhancing clarity of speech. This means using methods like sound filtration and feature extraction, using only the important parts like pitch, frequency, and tone to clean the audio for processing. After pre-processing of audio input, we enter the phase of converting speech to text. For this step, Automatic Speech Recognition (ASR) models are being used, e.g. HMM based models, deep learning algorithms like RNN Transformer models. They then analyze sound patterns and translate those into text. The generated text is subject to NLP (Natural Language Processing), which means tokenization, syntax analysis, and semantic as well. NLP processes user-input text by decomposing it into their structure and semantics and helps to predict the user's intent.

ALGORITHM :

1. **Hand Gesture**

Step 1: Introduction

Step 2: Start webcam video capture and system initialization. Step 3: Webcam frame capture

Step 4: Using Media Pipe and OpenCV for the detection of hands and hand tips, drawing hand landmarks, and a rectangle around the hand.

Step 5: Draw a rectangle around the computer window area, where we're going to use the mouse.

Step 6: Determines that which finger is raised.

Step 6.1: If all 5 fingers are up, then gesture is neutral and advance to the next step.

Step 6.2: Both middle and index fingers are up, cursor goes to step 2.

Step 6.3: If step 2 is performed and both index and middle fingers touch side by side this action will be a double click.

Step 6.4 (both index and middle fingers are still down): Perform a left click and continue at step 2.

Step 6.5: If middle finger is down, then we check if index is up. If so, we right click and then do step 2.

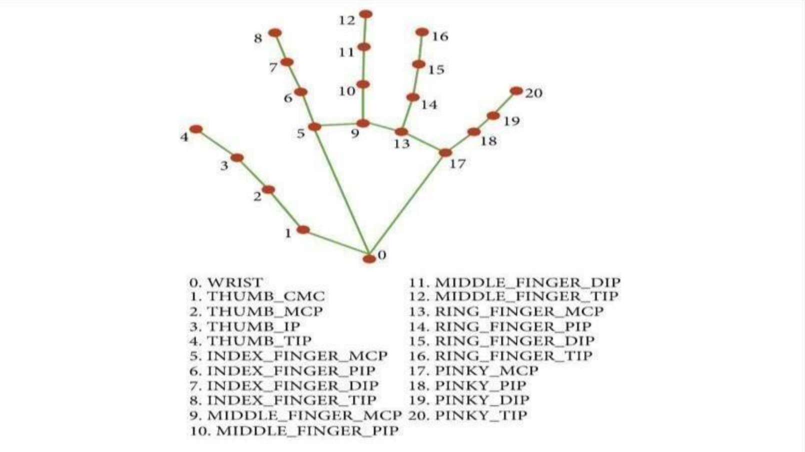
Tuning the finger (Step 6.6): It performs volume up or down by touching up of thumb and index finger to others tip and moving up and down.

Step 7: Exit (Press the EXIT key)

The AI Virtual Mouse System uses a camera. The

proposed AI virtual mouse system is based on frames captured by the webcam of a laptop or PC. Fig-
3.1 demonstrates the creation of the video capture object using the Python computer vision library OpenCV, and the web camera starts video capturing. The images captured by the webcam are forwarded to the AI virtual system.

The video will record and process; however, the AI virtual mouse system uses a webcam to take snapshots of each frame until the program is interrupted.



2. Voice Recognition

Step 1: Start

Step 2: Initialize the system and set up microphone access.

Step 3: Begin listening for a wake word (e.g., "Hey Assistant") to activate the voice assistant.

Step 4: Capture audio input through the microphone.

Step 5: Convert the audio input to text using a speech recognition library. Step 6: Process the text command to identify the intent.

- Step 6.1: If the command is a greeting (e.g., "Hello" or "Hi"), respond with a greeting and return to Step 3.
- Step 6.2: If the command is to perform a specific action (e.g., open an application, play music, check the weather), proceed to Step 7.
  - Step 6.3: If the command is a query (e.g., "What's the time?", "Who is the president of the U.S.?"), retrieve the relevant information and return the response to the user, then return to Step 3.
  - Step 6.4: If the command involves a setting adjustment (e.g., "increase volume," "turn off lights"), perform the action and return to Step 3.

  Step 7: Execute the identified action based on the command.

  - Step 7.1: For application-related commands, launch the application or perform the related function (e.g., open a browser, start a calculator).

  Step 8: Provide feedback to the user by voice, confirming the action or delivering the requested information.

  Step 9: Wait for further commands if in continuous listening mode, or return to standby.

Step 10: If the command to exit or "stop listening" is detected, terminate the assistant.

Step 11: End.

Results & Discussions :

For testing this algorithm Computer is set not to perform any mouse actions on the screen. No Action on the Screen as shown in Fig-4.1. All fingers have tip Id = 0, 1, 2, 3, and 4. The computer has been configured not to act for any mouse events on the screen.

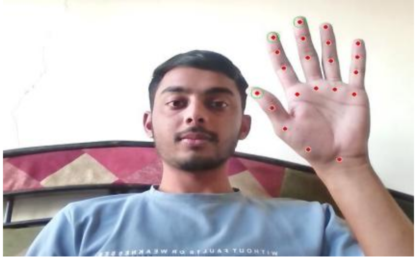


Fig.4.1 Neutral Gesture

For controlling the mouse cursor, navigate the computer window. It makes the mouse cursor to move around the computer window by using Python's AutoPy package in case both of its index finger with tip Id = 1 and middle finger with tip Id = 2 are up, as shown in Fig-4.2.

Fig-4.2 Cursor Control

To simulate the left mouse button click using the mouse. The computer carries out the left mouse button click for both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up and the distance between the two fingers is less than 30px, as can be seen in Fig-4.3.

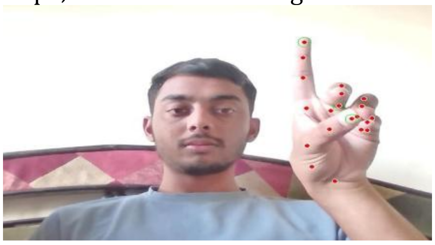


Fig-4.3 Left Click

To use the right-button click of the mouse. The computer is instructed to execute the right mouse button click provided that both the index finger with tip Id = 1 and the middle finger with tip Id = 2 are up and the distance between the two fingers be less than 40 px, as shown in Fig-4.4

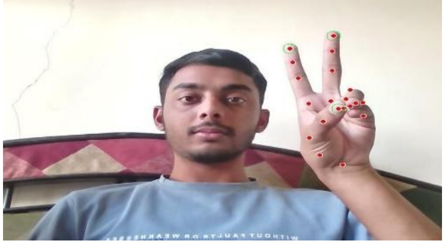


Fig-4.4 Right Click

The user is holding up their index and middle fingers with the remaining fingers folded; this is a "double-click gesture." The tips of the index (Id 8) and middle finger (Id 12) are prominent and erect. This gesture can be picked up by AutoPy of Python, which sends a double-click command to the computer screen in Fig-4.5

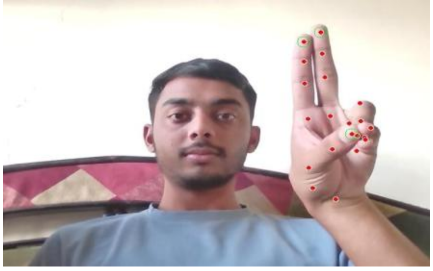
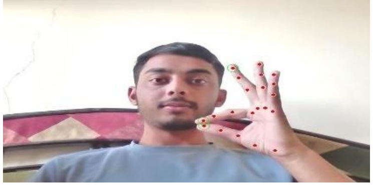


Fig-4.5 Double Click

the user includes his hand to make a gesture with the index finger and thumb touching in an "OK" shape, and the other three fingers (middle, ring, and pinky) pointing straight out. This gesture can be interpreted as a command for controlling volume and brightness settings as shown in fig 4.6

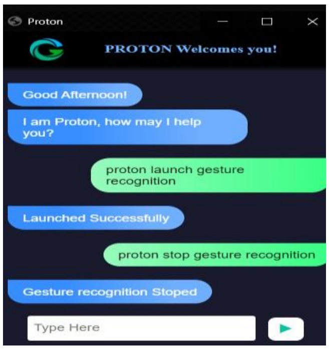
- Moving the hand up by this action increases the volume.
- Moving the hand downward decreases the volume.
- Moving the hand to the right increases the brightness.
- Moving the hand left decreases the brightness.

Fig-4.6 Volume And Brightness Control



Finally, the images above show the various mouse operations that can be carried out with hand gestures. Recognizing different fingertip ids allows you to perform various mouse operations.

Proton is a voice assistant that takes convenience to the next level. In addition to voice commands, you can control it with simple hand gestures. Open and close your hand to effortlessly launch and stop various functions, making your interactions with Proton even more intuitive and efficient shown in fig 4.7



PROTON Welcomes you!

Good Afternoon!

I am Proton, how may I help you?

proton launch gesture recognition

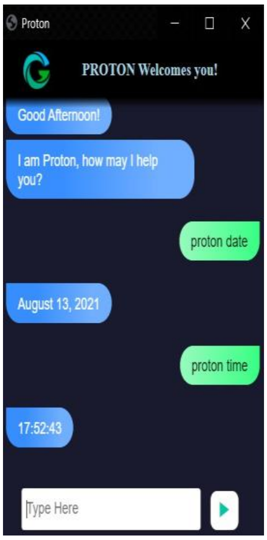
Launched Successfully

proton stop gesture recognition

Gesture recognition Stoped

Fig-4.7 Voice Assistance Proton for hand gesture

Proton is a voice assistant that takes convenience to the next level. In addition to voice commands, you can ask them current date and time also , making your interactions with Proton even more intuitive and efficient as shown in fig 4.8



PROTON Welcomes you!

Good Afternoon!

I am Proton, how may I help you?

proton date

August 13, 2021

proton time

17:52:43

Fig-4.8 Voice Assistance Proton give date and time

Limitations

Hardware Limitations:

Cam at the time of recording: Low-resolution cameras can fail to pick up hand gestures correctly under low lighting.

Microphone quality: Low-quality microphones may not pick up voice commands clearly, particularly in noisy settings.

Environmental Constraints:

Lighting conditions: In order, for hand gestures to be recognized, there needs to be a source of light. Inaccurate detection can arise from poor or uneven lighting.

Unintelligible speech: Voice recognition may struggle in noisier environments with concurrent speech.

Computational Power:

Both real-time hand gesture recognition and voice processing are resource-heavy tasks that need high processing power. The application may lag or throw errors on low performance systems.

Limited Gesture Recognition:

The complete dataset for synthetic gesture recognition is fixed gestures. If the natural characteristics of the users do not match with the programmed aspects then this feature may not be easy for the users.

Occlusion: if one hand blocks the other, or portions of the hand are out of view from the camera, gesture detection will fail.

Fatigue: Using certain hand gestures over an extended period may lead to user fatigue, thus reducing usability over time.

Voice Recognition Challenges:

The tone of voice: The system may misinterpret words spoken in a low, deep voice.

Muffled commands: Commands that sound similar might be misinterpreted (think "click" or "flick").

Latency: Real-time processing of voice commands, especially when cloud-based APIs are used, can incur delays.

User Experience:

Learning curve: New users might take time to get used to using hand gestures and voice commands side by side.

Data onboarding: Require a lot of new data to teach the model the differences in the context of gestures.

Simultaneous hand gestures & voice commands: Combined tasks like signing and speaking simultaneously can overwhelm users.

System Dependence:

Background Dependencies: The system may perform poorly if there are other memory-consuming applications running on the same device.

Network dependent: The system requires access to cloud API for voice recognition (e.g., Google Speech API) that needs a stable internet connection which may not be available all times.

Security and Privacy Concerns:

Access to Cameras: Users do not turn their webcams on and off likely throughout the day with the possible use of peer-to-peer networks.

Microphone access: Always-on voice recognition may inadvertently record sensitive conversations.

Data storage: The usage of gesture or voice data, if stored for processing or training can result in the potential misuse or breach of data.

General Limitations:

Dynamic backgrounds: The movement of unintentional people or objects in the background can disrupt gesture recognition

Inconsistent experiences: Limited

standardization of gesture or voice commands may result in inconsistent experiences for users. The system may not understand its context: there is a profound difference between an intentional gesture/command and an accidental one (e.g., random gesturing or casual speech).

Conclusions

In this paper, the authors present an innovative alternative to the traditional physical mouse by utilizing a computer vision-based approach that relies on a webcam to recognize finger and hand gestures. This virtual mouse system processes frames captured by the camera and applies machine learning algorithms to execute common mouse functions, such as moving the cursor, right-clicking, left-clicking, and scrolling. The system has been rigorously tested and shown to be of exceptional accuracy in addressing the shortcomings of earlier gesture-based systems. The paper also introduces a Voice Assistant that allows users to automate many computer tasks with simple voice commands. This includes web searching, file navigation, map- based location searching, and application launching, which will streamline user interactions and enhance productivity. The Voice Assistant functionality would be intuitive with versatility. Many complex tasks were made simple via straightforward voice commands. Future developments will integrate the virtual mouse and Voice Assistant systems using React Native to bridge the gap between desktop and mobile platforms. This will allow for real-time synchronization between devices, giving users a seamless and unified experience across platforms. With these developments, the proposed system minimizes the use of physical input devices and optimizes the control of devices through a combination of computer vision and voice recognition, thereby improving accessibility, usability, and convenience for users.

References

Munir Oudah, Electrical Engineering Technical College, Middle Technical University, Baghdad 10022, Iraq; "Hand Gesture Recognition Based on Computer Vision" Article. 23 July 2020

Zahirul Islam Department of Computer Science and Engineering University of Chittagong, Bangladesh; "Static Hand Gesture Recognition using Convolutional Neural Network"

Mayur V. Gore Department of electronics Government college of engineering Aurangabad; "Human Computer Interaction using Hand Gesture Recognition" Article. H.-T.V. July – 2014

Vivek S. Shende, Ghansham Daadi, Jitendra S. Chavan, Arjun Marrin, "Product Awareness through Hand Gesture Recognition for Real-World Applications", International Journal of Computational Intelligence Techniques, Vol. January – 2014

Abhay Dekate, Chaitanya Kulkarni, Rohan Killedar Department of Computer Engineering, AISSMS College of Engineering, Pune, Maharashtra, India; "A Study of Voice Controlled Personal Assistant Device" Article.

Rutvik P. Kshirsagar, Student, Diploma E&TC (3rd), Government Polytechnic Aurangabad, Maharashtra, India "Hand Based Gesture Recognition Technologies", Article. June 2019.

Girish B G, Project Guide and Assistant Professor, Department of Computer Science and Engineering; " A Smart System using Hand Gestures and Voice" Article. June 2022