[ICAETBM-2024]
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

Gesture And Voice Controlled Virtual Mouse

Prof. Supriya Bhosle, Saurabh Shinde, Swapnil Mohite, Akshay Mahajan, Swaraj Phalke, Dr. Chandrakant Kokane

Department of Information Technology, Nutan Maharashtra Institute of Engineering and Technology, Talegaon Dabhade

ABSTRACT

One of the most crucial components of a computer is the mouse. When using a Bluetooth or wireless mouse, a dongle and battery are required for the mouse to be powered on within the computer. The gesture-controlled mouse, which permits voice commands and hand gestures to facilitate human-computer interaction, is proposed in this research study. For the hand gestures and voice instructions, we require a camera and microphone. It will use a variety of hand motions to carry out activities like left and right clicking. You can also choose to drop and drag, enlarge and decrease the window, and perform a lot of other things. This system was built with the Python programming language (3.8.10).OpenCV, MediaPipe, and other more Python packages are installed on this machine

Keywords: OpenCV, Gesture Recognition, Virtual Mouse, Voice Commands.

I. INTRODUCTION

Human-computer connection is growing more and more convenient in daily life, and computer use has become a need in our existence. Although most individuals take these facilities for granted, using these gadgets properly presents significant challenges for those with disabilities. This research describes a gesture-based artificial intelligence (AI) virtual mouse system that employs hand gestures and hand tip detection to emulate mouse operations on a computer through computer vision. The primary objective of the proposed system is to perform computer mouse cursor and scroll operations using a webcam or a computer's built-in camera in place of a conventional mouse device [1,2]. Using a webcam or built-in camera, we can follow a hand gesture's fingertip and perform scrolling and mouse pointer operations. We can also move the cursor with an AI virtual mouse that is gesture-based. As part of an HCI with the computer, computer vision is utilized to recognize hand movements and tip detection [3,4,5]. The goal of gesture recognition technology is to develop tools that enable information to be sent by human gestures. When using gesture recognition, a camera records a person's movement and transmits that information to a computer, which then uses the movement as input to control an object or program. The purpose behind developing gesture recognition technology is to handle critical information and enhance human-computer interaction [6,7]. Data gathering, hand placement, hand recognition,

and gesture guiding are the key elements of the gesture recognition process [8,9]. Expressions of emotion or movement are called gestures. There are gestures and body language..

II. LITURATURE SURVEY

Gesture Recognition: A Review by Sundus Munir, Shiza Mushtaq, Afrozah Nadeem, Syeda Binish Zahra [1] System able to separate and analyze particular human gestures used for message management or conveyance. Hand detection methods, RGB color schemes, webcams, real-time tracking techniques, hidden Markov models, depth mapping techniques, and Kinect cameras are some of the approaches covered.

AI virtual mouse using gesture recognition, author: Abhishek R. Shukla [2] According to this article, the aim of the project is to develop a gesture-recognition-based AI-driven virtual mouse system that can replace conventional hardware mice. issues with things that require external devices, particularly with batteries and adapters. Abhishek R. Shukla's paper establishes a consensus on an instrument-free AI virtual mouse by highlighting the value of human computer interaction (HCI) and the shortcomings of present mouse technology. For manual detection, the system makes use of a unique machine learning method built on deep learning. The algorithm dispenses with the need for a hardware mouse by enabling users to digitally operate their computers. This includes the ability to scroll, click left or right, and move the cursor using gestures.

Virtual Mouse with Gesture Control using Artificial Intelligence: Rekha BN, G Satish, Sampat Kundanagar, Vikyath Shetty, Yogesh R Bhangigoudra[3] According to the publication, the system leverages sophisticated Python libraries like MediaPipe and OpenCV to assist with cursor control. Users can left-click, drag, and change system parameters like brightness and volume with gestures. Without the need for extra computers, the project recognizes voice and hand motions using computer vision and machine learning. Furthermore, the review could explore developments in computer vision, speech recognition, language, and managerial guidance, as well as the larger subject of AI-focused human-computer interaction. Research on virtual mice, voice assistant integration, gesture recognition in human-computer scenarios, and machine learning (e.g.,MediaPipe) are some possible areas of focus for this study.

Gesture controlled virtual mouse with voice automation, by Prithvi J, S Shree Lakshmi, Suraj Nair and Sohan R Kumar[4] The motion-controlled virtual mouse shown in this research study is made to respond to voice commands and gestures used in human-computer interaction. The system's architecture consists of two modules: one uses MediaPipe for hand detection, while the other makes use of gloves in complementary colors. It makes use of the most cutting-edge computer vision and machine learning techniques, particularly MediaPipe's CNN, to guarantee precise, dependable, and effective gesture and voice command execution. The system has speech automation features to improve usability and convenience, and it combines two modules to accommodate various client needs for manual dialing

Automatic feature extraction with memory and gesture recognition using deep learning algorithms Author: Rubén E. Nogales * and Marco E. Benalcázar[5]Because it is a high-standard recognition problem, scholars are interested in the problem of gesture recognition. Feature extraction and selection can be used to address the dimensionality issue. In this regard, evaluation models are advised for both automatic and manual feature extraction. CNN and BiL STM do automatic feature extraction, while the central preference statistical function is used for manual feature extraction. Additionally, these characteristics have been assessed in classes including Softmax, ANN, and SVM.

III. METHODOLOGY

1) Introduction: This article offers the required libraries for bespoke applications, such as enum, mediapipe for hand tracking, pyautogui for mouse handling, math for math operations, and cv2 for computer vision.
2) Libraries used in the project include PyAutoGUI, MediaPipe, OpenCV, and others.
3) Motion coding: Coding movements according to the angle and location of joints in the human body is one of the most effective techniques. Additionally, movements can be accessed by gesture recognition systems through the analysis of the motion of particular objects or items over time. This could involve monitoring hand movements.
4) Identification of Hands: Computer vision is used by several gesture recognition systems, like MediaPipe, to monitor and control cell signals. Particular locations on the hand, such as the palm, knuckles, and fingertip, are considered vital indicators.

5) Control operation: Upon detecting hand motions, the operating system advances the cursor. The hand's location within the camera frame can be used to gauge movement, and gestures can cause mouse events like clicks and scrolls.

6) Gesture Controller: To accomplish unique functions like modifying the screen's brightness, volume, and scrolling, we developed techniques like vertical and horizontal scrolling, as well as system brightness and volume changes. To handle various pointers and carry out commands, call the handle controls method.

IV. SYSTEM ARCHITECTURE

CAMERA:

The suggested AI virtual mouse system operates by utilizing the frames that are recorded by a laptop or PC's web camera. Using the Python computer vision package OpenCV causes the webcam to start recording video and creates the videocapture object. After that, the webcam records the frames and sends them to the AI virtual machine.

VIDEO CAPTURE: The AI virtual mouse technology uses a webcam to record every frame till the software is closed. The code that converts the video's BGR frames to RGB frames in order to locate the hands in each frame is shown below. defines the mp_hands function, which will allow us to keep track of our hands' locations. Identifying Which Finger is Up: Using the data from the library, it is now our responsibility to identify which finger movements are up and the corresponding coordinates of those fingers, then adjust the mouse function as necessary. This step entails keeping track of which finger is up based on the tip Id we discovered using the MediaPipe and their associated locations, after which the relevant mouse function is executed. It creates a loop that measures our hands' positions every 0.1 seconds. Computer Vision-Based Hand Gesture and Hand Tip Detection in Mouse Features Cursor Moving: Using the Python AutoPy package, the mouse pointer is made to move across the computer's window when the index and middle fingers are raised for the Moving Function. The HandLabel of the subject's right hand is used to initialize the HandRecog class. To initialize the hand result with the landmark points from the first frame, the initialize_hand_result method is used. When the optical flow is finished, this procedure is invoked. When the hand is moving, the update_hand_result method detects it and returns a None along with the gesture .ACTIVE. In the event that the hand remains motionless, a recognized gesture is looked up using the landmark points. After that, the function determines whether the gesture has altered. In the event that it has, the gesture frame count is increased and the current gesture is updated. The signed distance in pixels between two landmark sites is returned by the get_signed_dist function. The normal Euclidean distance between two landmark points is returned by the get_dist function.

Voice Assistant:

It receives input from the user, translates speech to text, and then evaluates the text to see if it meets the criteria.

If not, it will answer with "cannot recognize."

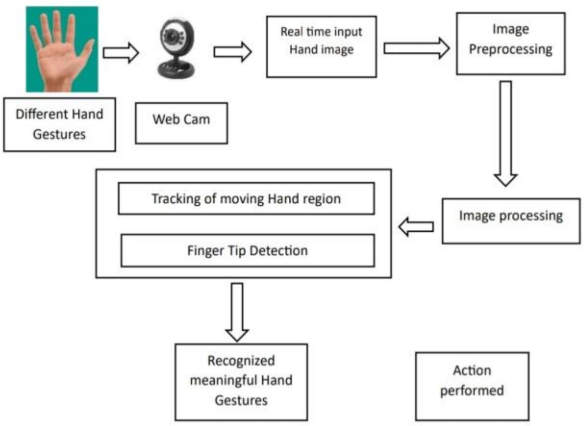

Fig 1: System Architecture

V. RESULT ANALYSIS

The proposed AI virtual mouse system introduces the concept of enhancing computer vision-based human-computer interaction. The limited amount of datasets makes it difficult to compare the AI virtual mouse system's testing. The webcam has been positioned at different distances from the user in order to assess hand motions and finger tip detection under different illumination conditions for hand gesture tracking and hand tip detection. The AI virtual mouse system is used to test the model multiple times in a variety of lighting conditions, including bright and dark. It is also tested up close to the webcam and at least four feet away from the screen. The AI virtual mouse technology had a 99 percent accuracy rate. Based on its accuracy of 99 percent, we may infer that the proposed AI virtual mouse system has worked well. Since the right click is the hardest gesture for computers to understand, accuracy is low. The accuracy of the right click is low since it requires a more complex gesture to perform the desired mouse movement. Furthermore, every other gesture has exceptional precision. Compared to previous methods, our virtual mouse model performed exceptionally well, with 99 percent accuracy. It is evident that the proposed AI virtual mouse has excelled all other virtual mouse models when it comes to accuracy. The novel feature of the proposed model is its ability to operate a computer in a way akin to that of a real mouse. This includes carrying out the majority of mouse functions, such as clicking left and right, scrolling up and down, and moving the mouse pointer by using fingertip recognition.

A. Gesture Controller Output:

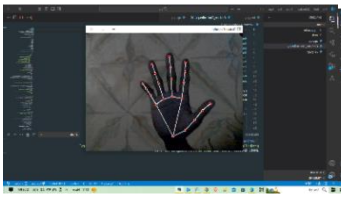

Fig 2: Natural Gesture

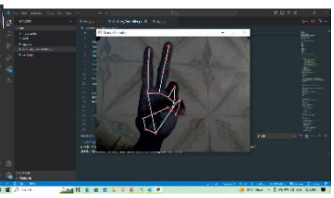

Fig 3: Moving Gesture

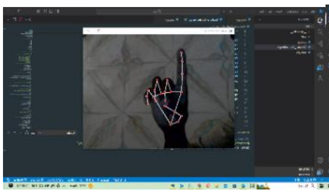

Fig 4: Right Click

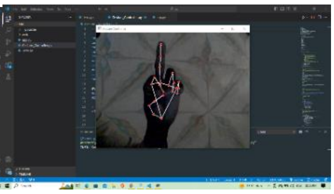

Fig 5: Left Click

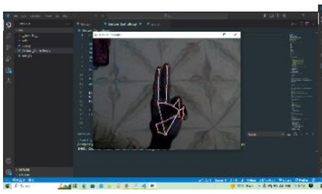

Fig 6: Double Click

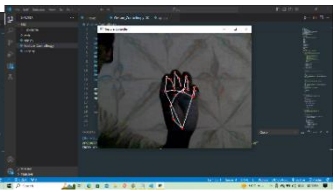

Fig 7: Drag

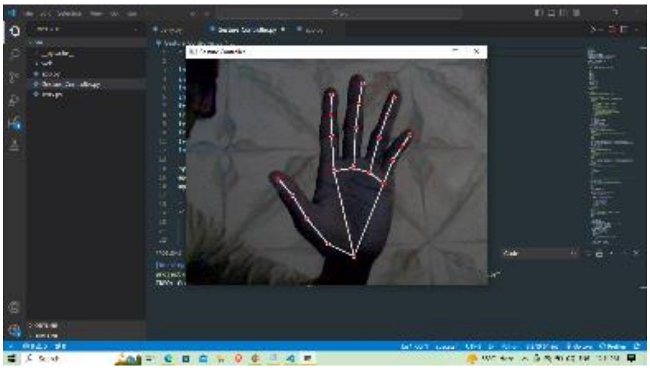

Fig 8: Drop

B. Voice Assistant Output:

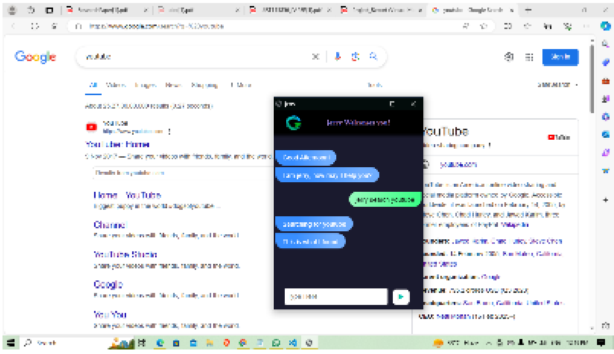

Fig 10: Search Location

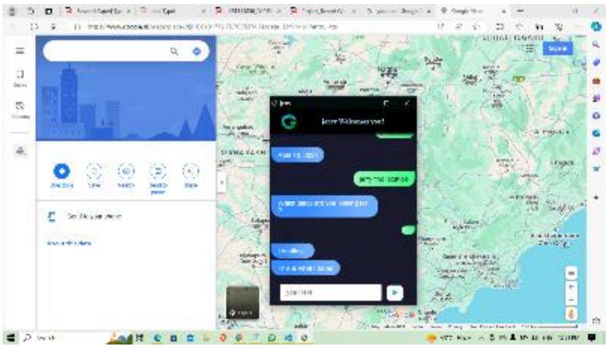

Fig 9: Google Search

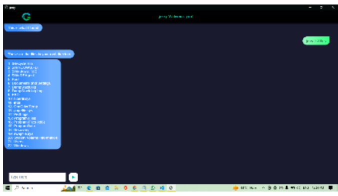

Fig 11: Display Files

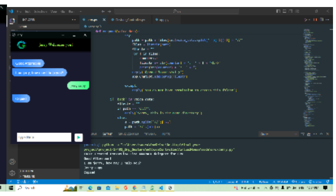

Fig 12: Copy And Paste

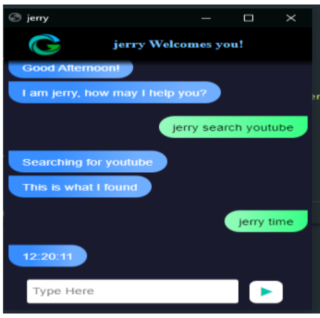

Fig 13: Time

Fig 14: Date

Fig 15: Exit

VI. CONCLUSION

Voice and gesture control Instead of utilizing a real mouse, users can connect with computers, portable devices, and voice commands through virtual machines. Using cameras and microphones, the system translates aural feedback and gestures into on-screen actions, giving users an easy-to-use, hands-free method to click, navigate, and operate apps. This innovative strategy encourages accessibility and offers a different form of counseling that is especially helpful in situations where conventional counseling resources are not available or are challenging to utilize.

VII. REFERENCES

[1] Kokane, C., Babar, S., & Mahalle, P. (2023, March). An adaptive algorithm for polysemous words in natural language processing. In Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022 (pp. 163-172). Singapore: Springer Nature Singapore.

[2] Kokane, C. D., Mohadikar, G., Khapekar, S., Jadhao, B., Waykole, T., & Deotare, V. V. (2023). Machine Learning Approach for Intelligent Transport System in IOV-Based Vehicular Network Traffic for Smart Cities. International Journal of Intelligent Systems and Applications in Engineering, 11(11s), 06-16.

[3] Kokane, C., Babar, S., Mahalle, P., & Patil, S. (2022). Word sense disambiguation: A supervised semantic similarity based complex network approach. Int J Intell Syst Appl Eng, 10(1s), 90-94.

[4] Kokane, C.D., Babar, S.D., Mahalle, P.N., Patil, S.P. (2023). Word Sense Disambiguation: Adaptive Word Embedding with Adaptive-Lexical Resource. In: Chaki, N., Roy, N.D., Debnath, P., Saeed, K. (eds) Proceedings of International Conference on Data Analytics and Insights, ICDAI 2023. ICDAI 2023. Lecture Notes in Networks and Systems, vol 727. Springer, Singapore. https://doi.org/10.1007/978-981-99-3878-0_36

[5] Kokane, C. D., & Sachin, D. (2021). Babar, and Parikshit N. Mahalle." Word Sense Disambiguation for Large Documents Using Neural Network Model.". In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE.

[6] Kokane, C. D., & Sachin, D. (2020). Babar, and Parikshit N. Mahalle." An adaptive algorithm for lexical ambiguity in word sense disambiguation.". In Proceeding of First Doctoral Symposium on Natural Computing Research: DSNCR.

[7] Kokane, C.D., Babar, S.D., Mahalle, P.N. (2021). An Adaptive Algorithm for Lexical Ambiguity in Word Sense Disambiguation. In: Patil, V.H., Dey, N., N. Mahalle, P., Shafi Pathan, M., Kimbahune, V.V. (eds) Proceeding of First Doctoral Symposium on Natural Computing Research. Lecture Notes in Networks and

Systems, vol 169. Springer, Singapore. https://doi.org/10.1007/978-981-33-4073-2_11

[8] Kokane, C., Babar, S., Mahalle, P. (2023). An Adaptive Algorithm for Polysemous Words in Natural Language Processing. In: Reddy, A.B., Nagini, S., Balas, V.E., Raju, K.S. (eds) Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems. Lecture Notes in Networks and Systems, vol 612. Springer, Singapore. https://doi.org/10.1007/978-981-19-9228-5_15

[9] C. D. Kokane, S. D. Babar and P. N. Mahalle, "Word Sense Disambiguation for Large Documents Using Neural Network Model," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2021, pp. 1-5, doi: 10.1109/ICCCNT51525.2021.9580101.