Engineering Applications
https://publish.mersin.edu.tr/index.php/enap
e-ISSN 2979-9201

Hand gesture and voice-controlled mouse for physically challenged using computer vision

Aarti Morajkar [1], Atheena Mariyam James [1], Minoli Bagwe [1], Aleena Sara James [1], Aruna Pavate*[1]

[1] *University of Mumbai, St Francis Institute of Engineering, Information Technology, India, morajkaraarti41@gmail.com, atheenajames96@gmail.com, minolibagwe@gmail.com, aleenajames998@gmail.com, arunaapavate@gmail.com*

Cite this study: Morajkar, A., James, A. M., Bagwe, M., James, A. S., & Pavate, A. (2023). Hand gesture and voice-controlled mouse for physically challenged using computer vision. Engineering Applications, 2 (2), 197-205

Keywords
HCI
Gesture
AI
Media pipe
Virtual Mouse

Research Article
Received:08.03.2023
Revised: 10.05.2023
Accepted: 23.05.2023
Published:29.05.2023

Abstract

A Human-Computer Interface (HCI) is presented in this paper to allow users to control the mouse cursor with hand gestures and voice commands. The system uses computer vision Efficient Net B4 architecture with no code ml to identify different hand gestures and map them to corresponding cursor movements. The objective is to create a more efficient and intuitive way of interacting with the system. The primary purpose is to provide a reliable and cost-effective alternative to existing mouse control systems, allowing users to control the mouse cursor with hand gestures and voice commands. The system is designed to be both intuitive and user-friendly, with a simple setup process. The highly configurable system allows users to customize how it works to suit their needs best. The system's performance is evaluated through several experiments, which demonstrate that the hand gesture-based mouse control system can accurately 100% and reliably move the mouse cursor. Overall, this system can potentially improve the quality of life and increase the independence of individuals with physical disabilities.

1. Introduction

Artificial intelligence is putting intelligence to make machines intelligent and capable of performing logical tasks designed by humans. Computer vision is part of AI that uses image samples to train machines. Computer vision provides different solutions such as disease prediction [1-2], landmine detection [3], designing adversarial samples [4-6], and improving input samples, analysis of samples [7-11] to make machine learning models more robust, Lip-reading recognition [12], and many more.

AI has a massive impact on people with disabilities to improve their lifestyles, providing the same access and services regardless of their disabilities. Gesture recognition is a technology that interprets hand gestures as commands using images. The voice assistant interface enables the hands-free operation of digital devices. This work aims to develop a new Human-Computer Interaction System that utilizes natural and intuitive hand gestures and voice commands rather than external mechanical devices such as a mouse. The proposed research introduces a novel system that utilizes hand gestures and voice commands to facilitate computer mouse movements for users. Voice assistants are hands-free and require minimal effort, allowing fast response times. This system benefits teachers, clinicians, and other users who can benefit from the hands-free operation and physically challenged people.

Many HCI systems capture human biological information as input, such as bioelectricity and speech signals, resulting in richer HCI modes. These new interactive methods made the HCI process more user-friendly and convenient. The field of human-computer interaction improved in terms of branching and interaction quality. Many researchers concentrated on using multimodality, intelligent adaptive interfaces rather than command/action-oriented ones, and active rather than passive interfaces instead of conventional interfaces [12].

This research aims to develop a cutting-edge Human-Computer Interaction System that simplifies the usage of natural and intuitive hand gestures and voice commands rather than relying on an external mechanical device like a mouse. Our proposed system utilizes hand gestures and voice assistant technology to enable users to efficiently control computer mouse movements, with the benefits of hands-free, effortless operation and speedy response times. This system has potential applications in various fields, such as education, healthcare, and defense, to enhance user experience and accessibility. Specifically, this system can benefit individuals with physical disabilities, in-car systems, and military operations. The objectives of the proposed system are:

1. To replace direct mouse clicks and points with gestures to control computers and other devices to simplify completing tasks.
2. To offer a cost-effective alternative to existing mouse control systems by eliminating the need for costly hardware such as additional sensors and special controllers using a deep learning model.

The significant contributions of this work improve the system's accuracy as it is affected by various environmental conditions such as lighting effects and more advanced mouse functions for the users such as drag and drop, moving folders, brightness, and voice controls.

The remaining work is organized as follows: section II discusses the related work, section III describes the methodology, section IV concludes the work.

2. Literature review

In recent years, a growing interest has been in developing new human-computer interaction (HCI) systems that replace traditional input devices such as the mouse with more natural and intuitive alternatives. One such alternative is hand gesture-based mouse control, which allows users to control cursor movements and perform mouse functions using hand gestures. In this paper, we present a review of the current state of the art in hand gesture-based mouse control, including recent developments in gesture recognition algorithms, sensing technologies, and applications of this technology in various fields.

Kabid et al. [12] proposed to create a novel mouse cursor control system that employs a webcam and a color-detecting technique. The system records every frame the webcam captures until the project is completed by implementing an infinite loop. Color-caught frames from the webcam captured frames are used to detect the color pixels on the fingertips. The distance between two detected colors is calculated using the OpenCV function. For clicking events, the proposed system uses close gestures. However, the system's efficiency could be improved due to the difficulties and complexity associated with background interference.

Rokhsana et al. [13] designed a real-time vision-based gesture-controlled mouse system. It employs color-based image segmentation for detecting hands, and contour extraction is performed to obtain the boundary information of the desired regions. The system uses a MATLAB function for moving operations, which calculates the centroid of the hand region. This approach is not limited to only controlling a mouse; it can control other devices such as televisions, robots in dangerous nuclear reactors, and other industrial setups. The system's sensitivity to surrounding noise and brightness can also be increased.

Kollipara et al. [14] implemented a system that utilizes libraries such as OpenCV, NumPy, and sub-packages. The model is built using computer vision techniques, and the detection and movement of the mouse are based on color fluctuations. The color detection model can be designed to identify a particular color from a colored image, which can improve the system's accuracy.

Reddy et al. [15] developed a model for recognizing motions, detecting fingers, and controlling mouse operations. The OpenCV library is used for image processing, and the PyAutogui module is used for mouse control. The algorithm's implementation involves two different approaches for mouse control: one using color caps and the other recognizing gestures made with bare hands. It involves integrating the video and processing the photos through backdrop removal. Background subtraction helps by ignoring steady items and only considering foreground objects. Fingertip detection includes finger guessing, circle recognition, and color identification. Gesture recognition involves identifying the skin tone, detecting contours, forming convex hulls, and inferring the gesture.

Sugnik et al. [16] come up with a technology that uses hand gesture recognition and image processing to create a virtual mouse and keyboard. The mouse operates using a convex hull technique, where gestures are detected or recorded and used to map the mouse's functionalities. The keyboard function uses a hand position system that records the user's hand position in a video. However, the Convex Hull algorithm may encounter issues and lose accuracy if there is external noise or flaws within the webcam's operational range.

Mishra et al. [17] used a deep convolutional neural network (CNN) called YOLOv3 to detect and localize the fingertips in the video frames. The authors used a custom-built data collection system that captured egocentric video of a user's hand performing various gestures. The annotated frames with fingertip locations used this annotated data to train and evaluate the YOLOv3 model. The proposed system showed promising results in terms of accuracy and efficiency. It could be applied to various applications involving hand gesture recognition, such as virtual or augmented reality interfaces.

Sharma et al. [18] used video processing techniques to track the position of the user's hand and translate its movements into corresponding movements of the computer cursor. To achieve this, the authors used a computer vision algorithm called skin color segmentation to detect the user's hand from the video stream. The authors applied a motion estimation algorithm based on the Lucas-Kanade method to track the movement of the hand. The authors also used a machine learning algorithm called K-Nearest Neighbour (KNN) to recognize hand gestures. This algorithm classifies hand gestures based on the fingers' and palm coordinates. The authors trained the algorithm using a dataset of hand gesture images and achieved a recognition rate of 95%.

Chaurasia et al. [19] provide a detailed review of self-healing concrete. The authors discuss the different mechanisms and materials that can be used for self-healing concrete and review recent research studies in the field. Overall, the work provides a comprehensive review of the topic, which can be helpful for researchers and practitioners interested in developing and implementing self-healing concrete.

Venkataramana et al. [20] present a novel Arduino-based system for converting hand gestures into speech to assist individuals with speech disabilities. The system utilizes gloves with sensors and an Arduino microcontroller to detect hand gestures and translate them into speech. The authors evaluated the accuracy and effectiveness of the scenario through experiments involving a sample of participants with speech disabilities. They found that the system could accurately detect and translate a range of hand gestures into speech. The authors suggest that the system has the potential for integration with other assistive technologies and could serve as an affordable and accessible solution for individuals with speech disabilities.

Fadiga et al. [21] have worked on reviewing previous research on the relationship between genetic variations in the HPA axis and stress-related disorders. It also describes a new study that found certain genetic variations in the HPA axis were associated with increased cortisol levels, subjective stress, and an increased risk of depression and anxiety. It also provides a literature review on the relationship between genetic variations in the HPA axis and stress. The authors note that previous research has found that genetic variations in the HPA axis can affect cortisol levels and the body's response to stress.

Sujatha et al. [22] comprehensively surveyed various hand gesture recognition techniques and their applications. It discusses the advantages and limitations of different approaches and provides insights into the challenges of developing accurate and robust gesture recognition systems.

Banik et al. [23] propose a hand gesture recognition system for controlling the computer mouse. The proposed method uses a depth-sensing camera to capture hand gestures and convert them into mouse movements. The authors evaluated the system's accuracy and achieved an average accuracy rate of 92.5%.

Solaunde et al. [24] compare the performance of four machine learning algorithms (Decision Tree, Random Forest, Logistic Regression, and Neural Networks) for credit scoring using a 1000 credit card applications dataset. The study evaluates the algorithms based on accuracy, precision, recall, F1-score, and AUC-ROC metrics. The results suggest that Random Forest is the best-performing algorithm, with an accuracy of 85.5% and an AUC-ROC score of 0.848. The study provides valuable insights for financial institutions in selecting an appropriate algorithm for credit scoring.

Huang et al. [25] provide a short literature review of the firefly algorithm and its variants, which are optimization algorithms inspired by the behavior of fireflies. The paper discusses the advantages and limitations of each variant and provides examples of their use in different applications. The authors conclude that the firefly algorithm and its variants have shown promising results and have the potential to be further improved and optimized.

The reviewed work has highlighted several issues and challenges related to hand gesture-based mouse control systems. For instance, one study [15] identified the problem of the model's sensitivity to specific color detection, leading to detection errors. Another study [16] reported limitations in detecting hand movements in a pre-defined zone and the lack of advanced mouse functionalities. Additionally, the system's accuracy is affected by various lighting conditions, further reducing the effectiveness of color and shape-based algorithms. To address these challenges, the proposed approach provides solutions that improve accuracy and efficiency and provide more advanced mouse functions for users.

3. Methodology

Gesture-controlled virtual mouse implementation using deep learning involves creating a pipeline to detect hand gestures and map them to mouse actions. The following are the steps:

1. Collection and processing of the Data: This is the initial stage of gathering information on the hand motions to operate the virtual mouse. The collected images are transformed into tensors of nodes. The data is preprocessed to remove pertinent details like hand position and orientation before being captured using a depth sensor or camera.
2. Gesture Recognition Model training: The model is trained using examples of labeled hand movements. A machine learning model recognized the hand motions.
3. Model: A Convoluted Neural Network (CNN) using the Efficientnet4 model is used for gesture recognition. The Efficient B4 trained on a custom dataset to accommodate customized gestures.
4. Run the model and Map Gestures to Mouse Actions: After building the pipeline, it is executed on a device to detect hand gestures in real time. The detected gestures were mapped to mouse actions, such as clicking, scrolling, or moving the cursor.

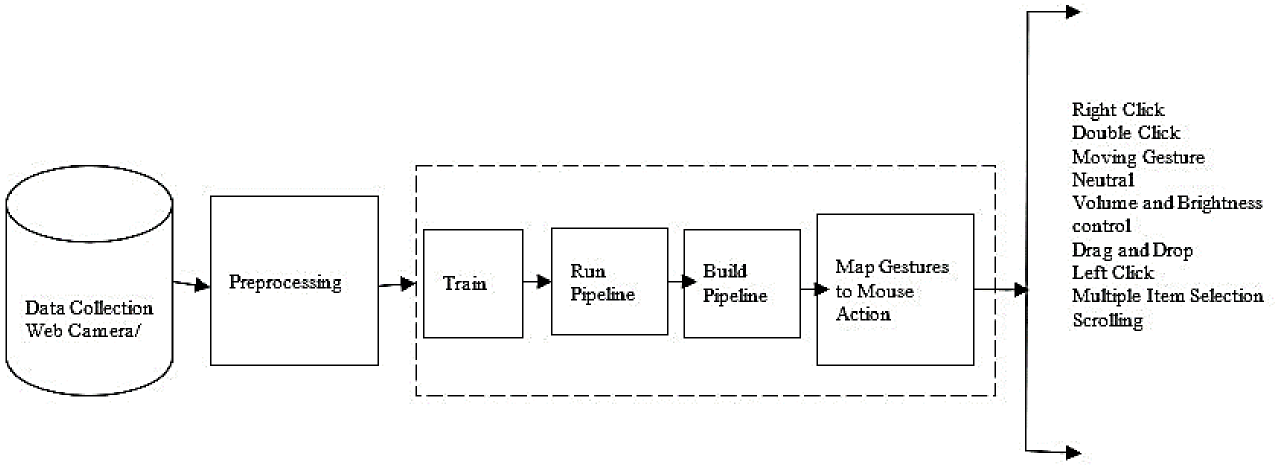

Figure 1. Gesture recognition model

A voice assistant can be added to a gesture-controlled virtual mouse implementation using MediaPipe. To do this, a voice recognition module can be included in the pipeline to detect and recognize voice commands from the user. The recognized voice commands can then be mapped to mouse actions or other actions, such as opening a file or launching an application. Figure 1 illustrates the implementation of a gesture-controlled virtual mouse with a voice assistant using MediaPipe. The hand gestures are captured using a depth sensor or a camera and preprocessed to extract relevant features such as hand position and orientation. The gesture recognition model is trained to recognize hand gestures from the collected data. It received the hand gesture as input and output as recognized gestures.

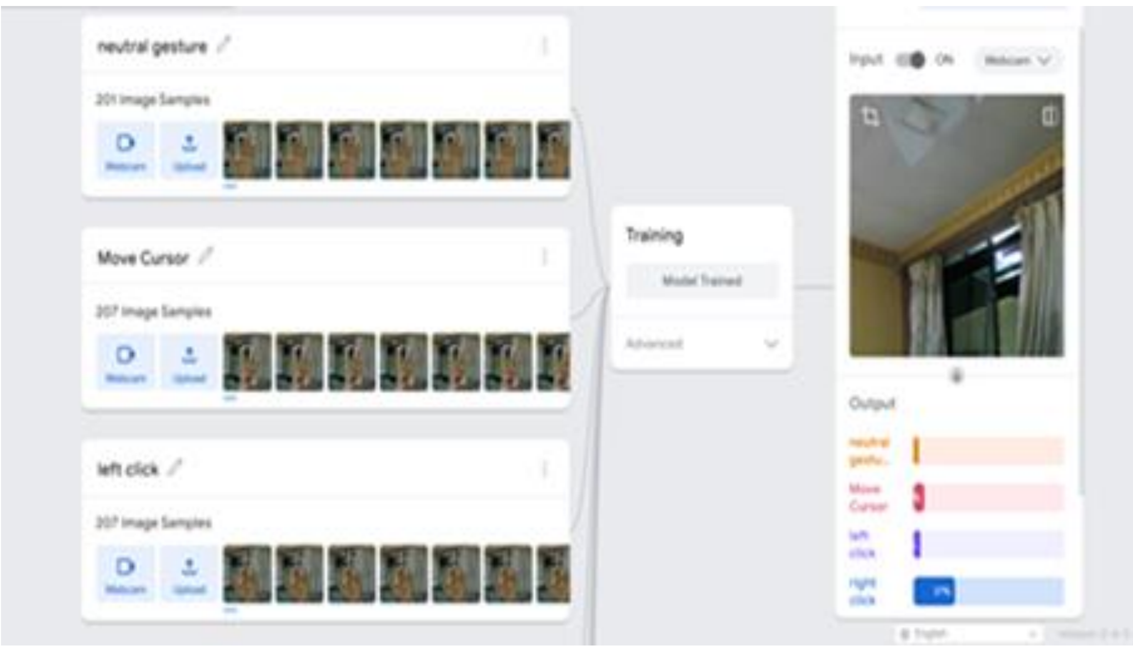

Figure 2. Model training using various gestures

The mouse action mapping module mapped the recognized hand gestures to mouse actions such as clicking, scrolling, or moving the cursor. It receives recognized hand gestures, the tracked hand position, and orientation as input and outputs the mapped mouse actions. Figure 2 represents the model training using various gestures. The input comes from the physical world, so collecting the proper samples is a challenging task. The virtual mouse module simulated the mouse's actions on the computer by accepting mouse actions as input. The voice recognition module detects and recognizes voice commands from the user, as shown in Figure 3. The implementation involved capturing and preprocessing hand gestures, recognizing the hand gestures using a machine learning model, tracking the hand in the video stream, mapping the recognized gestures and voice commands to the mouse and other actions, and executing the mapped actions on the computer.

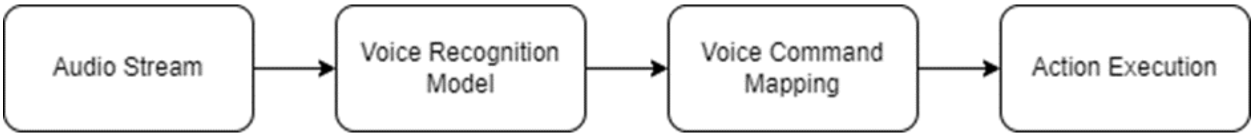

Figure 3. Voice recognition model

4. Results and Discussion

The hand gesture and voice recognition system incorporate ten gestures: neutral gesture, moving cursor, left click, right-click, double click, scrolling, drag and drop, multiple item selection, volume control, and brightness control. The voice assistant performs launch/stop gesture recognition, and content search on Google, identifies a location, navigates files, displays the current date and time, copies and pastes, sleeps/wakes up, and exit actions. In the proposed system, authors aimed to enhance human-computer interaction using computer vision.

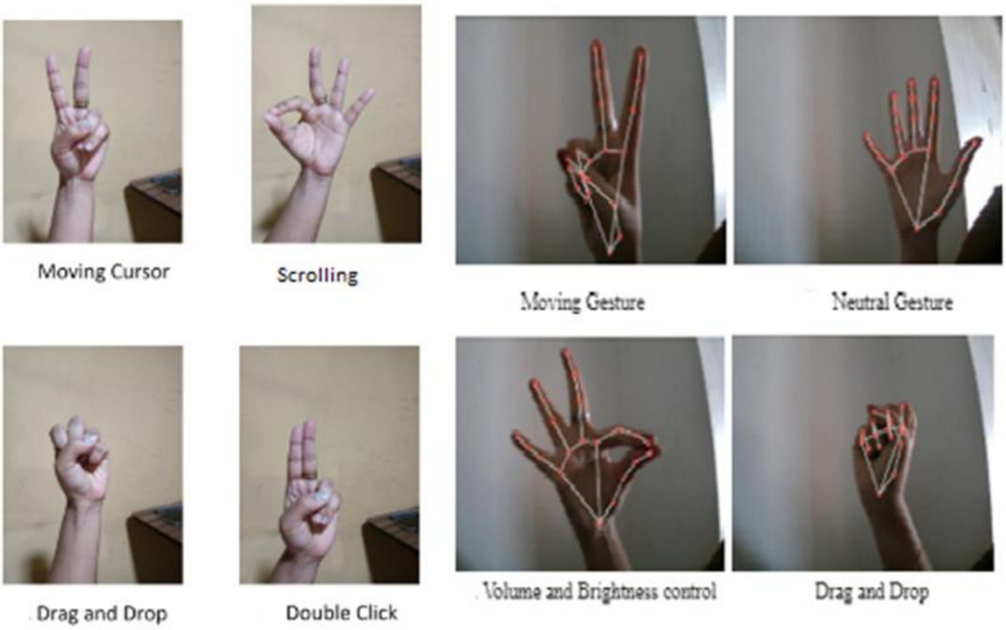

Figure 4. Gesture Controlled Output

In Figure 4, image indicates gesture and their specification user can perform a double-click using your index and middle fingers. You can place your index and middle fingers on the surface or mouse pad and then quickly tap both fingers simultaneously. This gesture simulates the action of double-clicking a mouse button. You can also use the place of your index finger to hold down the left-click button while rapidly tapping the surface with your middle finger to achieve the same effect. Image indicates to move the cursor using your index and middle fingers; you can rest the side of your index finger and middle finger on the surface or touchpad and then move your hand to move the cursor. Alternatively, you can use the pad of your index finger to hold down the left-click button while dragging the cursor with your middle finger. This gesture simulates the action of clicking and dragging with a mouse. Image is a neural cursor, a type of cursor control that uses brain-computer interfaces (BCIs) to detect and interpret neural signals to move the cursor. Therefore, no specific hand gesture is associated with a neural cursor as it does not rely on hand movements. Instead, users typically wear an EEG cap or other type of brain-sensing device to record and interpret their brain activity, which is then used to control the cursor on the screen. Image E, hold up your hand with your palm facing towards you. Curl your thumb and index finger towards your palm. Extend your middle, ring, and small fingers to be straight and perpendicular to your palm. Move your hand up or down to adjust the brightness, with the distance between your middle, ring, and small fingers representing the brightness level. The further apart they are, the brighter the screen will be; the closer they are, the dimmer the screen will be. To drag and drop shown in Image F, you can follow these steps:

1. Position your cursor over the item you want to drag.
2. Close your hand into a fist around the item, as if you are grabbing it.
3. Hold the left mouse button (or trackpad button) with your fist.
4. Drag the item to the desired location while holding the left mouse button.
5. Release the left mouse button by opening your fist, which will drop the item into its new location.

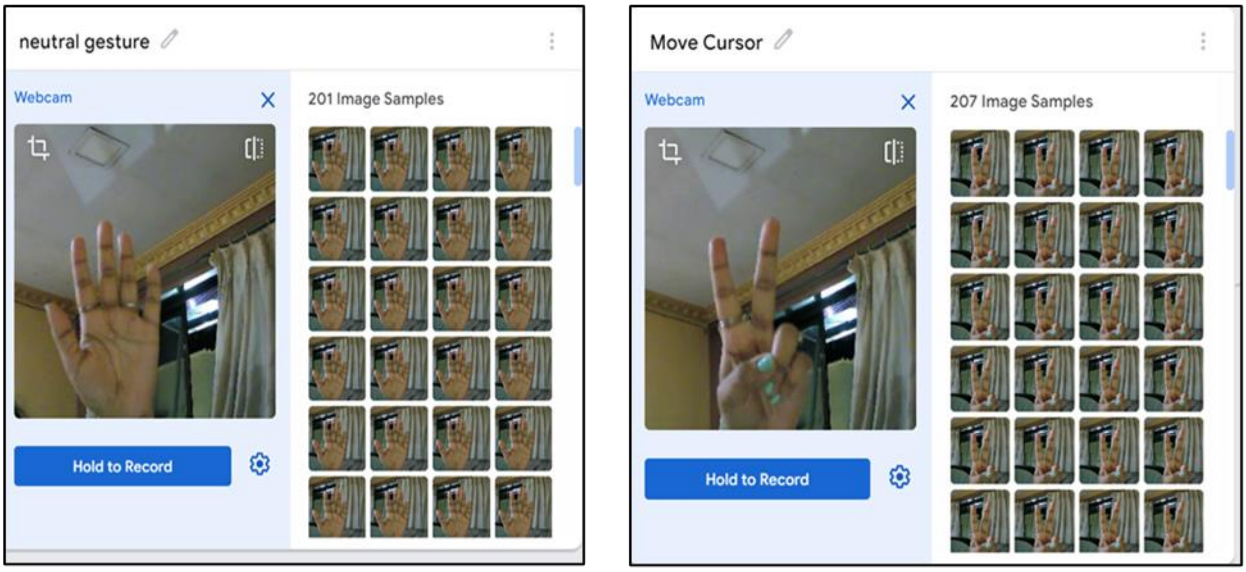

Figure 5. Hand gestures incorporated by gesture recognition system

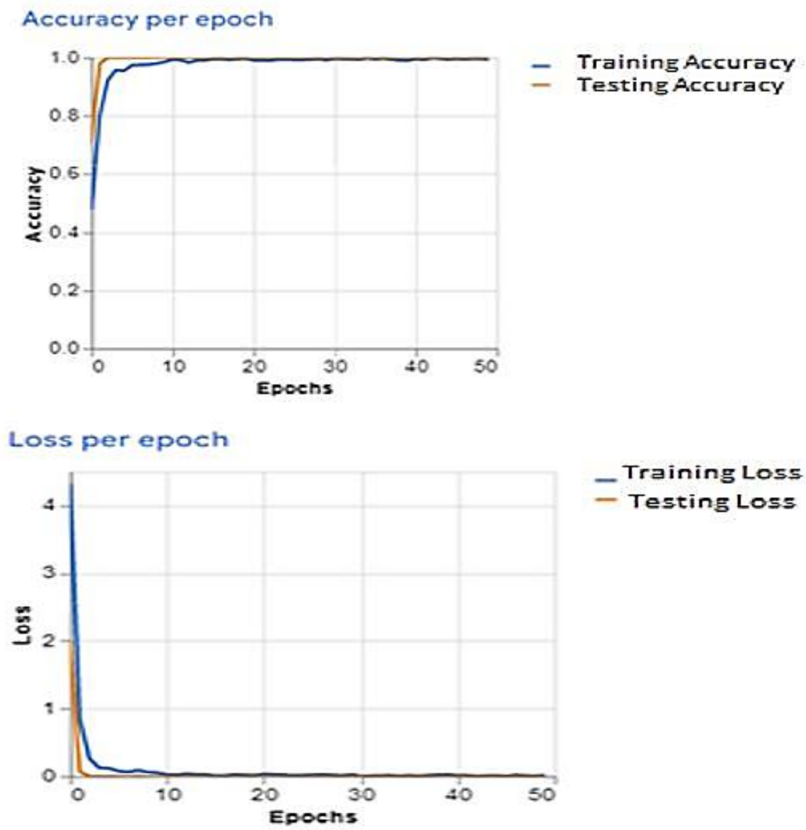

Figure 6. Performance of the model represented by accuracy per class and loss obtained by the model

The webcam is positioned at various distances from the user to monitor hand motions and gestures to detect fingertips as shown in Figure 5. Gesture's ability is assessed under diverse lighting conditions such as bright light settings, low-light configurations, at a much farther distance from the camera, at a closer distance from the camera, with a left hand, right hand, both hands in camera, different backgrounds, and different hands of individuals of varying ages. The Voice Assistant is tested by providing diverse input via the mic and executing various functions such as location, file navigation, current time and date, copy and paste, sleep/wakeup, google search, and start and exit under various conditions.

Table 1. Performance of the model with accuracy per class

Class	Accuracy	#Sample
Neutral gesture	1.00	32
Move cursor	1.00	28
Left click	1.00	24
Right click	1.00	24
Double click	1.00	25
Scrolling	1.00	25
Drag and drop	1.00	27

It is observed that every mouse action gives a few seconds of delay, but apart from that, all the gestures had excellent and high accuracy for all the classes as shown in Table 1 and in Figure 6.

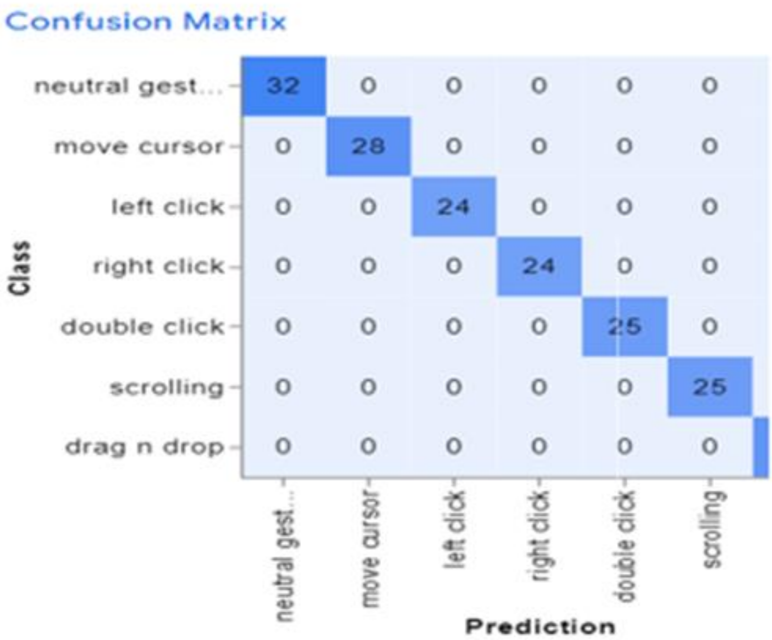

Figure 7. Confusion matrix for gesture detection system on x axis predicted samples and Y axis represents original samples

The Figure 7 shows the confusion matrix for the gesture recognition model by applying 50 frames from each class. The model's number of correct and incorrect predictions is represented by a tabular summary, as shown in Figure 7. We have seven classes, and the model predicts the gesture to one of the classes.

Some specific classes, such as natural gestures and mouse cursor, are more accurate to a given actual class.

Some classes are more challenging to predict, such as right click and left click, though it correctly predicted more samples. The proposed system is helpful for low-resolution images too. Many parameters affect the classifier's performance, such as resolution, image shape, boundaries of the pictures, and color. Finding the correct gesture is a challenging task.

The hand gestures are captured using an automated training machine learning model, showing promising results. Using hand gestures to control a mouse can increase productivity and ease of use, particularly for individuals with disabilities or those who find traditional mouse controls difficult.

The automated training machine learning model accurately detects and classifies hand gestures, allowing for smooth and precise cursor control. While further research and testing may be necessary to optimize the system's performance, the results thus far suggest that a hand gesture-controlled mouse could become a valuable tool for computer users in the future.

5. Conclusion

Human-Computer Interaction was a rapidly evolving technological sector. New technological advances were produced every year, and new efforts were taken toward seamless, natural contact between the computer and the user. It has progressed from the traditional keyboard and text-based interface to the more powerful mouse and touch-based interactions. With this study, we want to move forward to the next phase of virtual touchless interactions. This work developed a system for controlling the mouse cursor with a real-time camera. The technology was based on computer vision techniques such as CNN and could perform all mouse functions. However, due to the wide range of lighting and skin colors, it was impossible to obtain consistent results.

This method improves presentations for physically disabled individuals and enhances reliability. The system provides a comfortable PC and laptop experience for physically challenged persons. Future research involves eye movements to control mouse actions for those who cannot use their hands and introducing more functions to improve system performance.

Acknowledgement

This study was partly presented at the 6th Advanced Engineering Days [26].

We are thankful to Prof. Dr. Murat YAKAR, AED Symposium Chairman, for allowing presenting the work and Prof. Davron Juraev- Head of the Department of Scientific Research, Innovation and Training of Scientific and Pedagogical, Qarshi, Uzbekistan, for coordination and kind support in submitting the work.

Funding

This research received no external funding.

Author contributions

Aarti Morajkar: Conceptualization, Methodology, **Atheena James:** Data Collection, Literature review, **Aleena James:** Writing-Original draft, Visualization, **Minoli Bagwe:** Testing, Result. **Aruna Pavate:** Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

1. Pavate, A., Mistry, J., Palve, R., & Gami, N. (2020). Diabetic retinopathy detection-MobileNet binary classifier. Acta Scientific Medical Sciences, 4(12), 86-91.
2. Pavate, A., & Ansari, N. (2015, September). Risk prediction of disease complications in type 2 diabetes patients using soft computing techniques. In 2015 Fifth International Conference on Advances in Computing and Communications (ICACC) (pp. 371-375). IEEE.
3. Kumar, A., Pavate, A., Abhishek, K., Thakare, A., & Shah, M. (2020, February). Landmines detection using migration and selection algorithm on ground penetrating radar images. In 2020 International Conference on Convergence to Digital World-Quo Vadis (ICCDW) (pp. 1-6). IEEE.
4. Pavate, A. A., & Bansode, R. (2020). Performance evaluation of adversarial examples on deep neural network architectures. In Intelligent Computing and Networking: Proceedings of IC-ICN 2020 (pp. 239-251). Singapore: Springer Singapore.
5. Pavate, A., & Bansode, R. (2022, July). Design and analysis of adversarial samples in safety–critical environment: Disease prediction system. In Artificial Intelligence on Medical Data: Proceedings of International Symposium, ISCMM 2021 (pp. 349-361). Singapore: Springer Nature Singapore.
6. Pavate, A. A., & Bansode, R. (2022). Evolutionary algorithm with self-learning strategy for generation of adversarial samples. International Journal of Ambient Computing and Intelligence (IJACI), 13(1), 1-21.
7. Hamal, S. N. G., Ulvi, A., Yiğit, A. Y., & Yakar, M. (2022). Su altı yapılarının 3B modellemesi ve dokümantasyonunda kullanılan video ve fotoğraf çekimi yöntemlerinin karşılaştırmalı analizi. Journal of the Institute of Science and Technology, 12(4), 2262-2275.

8. Lalitha, R. V. S. S., & Srinivasu, P. N. (2017). An efficient data encryption through image via prime order symmetric key and bit shuffle technique. In Computer Communication, Networking and Internet Security: Proceedings of IC3T 2016 (pp. 261-270). Springer Singapore.
9. Srinivasu, P. N., Norwawi, N., Amiripalli, S. S., & Deepalakshmi, P. (2021). Secured compression for 2D medical images through the manifold and fuzzy trapezoidal correlation function. Gazi University Journal of Science, 35(4), 1372 – 1391
10. Yakar, M., Ulvi, A., & Toprak, A. S. (2015). The problems and solution offers, faced during the 3D modeling process of Sekiliyurt underground shelters with terrestrial laser scanning method. International Journal of Environment and Geoinformatics, 2(2), 39-45.
11. Shi, B., Hsu, W. N., Lakhotia, K., & Mohamed, A. (2022). Learning audio-visual speech representation by masked multimodal cluster prediction. arXiv preprint arXiv:2201.02184.
12. Shibly, K. H., Dey, S. K., Islam, M. A., & Showrav, S. I. (2019, May). Design and development of hand gesture based virtual mouse. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1-5). IEEE.
13. Titlee, R., Rahman, A. U., Zaman, H. U., & Rahman, H. A. (2017, December). A novel design of an intangible hand gesture controlled computer mouse using vision based image processing. In 2017 3rd International Conference on Electrical Information and Communication Technology (EICT) (pp. 1-4). IEEE.
14. Varun, K. S., Puneeth, I., & Jacob, T. P. (2019, April). Virtual mouse implementation using open CV. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 435-438). IEEE.
15. Reddy, V. V., Dhyanchand, T., Krishna, G. V., & Maheshwaram, S. (2020, September). Virtual mouse control using colored finger tips and hand gesture recognition. In 2020 IEEE-HYDCON (pp. 1-5). IEEE.
16. Chowdhury, S. R., Pathak, S., & Praveena, M. A. (2020, June). Gesture recognition based virtual mouse and keyboard. In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184) (pp. 585-589). IEEE.
17. Mishra, P., & Sarawadekar, K. (2019, December). Fingertips detection in egocentric video frames using deep neural networks. In 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ) (pp. 1-6). IEEE.
18. Gupta, A., & Sharma, N. (2020). A real time air mouse using video processing. International Journal of Advanced Science and Technology. 29, 4635-4646.
19. Chaurasia, R., Tiwari, A., Vishwakarma, A., Mandal, B., & Agnihotri, U. (2020). Hand gesture recognition and voice conversion system. International Journal of Creative Research Thoughts (IJCRT), Uttar Pradesh, 8(7), 1993-1997
20. Manisha, K., & Venkataramana, T. (2020). Arduino based gestures to speech converter gloves for deaf and dumb people. Journal of Emerging Technologies and Innovative Research (JETIR), 10(5), 220-223
21. Yasen, M., & Jusoh, S. (2019). A systematic review on hand gesture recognition techniques, challenges and applications. PeerJ Computer Science, 5, e218.
22. Sultana, A., Ahmed, F., & Alam, M. S. (2022). A systematic review on surface electromyography-based classification system for identifying hand and finger movements. Healthcare Analytics, 100126.
23. Ibraheem, N., & Khan, R. Z. (2012). Hand gesture recognition: A literature review. International Journal of Artificial Intelligence & Applications (IJAIA), 3, 161-174.
24. Patel, K., Solaunde, S., Bhong, S., & Pansare, S. (2022). Virtual mouse using hand gesture and voice assistant. Internatıonal Journal of Innovatıve Research ın technology, 9(2), 84-88
25. Huang, H., Chong, Y., Nie, C., & Pan, S. (2019, June). Hand gesture recognition with skin detection and deep learning method. In Journal of Physics: Conference Series (Vol. 1213, No. 2, p. 022001). IOP Publishing.
26. Morajkar, A., James, A. M., Bagwe, M., James, A. S., & Pavate, A. (2023). Hand gesture and voice-controlled mouse for physically challenged using computer vision. Advanced Engineering Days (AED), 6, 127-131.