Clareus Scientific Science and Engineering
Volume 2 Issue 4 April 2025
ISSN: 3065-1182

Enabling Cursor Control through Eye Movement using Hidden Markov Model

Citation: G Tanusha., et al. "Enabling Cursor Control through Eye Movement using Hidden Markov Model". Clareus Scientific Science and Engineering 2.4 (2025): 41-51.

Article Type: Research Article

Received: February 07, 2025

Published: April 16, 2025

G Tanusha*, P Havirbhavi and K Ashwini

Department of Computer Science and Engineering, Amrita School of Computing, Chennai, Amrita Vishwa Vidyapeetham, India

***Corresponding Author:** G Tanusha, Department of Computer Science and Engineering, Amrita School of Computing, Chennai, Amrita Vishwa Vidyapeetham, India.

Abstract

First and foremost, we describe a new technique that uses eye movement to control the cursor. Certain solutions impose stringent constraints on individuals with motor impairments and depend on tangible objects such as mice or touchpads. Using eye-tracking technology, our method establishes a connection between human visual attention and computer interaction. We study the subtle gaze patterns that emerge, enabling both cursor control and the formation of an interaction context sense. Our approach converts gaze data into a cursor's movements. It is very adaptable and accurate. Because cursor motions are produced by the pattern of gaze movement, the interface is more dynamically accurate and behaviourally specific. The context- aware interface learns and adapts, maximizing efficiency through its adaptability. Furthermore, our study presents a novel strategy to provide an eye-controlled mouse system using software tools. This system accurately recognizes facial features and interprets eye movements through image processing techniques. The system enables intuitive cursor control by interpreting observable eye movements. Transforming eye motions into a cursor's actions enhances user accessibility and hands-free interaction; in other words, its performance is captured in the experimental analysis.

Keywords: Eye-controlled cursor interface; Visual tracking technology; Pattern analysis; Software tools; User accessibility; User-centric design; Adaptive systems

Introduction

It is essential to achieve seamless human-computer connection in the modern technological world. Although mice and touchpads are common input devices, people with motor difficulties may find them difficult to use. Accessibility problems are exacerbated since people who are unable to physically operate these gadgets are frequently excluded from digital activities. Researchers and engineers are looking into creative ways to close the gap between technology and human capabilities to overcome these constraints. With the use of eye-tracking technology, digital gadgets may be operated hands-free thanks to intuitive interfaces that respond to visual signals. The development of eye-controlled cursor systems, which make use of patterns of eye movement to enable smooth navigation of digital interfaces and accurate cursor manipulation, is a noteworthy breakthrough in this field.

Using concepts from pattern analysis and computer vision, this research project investigates the creation of a novel eye-controlled cursor system. The foundation of our approach is the use of analytical methods to translate complex eye movement patterns into precise cursor motions. Our goal is to create a robust and adaptable control system that can appropriately accommodate a variety of user behaviors and preferences by training with observed eye movement data. We hope to go above the limitations of conventional cursor control methods and improve accessibility for people with motor disabilities by fusing cutting-edge methods with assistive technology resources. Our study seeks to pave the way toward a more comprehensive and adaptable human-computer interaction, empowering users of diverse abilities to interact with digital technology autonomously and effortlessly.

Eye-Controlled Cursor System

A novel cursor control system has been developed using statistical models like the Hidden Markov Model and eye movement tracking. This innovation allows for hands-free computer interaction, significantly enhancing accessibility for individuals with limited motor function. It leverages eye movements and blinks as control signals, offering a natural and intuitive interaction method.

1. Initialization of the FaceMesh Model and Open Web Camera.
2. Capture frame from Webcam (Real-time data).

Data Processing

The system employs advanced pattern recognition (pupil analysis) and image analysis techniques to convert complex eye movements into precise cursor control on the screen. This preprocessing ensures the system works efficiently and accurately by providing standardized output.

1. Extract eye-coordinates.
2. Calculate Cursor Movement.
3. Blink Detection.
4. Cursor Position Update.

Adaptive Interface

The interface is designed to be context-aware, improving accuracy and efficiency by learning and adapting to each user's unique interaction patterns over time. The interface can accommodate various levels of precision by adjusting the smooth factor, making it more adaptable and user-friendly.

1. Display processed frame with landmarks
2. Cursor Movement on the user's interface on eyeball movement

Practical Testing

Extensive research and trials have been conducted to assess the eye-controlled system's real-world effectiveness and usability, focusing on future applications and improvements. This design allows for iterative testing and improvement. After execution use the Exit condition "ESC" and release the webcam ensuring it meets the needs of the target audience effectively.

Here, the research paper provides a thorough summary of our work on employing a Hidden Markov Model to enable cursor control using eye movement in a structure. It begins with an Introduction that summarizes the study's purpose, goals, and significance by significant contributions of the work. Related Work follows, reviewing and highlighting relevant literature and pointing out any gaps. The references that we studied to compare our work with and highlight the novelty and uniqueness of the Hidden Markov Model are provided here. The dataset is used to train the model, to provide accurate cursor movement by extracting the probabilities of the movement using HMM. The design, development, collecting data, and image processing methods are covered comprehensively in the Methodology section. The combination of hardware and software is described by system architecture. The algorithm provided widely describes the logic of the model elaboratively. Testing procedures and metrics are provided in Experimental Setup and Evaluation, and results are evaluated and compared with current solutions in Results and Discussion. The document ends with a Conclusion that provides a summary of the main conclusions and their consequences as well as references. Our thorough examination and testing determine the approach's feasibility and effectiveness, providing the foundation for its use. We see a time when eye-controlled pointer technologies allow people to interact with technology without limitations.

Related Work

Information on several strategies for enhancing eye movement-based HCI for people with impairments may be found in the mentioned studies. Sivasangari et al. [1] and Ganga et al. [4] replaced conventional input devices like mice and keyboards with eye motions using a mix of Raspberry Pi pupil detection and webcam-based eye tracking. Based on eye movements, hands-free cursor control methods were developed by Mangaiyarkarasi and Geetha [6] and Dhanasekar et al. [2]. These systems employ machine learning and image processing techniques to properly track and analyze eye movements. Human-computer interaction systems that detect eye movements to enable hands-free cursor control and device communication through eye gaze were proposed by Narahari et al. [3] and Chandra et al. [5] using approaches like eye-aspect ratio and nonverbal communication.

Emphasis is placed on the challenges posed by conventional input devices, with particular attention to musculoskeletal disorders brought on by prolonged computer usage [4]. A number of the proposed solutions include external equipment, such as infrared-based eye-tracking systems or head-mounted devices [3, 6]. For average individuals, however, implementing these solutions could be costly and challenging [2]. In contrast, the approach proposed by Sivasangari et al. [1] makes use of a simple camera that does not require any infrared technology and offers a workable and reasonably priced replacement.

A hands-free cursor control system based on eye movements is provided by Dhanasekar et al. [2] as an example of how machine learning algorithms may enhance accessibility. Furthermore, Narahari et al. [3] and Ganga et al. [4] study eye-based interaction systems, emphasizing how important accurate eye movement tracking and interpretation is for managing cursor control. In addition to highlighting the importance of nonverbal engagement strategies, Chandra et al. [5] provide a method that assists physically challenged individuals through eye gazing. When taken as a whole, these studies demonstrate the significance of eye movement-based engagement techniques for enhancing the accessibility and independence of individuals with disabilities. Hidden Markov Models (HMMs) are a useful tool for improving the effectiveness and adaptability of cursor control systems for individuals with impairments, which provide an additional level of complexity to the study and prediction of eye movements.

The majority of the approaches discussed for eye tracking and cursor control rely on conventional image processing techniques or machine learning algorithms; however, the use of Hidden Markov Models (HMMs) offers a fresh perspective in this field. Unlike previous approaches that primarily focus on directly mapping eye movements to cursor positions, HMM-based systems can uncover underlying patterns and states related to gaze behaviors. With a more flexible and dependable base for cursor control, this capability might lead to improved accuracy and usability in human-computer interaction.

In the realm of research leveraging Hidden Markov Models, the proposed method for detecting fraudulent activities in electronic auctions by employing HMM in conjunction with the X-means clustering algorithm [21]. This advancement parallels the application of HMM in enabling cursor control through eye movement, showcasing the versatility of HMM across diverse domains, from fraud detection to assistive technology development. The research by P Havirbhavi et al. proposes a novel method for controlling the cursor using eye movements. Its flexibility and ability to adapt to individual user behaviors represent a notable advancement in seamless human-computer integration, signifying a crucial intersection where human capabilities and technological solutions converge within the domain of cursor control interfaces [24].

Dataset

The dataset used in this project is the MPIIGaze dataset, which is a well-known resource in the field of gaze estimation research, (see Fig. 1). It consists of pictures that show people's faces in a variety of environments, including different lighting, backdrops, and stances. Most importantly, the collection contains annotations describing where the participants' pupils are in the photos. To anticipate hidden states connected to eye movements, the project's algorithm reads these annotation files, extracts relevant data including picture pathways and eye pupil locations, and then trains a Hidden Markov Model (HMM) on the data. The annotated eye locations are displayed with sample images through the use of visualizations, which offer insight into the dataset and possible uses for the built model.

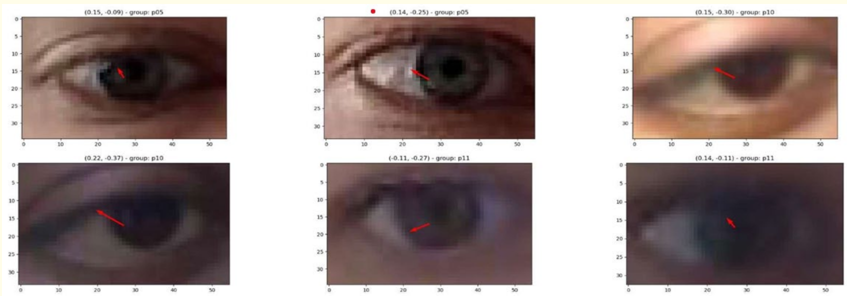

Figure 1: Defining the pupil position by visualizing the dataset.

In this work, eye pupil coordinate extraction, (see Fig. 2), involves a computationally intensive process to accurately determine the positions of the pupils inside images. Preprocessing is done on images initially to enhance quality and for additional analysis. Next, pertinent information is extracted from the annotations in the dataset, including pupil coordinates. These coordinates are modified to ensure accurate alignment with the associated photographs. The areas that are most likely to contain the pupils are then identified using feature extraction techniques based on pixel intensities, gradients, or other visual clues. Computers then employ thresholding or edge detection techniques to precisely pinpoint the students. Consequently, this enables further analysis such as gaze estimation since the pupil positions are saved as coordinates in the image space. Visualizations, which are often created using programs like Matplotlib, give important insights into the effectiveness of the extraction techniques employed in the study and assist validate the extraction process by presenting example pictures together with annotated pupil coordinates.

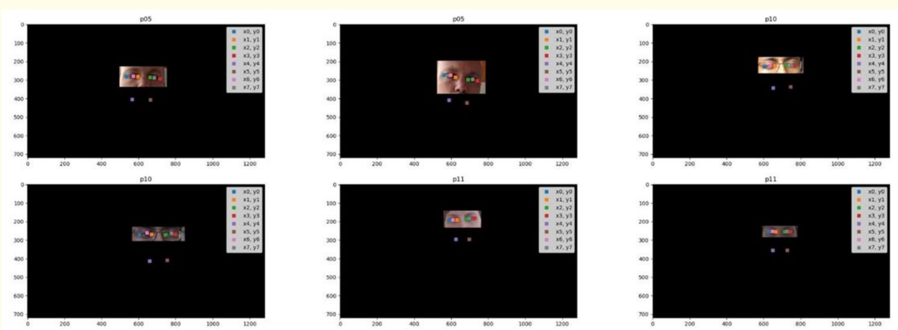

Figure 2: Pupil location extraction.

In this study, the `get_eyeball` custom function is used to extract eye pupil coordinates from the dataset `all_annot_df}. It computes the eyeball image and normalized pupil position for each data row. The extracted coordinates are then added to the data frame in the form of the newly added columns {eyeball{, {pupil_x}, and {pupil_y}, enabling further analysis. Using a visualization technique, (see Fig. 3), a study of the distribution of pupil coordinates and the corresponding eyeball photographs is then conducted. The next subplot grid has histograms illustrating the pupil coordinate distribution along with eyeball images for the lowest, mean, and maximum pupil coordinate values. This technique offers a comprehensive understanding of the variation in pupil positions across the dataset and their relationship with different ocular configurations, which considerably improves insights into gaze estimation research.

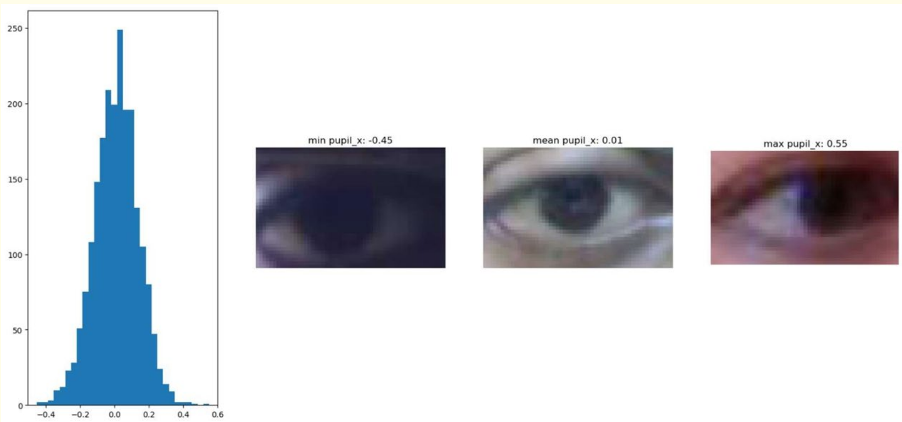

Figure 3: Annotating the pupil position.

The normalized data from a set of '. mat' files are handled methodically in this research work. Every file is parsed to extract the necessary vectors and envision data, and then it is formatted into a data frame for more in-depth analysis. To ensure the integrity of the final dataset, special attention is paid to managing potential mistakes detected during data processing. Important details about the images are contained in the generated Dataframe as shown in Fig. 4., tagged {all_norm_df}, along with related vectors and metadata like group and day identifiers.

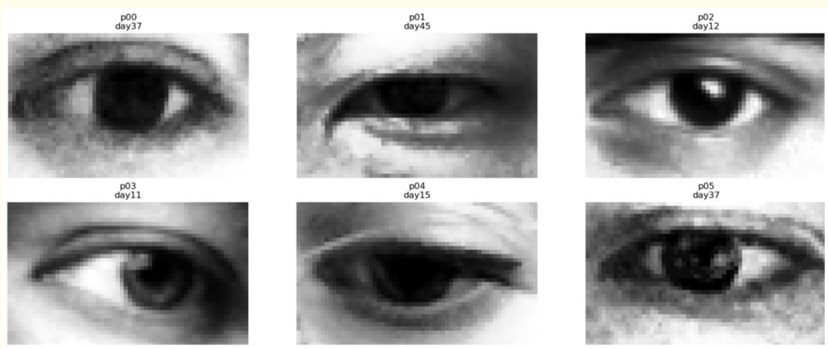

Figure 4: Loading 213656 images from the dataset.

Methodology

Problem Definition

The project aims to develop an eye-controlled mouse application, providing an alternative input method for users to control the computer mouse cursor solely using their eye movements. This addresses the need for accessibility solutions, particularly benefiting individuals with mobility impairments or special needs who may find traditional mouse input challenging.

Library Selection

For implementing the project, several Python libraries such as OpenCV, Mediapipe, PyAutoGUI, and HMM Learn) are chosen based on their functionalities and suitability. OpenCV is selected for its robust capabilities in image processing and computer vision tasks. Mediapipe is chosen for its specialized models, such as FaceMesh, which facilitate facial landmark detection, including eye tracking. PyAutoGUI is utilized for its ability to simulate mouse and keyboard actions, enabling cursor control. Further, HMM Learn is used for hidden Markov model (HMM) implementation in the project, providing tools for sequence prediction and analysis.

Data Acquisition

The research work requires the acquisition of a real-time video feed from a webcam. This involves setting up the webcam hardware and configuring the software to capture video frames continuously to perform the implementation. The webcam feed serves as input data for eye tracking and cursor control based on the eye pupil position.

Eye Tracking Implementation

The Mediapipe FaceMesh model is used to implement eye tracking. To identify facial landmarks, such as the user's eye location, and eye pupil location each frame of the camera video must be processed. In particular, the coordinates of the left eye are taken from the identified landmarks, revealing the direction of the user's look.

Cursor Movement

Based on the model's performance, the identified eye movements are converted into equivalent cursor motions on the computer screen. This involves employing the user's gaze to determine the direction and speed of the cursor motion, as well as mapping the range of observed eye movements to the range of cursor motions.

Smooth Cursor Transitions

To ensure a smooth user experience with an intuitive interface, a parameter is introduced to control the speed of cursor movements. This parameter helps in reducing abrupt cursor jumps and ensuring gradual transitions, enhancing the usability of the eye-controlled mouse application.

Blink Detection

Blink detection functionality is integrated into the application to detect when the user blinks. This is achieved by monitoring changes in the distance between consecutive eye positions. Upon detecting a blink, a predefined action, such as a mouse click event, may be triggered, allowing users to perform selection actions using eye movements.

Testing and Evaluation

The completed eye-controlled mouse application undergoes extensive testing to assess its functionality, accuracy, and usability. Testing involves verifying the performance of eye tracking, cursor control, blink detection, and overall system responsiveness. User feedback may be collected through usability testing to identify areas for improvement and validate the effectiveness of the application in meeting user needs.

Hidden Markov Model

The implementation includes an extensive set of features designed to interpret eye-tracking data and carry out image analysis operations. Fundamentally, the implementation makes use of a variety of image analysis techniques in conjunction with HMMs to analyze patterns of eye movement. The compilation of annotated eye- tracking data to train the HMM model is the first step in the implementation process. The data must be arranged in this preprocessing step so that it can be fed into the HMM for training in the proper sequence. After the data is ready, the Gaussian HMM model—which is good at capturing the underlying patterns within the eye movement sequences—is used to train the HMM using the “hmmlearn” package.

After training the model, the HMM is utilized to forecast hidden states that signify unique eye movement patterns. These anticipated states offer valuable insights into users’ navigation within their visual surroundings, thereby enriching comprehension of human behavior and cognition. Furthermore, amalgamating HMM-based analysis with image processing functions amplifies the versatility and applicability of the method across various domains.

The implementation includes an extensive set of features designed to interpret eye-tracking data and carry out image analysis operations. Fundamentally, the implementation makes use of a variety of image analysis techniques in conjunction with HMMs to analyze patterns of eye movement. The compilation of annotated eye-tracking data to train the HMM model is the first step in the implementation process. After the data is ready, the Gaussian HMM model—which is good at capturing the underlying patterns within the eye movement sequences— is used to train the HMM using the “hmmlearn” package. The use of HMM-based eye movement analysis in the study paper highlights how crucial machine learning methods are for analyzing user behavior. By deciphering the minute details of eye movement patterns, this application greatly improves convenience, medical imaging, and human-computer interaction. The effectiveness of the technique is demonstrated through experimental validation, underscoring the possibility of its practical implementation.

Cursor Control Algorithm

Algorithm: Cursor Control.

Input: Real-time eye movement data of the user.

Output: Cursor movement on the user’s screen.

1. Start.
2. Initialization.
3.   Initialize the FaceMesh model.
4.   Open Webcam.
5.   Set parameters.
6.   `smooth_factor` = 0.5.
7.   `previous_cursor_position` = (0, 0).
8.   `blink_counter` = 0.
9.   `blink_threshold` = 5.
10. Main Loop.
11.   Capture frame from webcam.
12.   Convert frame to RGB.
13.   Detect faces and landmarks.
14.   If faces detected:
15.   Extract left-eye coordinates.
16.   Calculate Cursor Movement.
17.   Determine the center of the left eye.

18. Calculate `cursor_movement` as the difference in X and Y directions.
19. Apply `smooth_factor` to cursor movement.
20. Blink Detection.
21. Measure the distance between consecutive left eye positions.
22. If `distance` < `blink_threshold`:
23. Increment `blink_counter`.
24. Cursor position update.
25. Update `cursor_position` based on calculated movement.
26. Move the cursor using PyAutoGUI.
27. Show processed frame with landmarks and cursor movement.
28. If 'ESC' key is pressed:
29. Exit condition and Cleanup.
30. Release the webcam and close OpenCV windows.
31. End.

Variables

smooth_factor: A parameter to smooth cursor movements. previous_cursor_position: Keeps track of the previous cursor position. blink_counter: Counts the number of blinks.

blink_threshold: The threshold to detect a blink.

cursor_movement: The difference in X and Y directions to move the cursor. cursor_position: The current position of the cursor.

Conditional Statements

If faces detected: Only proceed with eye extraction and further processing if faces are detected.

If distance < blink_threshold: Increment blink counter only if the distance between consecutive eye positions is less than the blink threshold.

If the 'ESC' key is pressed: Exit the loop and perform cleanup operations.

The disclosed cursor control technique tracks eye movements by using blink detection and facial landmark identification to control the cursor (see Fig. 5). First, the webcam is opened and the FaceMesh model is initialized. The main loop moves the pointer by the left eye's movements, continually records frames, and recognizes facial landmarks. To detect blink occurrences, blink detection is used, and the cursor position is adjusted correspondingly. The processed frame is shown on the screen together with the movement of the pointer and facial landmarks. When a key is pressed to break out of the loop, the algorithm ends, releasing the camera and shutting off OpenCV windows.

Results

This innovative cursor control technology offers users a seamless and intuitive interaction experience with computing devices. By enabling real-time adjustment of the cursor through eye movements, users can effortlessly navigate and interact with their devices. The eye pupil positions are extracted using a Hidden Markov Model. Eye-pupil coordinates are initially obtained from images using feature extraction and localization techniques. These coordinates are then fed into an HMM, which leverages the sequential nature of eye movements to predict the next state of the pupil. By analyzing the sequence of pupil positions over time, the HMM infers underlying patterns in eye movements and predicts the most likely future position of the pupil. Subsequently, this predicted position is utilized to control the movement of the cursor on the screen.

A grayscale image, obtained from the dataset, is subjected to blob detection using the Difference of Homogeneous (DoH) method implemented in the `blob_doh` function from the `skimage.feature` module. Blob detection is then performed on the binary mask with specified parameters such as `max_sigma`, `min_sigma`, and `threshold` to identify potential blobs representing the pupil. The resulting blobs are visualized on the grayscale image using a red circle overlay (see Fig. 5), indicating the detected pupil positions. This process enables the identification and localization of the pupil within the image, facilitating further analysis of eye movement behavior. The generated visualization aids in understanding the efficacy of the blob detection method in accurately capturing the pupil's position within the image, contributing valuable insights to the research findings.

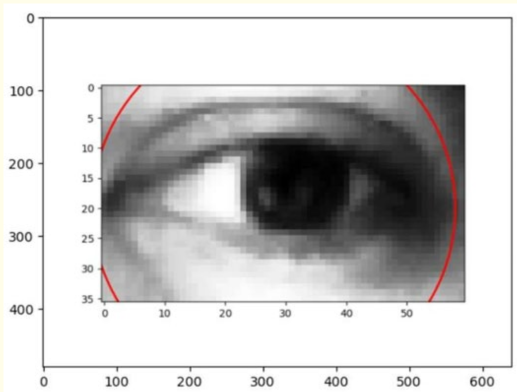

Figure 5: The picture depicts the detected eye pupil.

The built eye-controlled mouse program tracks the user's left eye's movement in real-time using a camera feed by utilizing Python modules like OpenCV, Mediapipe, and PyAutoGUI. With the use of the program, users may operate the mouse cursor with just their eye movements by precisely translating observed eye movements into matching motions on the screen. To minimize abrupt cursor leaps and guarantee smooth cursor transitions, a smooth factor parameter is utilized to improve the user experience. Blink detection functionality is another element of the program that lets it know when the user blinks. When the program detects a blink, it simulates a selection action by launching a mouse click event (see Fig. 6). Overall, the outcome of the implementation is a functional and intuitive eye-controlled mouse application that provides an alternative input method for users, particularly those with mobility impairments or special needs.

```
Cursor Movement (X): 78
Cursor Movement (Y): 71
Eye Distance: 0.0
Blink detected! Performing a select event.
Blink Counter: 1
```

Figure 6: Cursor movement position with blink counter.

Conclusion

This research paper presents a novel method of eye movement-based cursor control, which represents a major advancement in the field of human-computer interaction. The system seamlessly integrates assistive technologies and advanced computer vision algorithms to let people navigate digital interfaces with ease and organicity. Its functionality depends on the accurate recognition of facial landmarks, namely the eyes. The gadget records eye movements in real-time and translates them into cursor movements using strong facial landmark verification algorithms. Enhancing the user experience and facilitating hands-free interaction by utilizing the blink detection algorithm, the system's central component. Through monitoring changes in ocular movements and detecting blinking, the system may trigger specific actions such as mouse clicks or picks, providing users with natural control over their virtual environment. This feature is particularly helpful for those who have trouble with their motor skills since it eliminates the need for physical input devices and meets a variety of user needs and preferences. Through the integration of machine learning techniques based on user behaviors and usage patterns, the system gains the capacity to enhance its functions through user experience modification and adjustment. The research initiative establishes the groundwork for future advancements in assistive technology and accessibility. The project provides new possibilities for accessible design and digital access through a creative investigation into human-computer interaction methodologies and the application of modern technologies such as computer vision and machine learning.

This research work establishes the groundwork for future advancements in assistive technology and accessibility. This model provides new possibilities for digital access through a creative advancement in human-computer interaction methodologies and application of modern technologies such as computer vision and machine learning. As technology continues to evolve, ongoing efforts to refine and expand upon this eye-controlled cursor system will play a crucial role in shaping the future of accessible design and digital interaction.

References

1. Sivasangari A., et al. "Eyeball-based cursor movement control". 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE (2020).
2. Dhanasekar J., et al. "System Cursor Control Using Human Eyeball Movement". 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE (2023).
3. Narahari Dheeraj., et al. "Eyeball Movement-Based Cursor Control for Physically Challenged People. No. 7587". EasyChair (2022).
4. Ganga Vidhat., et al. "Eye ball movement controlled mouse and keyboard using camera". International Journal of Advance Research, Ideas and Innovations in Technology.
5. Chandra B, M Rohit and R Sriram Vignesh. "Eyeball Movement Cursor Control Using OpenCV". ECS Transactions 107.1 (2022): 10005.
6. Mangaiyarkarasi M and A Geetha. "Cursor control system using facial expressions for human-computer interaction". International Journal of Emerging Technology in Computer Science & Electronics 8.1 (2014): 30-34.
7. Sabab Shahed Anzarus., et al. "Eye Pointer: A real time cost effective computer controlling system using eye and head movement". no. c (2016): 153-159.
8. Ray Anshu Kumar., et al. "Cursor Motion Control Using Eye Tracking and Computer Vision". International Conference on Advanced Communication and Intelligent Systems. Cham: Springer Nature Switzerland (2022).
9. Narendra, Chilukuri Sai Durga, and BN Srinivasa Gupta. "Eyeball Movement Based Cursor Control".
10. Pradhan Kanchan., et al. "Eyeball Movement based Cursor Using Machine Learning". International Research Journal of Modernization in Engineering Technology and Science 5 (2023): 362-368.
11. Kumar Kiran Nagabhushan. "Analyzing cursor movements with an HMM to assess individual differences in cognition reliably and quickly". Indiana University (2020).
12. Salunkhe Prashant and Ashwini R Patil. "A review on device controlled using eye movement". International Journal of Emerging Trends in Science and Technology 2.1 (2015): 1773-1778.

13. Kim Jaeah., et al. “A hidden Markov model for analyzing eye-tracking of moving objects: Case study in a sustained attention paradigm”. Behavior research methods 52 (2020): 1225-1243.
14. Feng Gary. “Eye movements as time-series random variables: A stochastic model of eye movement control in reading”. Cognitive Systems Research 7.1 (2006): 70-95.
15. Aziz Md Fayeem. “Applications of Hidden Markov Model and Support Vector Machine for State Estimation”. University of Malaya (Malaysia) (2014).
16. Hsiao Janet H., et al. “Eye Movement analysis with Hidden Markov Models (EMHMM) with co- clustering”. Behavior Research Methods 53.6 (2021): 2473-2486.
17. Alhamzawi Hussein Ali. “Control mouse cursor by head movement: Development and implementation”. Applied Medical Informatics 40.3/4 (2018): 39-44.
18. Madhuri K and L Praveen Kumar. “Cursor Movements Controlled by Real Time Hand Gestures”. International Journal of Science and Research 2.2 (2013).
19. Phillips James G and Thomas J Triggs. “Characteristics of cursor trajectories controlled by the computer mouse”. Ergonomics 44.5 (2001): 527-536.
20. Sharanyaa S and Madhumitha RP. “Eyeball Cursor Movement Detection Using Deep Learning”. Proceedings of the International Conference on Innovative Computing & Communication (ICICC) (2021).
21. Lakshmy TS Swathy, Aswathi Sivadasan and L Nitha. “Detection of Fraudsters in Electronic Auction Using Hidden Markov Model with X-Means Clustering Algorithm”. International Journal of Pure and Applied Mathematics 114.11 (2017): 157-165.
22. Ashwini K, R Ponuma and R Amutha. “Fine motor skills and cognitive development using virtual reality-based games in children”. Handbook of Decision Support Systems for Neurological Disorders. Academic Press (2021): 187-201.
23. Sachin Krishnan P, K Rameshkumar and P Krishnakumar. “Hidden Markov modeling of high-speed milling (HSM) process using acoustic emission (AE) signature for predicting tool conditions”. Advances in Materials and Manufacturing Engineering: Proceedings of ICAMME 2019. Springer Singapore (2020).
24. Havirbhavi P, G Tanusha and K Ashwini. “Controlling of Cursor movement with Eyeball using Hidden Markov Model”. 2024 3rd International Conference for Innovation in Technology (INOCON). IEEE (2024).