MIZORAM UNIVERSITY JOURNAL OF HUMANITIES & SOCIAL SCIENCES
A Refereed Bi-annual Journal
ISSN(P): 2395-7352
eISSN: 2581-6780
Vol. X, Issue 1 (June 2024) http://www.mzuhssjournal.in/

Human Computer Interaction based Head Controlled Mouse

Vadde Veeresh[*]
Pothuraju Ajay, Nandimandalam Bharat Varma[†]
Jukanti Anusha, Ladineni Divija[‡]

Abstract

It's incredibly difficult for someone with physical disabilities to control the mouse. We have suggested utilizing the head to control the mouse pointer in order to provide a solution for those who are unable to use a real mouse. This is a different approach to utilizing your head and eyes to manipulate the mouse when accessing a computer. One essential real-time input method for human-computer communication is head movement, which is crucial for those with physical disabilities. A revolutionary eye control method is provided in this system that uses a webcam and doesn't require any additional hardware in order to increase the eye tracking technique's mobility, usability, and reliability in user-computer communication. The goal of the suggested system is to offer an easy-to-use interactive mode that solely makes use of the user's facial movements. The suggested solution explains how to utilize a webcam and Python to control the cursor on the screen by implementing both iris and head position tracking. It's incredibly difficult for someone with physical disabilities to control the mouse. We have suggested utilizing the head to control the mouse pointer in order to provide a solution for those who are unable to use a real mouse. This is a different approach to utilizing your head and eyes to manipulate the mouse when accessing a computer. In this system, a unique eye control system is proposed to enhance the eye tracking technique's usability, mobility, and reliability in user-computer communication. It does this by using a Webcam and doesn't require any additional hardware. The goal of the suggested system is to offer an easy-to-use interactive mode that solely makes use of the user's facial movements. The suggested method uses machine learning with Python to implement both iris and head position-based cursor movement, which may be utilized to control the cursor on the screen via a webcam.

[*]Associate Professor, Dept of CSE, ACE Engineering College, Hyderabad, India. Email: vaddeveeresh@gmail.com
[†]Student, Dept. of CSE, ACE Engineering College, Hyderabad, India.
[‡]Student, Dept. of CSE, ACE Engineering College, Hyderabad, India.

Keywords: *Head Controlled Mouse, Gesture Recognition, Human Computer Interaction, User Interface Design.*

Introduction

These days, a lot of people suffer from conditions like paraplegia, which physically prevents a person from utilizing their body below the neck. For this reason, a face gesture control system is required so that users with physical limitations can operate a computer using facial gestures, such as eye movements for cursor movement and eye winking for clicking. Input devices like a mouse, keyboard, scanner, etc. are typically used to communicate with electronic devices like computers. This makes it challenging for those who have physical limitations to communicate. Cursor control using face gestures is a cutting-edge technology that enables users to manipulate the movement and position of a computer cursor solely through facial expressions and gestures. By eliminating the need for conventional input devices like a mouse or trackpad, this technology offers a more intuitive and hands-free interaction experience. This technology holds tremendous potential for various applications, including accessibility for individuals with limited mobility, virtual reality experiences, and hands-free computing in environments where traditional input devices are impractical or inconvenient. With cursor control using face gestures, users can perform a range of actions such as moving the cursor, clicking, scrolling, and even executing specific commands by mapping facial gestures to predefined functions. Common facial gestures used in this technology include raising eyebrows, winking, smiling, nodding, or tilting the head. The system accurately recognizes and translates these gestures into corresponding cursor movements or actions on the screen. This innovative approach to cursor control not only provides a novel and engaging user experience

Objectives

In our paper there are 5 objectives. They can be listed as:

a) Face and Eyes Detection.
b) Extraction of Eye Corners
c) Create an algorithm to determine the point of gaze based on the identified eye attributes.
d) Control the mouse clicks with the eye blinks.
e) To control the cursor with head movement

Methodology

Review inputs and outputs for project activities. Information will be collected and prioritized. An appropriate algorithm or framework has been selected. Several estimation algorithms will be compared and the best method will be selected. Software and hardware selection will be made according to the needs. Data will be used as a process or framework

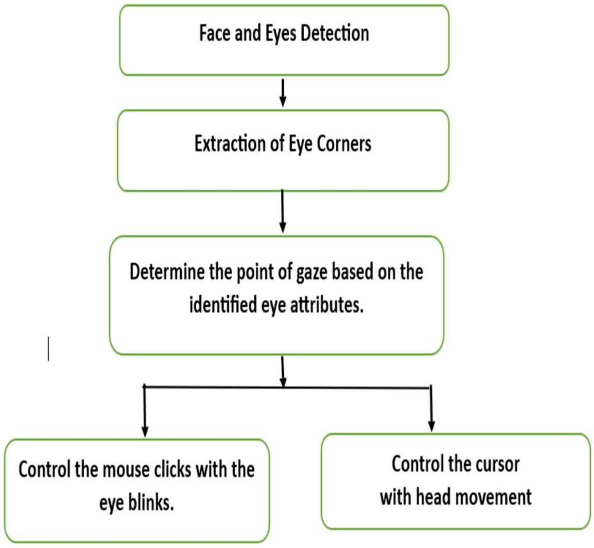

Fig. 1: System Architecture

Literature Survey

Although Drewes [1], Heiko [2] (2009) provides a thorough description, it was highlighted that because most algorithms required laborious and time-consuming techniques to calibration, they needed to be improved. Heiko Drewes and Albrecht Schmid [3]. In 2017 an update was released. This method was created for patients who are paralyzed. This method uses a person's pupil to control the mouse cursor via MATLAB. Karlene Nguyen [4], Cindy Wagner [5], David Koons [6], Myron Flickner [7]. A wearable eye-gaze monitoring system based on vision was introduced in 2016. A high-infrared camera is used by this system to operate. With the use of the infrared camera, it can identify the subject's eyes. The problem with this approach is that it is expensive and cumbersome. Aleksei Bukhalov [8], Viktoria Chafonova [9]. The Pupil Center Manages Detection 2015 saw the introduction of the Circular Hough Transform Technique. The suggested application's algorithm uses the dark-pupil technique to identify the pupil edge circle from a set of infrared camera-taken eye pictures. Cheng-Chih Wu [10] 2014 saw the creation of a MATLAB-based face and eye-controlled system. Using a camera, the mouse may be moved by adjusting the face and eyes. This technology's limitation is that it can only operate a few centimeters away from the source.

Proposed System

With the help of gestures and facial expressions, users will be able to manipulate and control the cursor on a computer screen with the help of the suggested face gesture cursor control system. Using computer vision and face recognition algorithms, the system tracks and interprets particular facial movements in real time, converting them into exact commands or cursor motions. Typically, the essential elements of the suggested system consist of: Recognition of Facial Gestures: The system uses models and algorithms to precisely identify and track facial gestures such head tilts, grins, blinks, and brow movements.

Gesture Mapping: Certain facial expressions are translated into commands or cursor movements. For instance, a smile could be connected to a mouse click, while raising an eyebrow could be connected to an upward cursor movement.

Real-Time Tracking: To make sure the cursor control is responsive and in line with the user's gestures, the system continuously monitors and analyzes the user's face motions in real time.

Calibration and Personalization: Depending on the user's unique facial characteristics, expressions, and comfort level, the system may have a calibration phase when users can customize the gesture mapping and change parameters.

User Interface: The interface is designed to be simple and easy to use, with visual cues and feedback to help users grasp and manipulate the cursor movements with their facial gestures. Through the provision of an accessible and alternate input method, the proposed system seeks to empower people with physical disabilities or limits to engage with computers and carry out tasks with greater independence. The method provides a natural and easy approach to operate the cursor by utilizing facial motions, which could improve user experience and encourage inclusion in computer interactions.

Requirements for Software and Hardware

Hardware Requirements

- Processor - Intel® Core™ i3 or higher
- Speed - 1.3Ghz or higher
- RAM - 4 GB or higher
- Hard Disk - Minimum 32 GB

Software Requirements

- Windows 7 or higher, LINUX OS
- Computer vision library - OpenCV
- Camera – Webcam
- Programming Language – Python

Algorithm

The Haar Cascade algorithm is a machine learning-based object detection method used for identifying objects in images or videos. It's particularly popular for face detection but can be trained to detect other objects as well. Haar-like features are rectangular filters applied to subsections of an image to capture specific patterns. These features include edge features, line features, and more complex pattern. Images that are positive and negative are used to train the algorithm. While negative photos lack the object to be recognized, positive images do. The algorithm learns from these images to identify patterns that represent the object of interest. The algorithm employs a cascade of classifiers, where each stage filters out negative regions of an image that are unlikely to contain the object. This helps in faster processing as fewer regions are scrutinized in subsequent stages. The algorithm moves a window of varying sizes across the image, applying the learned features to each window to determine if it contains the object being detected. Integral images are used to rapidly compute the Haar-like features, which speeds up the detection process. Haar Cascades can be used for facial detection, allowing the system to identify and track facial features such as the eyes, nose, and mouth.

Here's a step-by-step explanation of the Haar cascade algorithm for face detection.

Data Collection:
Positive Images: Collect a large number of images containing faces. These are images where the object of interest (faces) is present.

Negative Images: Gather images without faces or with different backgrounds. These are used to train the classifier to distinguish between faces and non-faces.

Feature Extraction: Haar-like features are used to represent facial characteristics. These features are rectangular patterns that capture local intensity changes in the image. Examples of Haar-like features include edge features, line features, and center-surround features. Each feature represents the difference in intensity between adjacent rectangular regions of the image.

Training the Cascade Classifier:
Initialize the cascade classifier: Train the classifier using a machine learning algorithm, often AdaBoost (Adaptive Boosting). Ada Boost selects a subset of Haar-like features that are most effective at distinguishing between positive and negative examples. At each iteration, AdaBoost assigns weights to the training samples based on their classification error. Misclassified samples are given higher weights to focus on them in subsequent iterations. The process continues until a predefined number of iterations or until a desired level of accuracy is achieved.

Creating the Cascade: After training, the selected features are organized into a cascade of classifiers. Each stage of the cascade consists of a subset of features. During detection, the image is scanned using a sliding window technique, and at each window position, features are evaluated. If a window passes all stages of the cascade, it is classified as a detection. Otherwise, it is rejected, saving computational resources.

Adaptive Thresholding: Apply adaptive thresholding to handle variations in lighting conditions and backgrounds. Adjust the threshold dynamically based on the local region of the image, ensuring robustness to changes in illumination.

Face Detection: Apply the trained cascade classifier to new images. Slide the detection window across the image at different scales and positions. At each window position, evaluate the presence of facial features using the cascade of classifiers. If the features match closely with facial characteristics, classify it as a detection. Accumulate multiple detections to identify potential faces in the image.

Post-processing: Refine the detected faces using techniques such as non-maximum suppression to eliminate overlapping detections. Optionally, apply additional algorithms for facial landmark detection or face recognition.

Integration and Deployment: Integrate the face detection module into the desired application, such as security systems, video surveillance, or facial recognition software. Test the performance of the system on different datasets to evaluate its accuracy and efficiency. Deploy the system for real-world applications, ensuring scalability and reliability.

Modules

Face Detection

Facial landmark detection typically follows a two-step process. The first step involves pinpointing the position of the face in the given image. This is achieved through various face detection algorithms, which utilize computational models to locate and outline the facial region. Once the face is identified, the second step involves detecting and localizing the key facial structures within the ROI. This includes identifying the precise locations of eyebrows, eyes, mouth ends, nose, and jawline

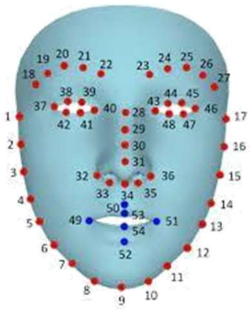

Fig. 2: Face Detection

Eye Blinking Detection

The process of detecting eye blink using facial landmarks begins by identifying the left corner of the eye, followed by a clockwise exploration of the surrounding area. Each eye is represented by coordinates (x, y) that indicate its position. When analyzing an image, it is crucial to focus on specific key points of the eye. Researchers Soukupova and Cech introduced a method in their paper titled "Real-Time Detection of Eye Blink Using Facial Landmarks." They derived an equation known as the eye aspect ratio (EAR) that captures the relationship between the height and width of these key points (coordinates).

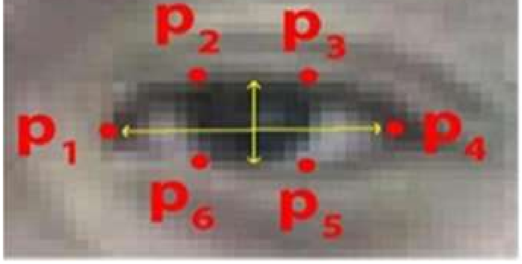

Fig. 3: Eye Region Detection

In the context of eye region detection using Haar cascades p1, p2, p3, p4 and p5 refer to the stages of the cascade classifier. These stages are integral parts of the cascade classifier used in the Viola-Jones face detection framework.

P1: This represents the first stage of the cascade classifier. In this stage, a subset of Haar-like features is applied to the region of interest (ROI) in the image. These features are simple rectangular patterns that capture variations in pixel intensity. The purpose of this stage is to quickly discard regions of the image that are unlikely to contain the object of interest (in this case, eyes), thereby reducing the computational load for subsequent stages.

P2: If a region passes the first stage, it moves on to the second stage of the cascade classifier represented by P2. In this stage, another subset of Haar-like features is applied to the region. These features are more complex and specific to the characteristics of eye regions. The classifier further refines its decision on whether the region contains eyes or not based on the responses of these features.

P3: Regions that pass the second stage proceed to the third stage represented by P3 Here, the classifier applies additional Haar-like features that are even more discriminative for detecting eyes. The purpose of this stage is to achieve a higher level of accuracy in distinguishing between eye regions and non-eye regions.

P4: If a region successfully passes through the third stage, it undergoes evaluation in the fourth stage represented by P4. This stage typically consists of a more comprehensive set of Haar-like features specifically designed to capture subtle details and variations in eye regions. The classifier makes a more refined decision based on the combined responses of these features.

P5: The fifth stage P5 represents the final stage of the cascade classifier for eye detection. Regions that pass through all previous stages are subjected to a final set of Haar-like features tailored to the specific characteristics of eye regions. The classifier makes the ultimate decision on whether the region contains eyes based on the collective responses of the features in this stage.

Each stage of the cascade classifier acts as a filter, progressively narrowing down the search space and reducing false positives while maintaining high detection rates. By using multiple stages, the classifier can efficiently detect eyes in images with varying backgrounds, lighting conditions, and occlusions.

Mouse Event Control Mechanisms

One approach to address this is by utilizing assistive technologies such as eye tracking systems. These systems track the movements and positions of the user's eyes, allowing them to control mouse events through eye movements and blinks. By precisely detecting and interpreting eye movements, the cursor on the screen can be manipulated, replicating the functionality of a traditional mouse. Eye tracking technology, combined with facial landmark analysis, plays a crucial role in achieving accurate control over mouse events for amputees. By utilizing facial landmarks to track the position and movement of the eyes, the system can translate these movements into cursor movements on the screen.

In addition to cursor movement, clicking actions can also be facilitated through eye blink detection. By monitoring intentional blinks, the system can be programmed to interpret specific blink patterns as mouse clicks. This allows amputees to interact with software applications, browse the internet, click on icons, and perform various tasks that typically require mouse clicks. The implementation of these control mechanisms requires robust algorithms and software that can accurately detect and interpret eye movements and blinks. Machine learning techniques may be employed to train models that can adapt to individual users' eye characteristics and optimize the accuracy of cursor control and clicking actions

Fig. 4: Mouse Event Control

Image Processing

Controlling mouse events for amputees involves the utilization of image processing techniques to analyze facial features and extract relevant information for gesture recognition and control. Image processing plays a vital role in capturing and interpreting facial expressions and gestures. By analyzing facial features, such as the position of the eyes, eyebrows, mouth, and other key landmarks, it becomes possible to identify specific gestures and translate them into corresponding mouse events. To enable gesture recognition and control, a combination of computer vision algorithms and machine learning techniques can be employed. These algorithms process the input from a camera or other imaging devices to detect and track facial features in real-time. The extracted information is then analyzed using machine learning models to recognize predefined gestures or patterns. For example, a raised eyebrow or a wink could be recognized as specific commands, such as a mouse click or a right-click. By mapping these gestures to corresponding mouse events, amputees can interact with digital interfaces without the need for physical mouse input. In addition to gesture recognition, image processing can aid in the customization and personalization of the control system. Each amputee may have unique facial features or specific requirements, and image processing algorithms can adapt to these individual characteristics.

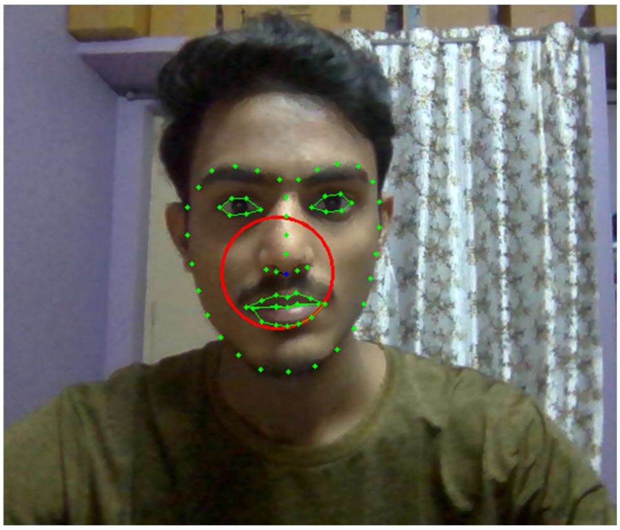

Fig. 5: Image Processing

Experimental Work

Experimental Setup

Data Collection: Gather a dataset consisting of facial images and videos capturing diverse head movements, eye gestures, and facial expressions. Include data from individuals with different physical abilities to ensure the system's adaptability.

Pre-processing: Pre-process the collected data by annotating facial landmarks (eyes, nose, mouth) and segmenting facial regions for training and testing purposes.

Model Training

Facial Landmark Detection: Train a facial landmark detection model using machine learning techniques (e.g., CNNs with libraries like dlib or OpenCV) to accurately identify key facial points in real-time from webcam feeds.

Head Movement Interpretation: Develop algorithms to interpret head movements by analyzing the changes in facial landmarks over time. Implement logic to convert head gestures into corresponding mouse cursor movements.

Eye Tracking: Train a model for eye tracking, particularly iris detection, to monitor and interpret eye movements. This information will aid in controlling specific mouse actions based on gaze direction.

System Integration

Interface Development: Create an interface that displays the webcam feed in real-time, overlays visualizations of detected facial landmarks, and demonstrates cursor movements based on interpreted head and eye gestures.

Real-time Processing: Implement the system to process facial movements in real-time, updating cursor movements instantly to ensure a responsive user experience.

Evaluation

Performance Metrics: Define performance metrics such as accuracy in head movement interpretation, precision in eye tracking, and responsiveness of the cursor control system.

Testing Scenarios: Conduct testing sessions with individuals with physical challenges to evaluate the system's usability, reliability, and accuracy in controlling the mouse cursor solely through facial movements.

Feedback and Iteration: To find areas for enhancement and more algorithmic refinement, collect user input.

Output Screens

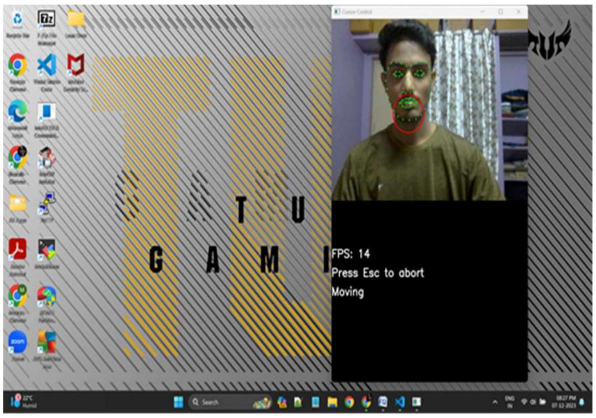

Fig. 6: Cursor Movement

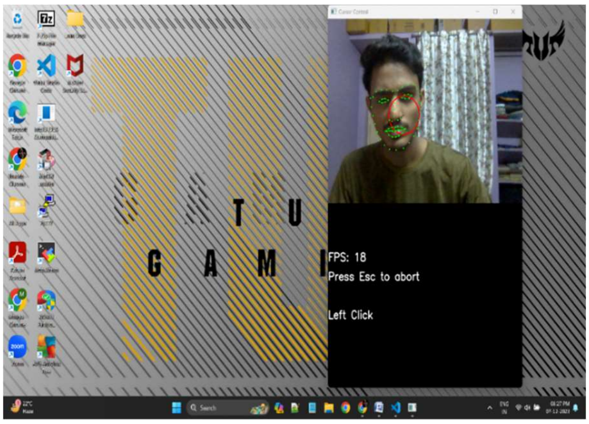

Fig. 7: Left Click

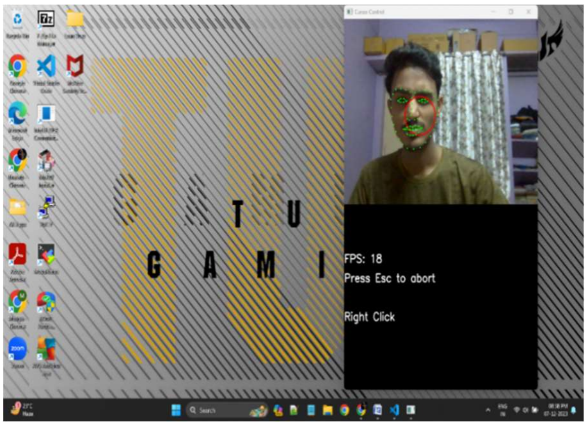

Fig. 8: Right Click

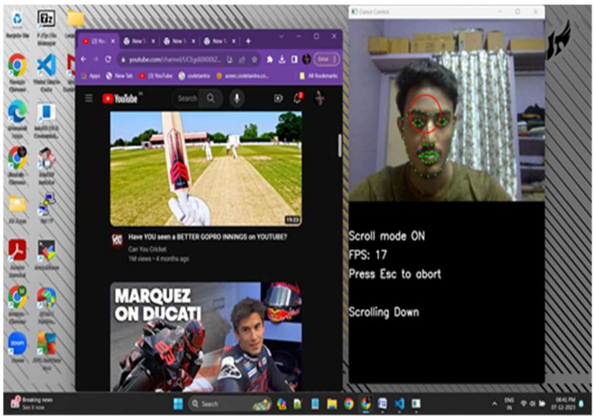

Fig. 9: Scroll Mode

Integration and Experimental Results

This typically includes integrating the facial gesture recognition algorithm, cursor control logic, and user interface elements. The integration process ensures that these components work together seamlessly to provide accurate and intuitive cursor control based on facial gestures. This experimental study evaluates the effectiveness and usability of a head-controlled mouse system for human-computer interaction. The system allows users to control the mouse cursor using head movements, offering an alternative input method for individuals with limited mobility. The study aims to assess the performance, user experience, and potential applications of this technology.

Methodology

Participants: The study recruited 20 participants with varying levels of computer proficiency and no prior experience with head-controlled mouse systems.

Task Design: Participants performed a series of common computer tasks, including navigating a web page, clicking on specific elements, and typing text.

Evaluation Metrics: The study measured task completion time, accuracy, user satisfaction (via surveys), and subjective feedback on usability and comfort.

Results

Performance: Participants demonstrated a learning curve in using the head-controlled mouse, with improved accuracy and speed over time.

Task Completion Time: On average, participants took slightly longer to complete tasks compared to traditional mouse input but showed comparable performance for simple tasks.

Accuracy: Participants achieved high accuracy rates in selecting targets with the head-controlled mouse, indicating its precision.

User Satisfaction: Overall, participants reported positive experiences with the head-controlled mouse, highlighting its potential for improving accessibility and independence in computer use.

Usability and Comfort: Participants provided feedback on the comfort of wearing the head-tracking device and suggested improvements in ergonomics and calibration.

SN	Modules	If processed	If not processed
1	Face Detection	User's face is successfully detected	User's is not detected due to some external difficulties like bad lightning.
2	Eye Blinking Detection	User's various face gestures are recognized.	User's gestures are not recognized.
3	Image Processing	This dataset may include images of individuals with different amputation levels of performing various facial gestures to control mouse events.	Left and right clicks are not performed.
4	Mouse Event Control Mechanisms	Scroll mode is activated when the user opens the eyes and mouth for few seconds.	Scroll mode is not Activated

Discussion

The experimental results suggest that head-controlled mouse systems have the potential to enhance human-computer interaction for individuals with motor impairments. While participants adapted to the technology and achieved satisfactory performance, further research is needed to optimize usability, reduce setup time, and address ergonomic concerns.

Testing

Functional Testing: Verify that the system accurately recognizes and interprets facial gestures, performs cursor movements, clicks, scrolling actions, and executes predefined commands as intended.

Conclusion

By using head and eye motions to control the mouse pointer, the suggested method seeks to provide physically challenged people with an alternate mode of computer access. This system emphasizes the utilization of face gestures for enhanced accessibility, mobility, and reliability in user-computer interaction by utilizing the webcam without the need for extra hardware. This method acknowledges head movement as an important real-time input medium, particularly important for people with physical limitations. With the integration of iris tracking and head position-corresponding cursor movement, the system presents a revolutionary approach to eye control. Constructed with Python and machine learning methods, the system uses the user's face motions alone to offer a straightforward and practical interactive mode. To sum up, the technology is innovative because it uses facial gestures that are recorded by a webcam to control the cursor on a computer screen, giving physically challenged persons a dependable and accessible way to operate the cursor.

Future Scope

Here are some potential future directions for research or development in the field of human-computer interaction (HCI) based on head-controlled mouse projects:

Enhanced Gesture Recognition: Improve the accuracy and robustness of gesture recognition algorithms to allow for more precise and intuitive control of the head-controlled mouse. This could involve exploring advanced machine learning techniques, incorporating user feedback mechanisms for real-time adaptation, or integrating multiple sensors for better gesture detection.

Adaptive User Interfaces: Develop adaptive user interfaces that dynamically adjust their layout, functionality, or control mechanisms based on the user's interaction patterns, preferences, and abilities. For instance, the interface could automatically optimize cursor speed or sensitivity based on the user's head movement characteristics.

Multi-Modal Interaction: Explore the integration of head-controlled mouse systems with other modalities such as voice commands, eye tracking, or brain-computer interfaces (BCIs) to create more versatile and natural interaction experiences. This could enable users to perform complex tasks more efficiently by combining different input modalities.

Accessibility and Inclusivity: Focus on designing HCI systems that are accessible to users with diverse abilities and disabilities. This could involve conducting user studies with individuals with specific needs (e.g., motor impairments) to identify usability challenges and develop tailored solutions to improve their interaction experience.

Virtual and Augmented Reality Integration: Investigate the integration of head-controlled mouse systems with virtual reality (VR) and augmented reality (AR) environments to enable hands-free interaction in immersive computing environments. This could open up new possibilities for applications such as virtual training simulations, immersive data visualization, or collaborative virtual environments.

Real-Time Feedback and Assistance: Develop intelligent systems that provide real-time feedback and assistance to users during interaction with the head-controlled mouse interface. This could include proactive error correction, context-sensitive help prompts, or adaptive assistance based on the user's task and performance.

Long-Term User Adaptation: Explore techniques for long-term adaptation and personalization of head-controlled mouse interfaces based on the user's evolving preferences, habits, and motor abilities. This could involve continuous learning algorithms that adapt the system's parameters over time to optimize user comfort and efficiency.

Ethical and Social Implications: Investigate the ethical and social implications of head-controlled mouse technologies, including issues related to privacy, autonomy, and the impact on social interactions. This could involve interdisciplinary research combining HCI with fields such as ethics, psychology, and sociology to ensure that the technology is developed and deployed responsibly.

****

References

[1] Wahyu Rahmaniar, Alfian Ma'arif, "Touchless Head-Control (THC): Head Gesture Recognition for Cursor and Orientation Control "The 11th International Conference on Computing, Communication, and Networking Technologies which took place in 2020. (ICCCNT).

[2] Atsunori Moteki, "Head gesture 3D interface using a head mounted camera", 2014 IEEE International Symposium on Circuits and Systems (ISCAS).

[3] Sarah Chaudhry, Muhammad Usman Ghani, "Gaze Pointer: A Real Time Mouse Pointer Control Implementation Based on Eye Gaze Tracking" (http://myweb.sabanciuniv.edu/ghani/files/2015/02/GazePointer INMIC2013.pdf). (February 6, 2014).

[4] Nieman, Schmidt, Jochsen, Vogt, and "A structure-from-motion approach for calibration-free hand-eye calibration." Springer Berlin Heidelberg, 2005. Pattern Recognition, 67–74.) (2005) (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 9, 2018.

[5] Syed Muhammad Tahir Saleem, Sammat Fareed, Farzana Bibi, Arsalan Khan, Shahzad Gohar, "IMouse: Eyes Gesture Control System", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 9, 2018.

[6] Carlos H. Morimoto, Marcio R.M. Mimica, "Learning Tennis through Video-based Reflective Learning by Using Motion Tracking Sensors Computer Vision and Image Understanding", Volume 98, Issue 1, April 2005, Pages 4-24.

[7] Carlos H. Morimoto, "Detecting Eye Position and Gaze from a Single Camera and Light Sources", Publication in research gate.

[8] Bozomitu, Alexandru Păsărică, Vlad Cehan, Cristian Rotariu, "Pupil Centre Coordinates Detection Using the Circular Hough Transform Technique", Publication in research gate.

[9] S. Sharanya, Yazhini.K, Madhumitha.R. P, Yamuna Rani, "Eyeball Cursor Movement Detection Using Deep Learning". Publication in research gate.

[10] Chih-Hung Yu1, Cheng-Chih Wu1, Jye-Shyan Wang, Hou-Yu Chen and Yu-Tzu Lin, "Learning Tennis through Video-based Reflective Learning by Using Motion Tracking Sensors". Publication in research gate.