

Eye Controlled Mouse Cursor for Physically Disabled Individual Using CNN

M. RAJAMANI¹, M BALAMURUGAN², R SURYA³
^{1, 2, 3} Kalasalingam Academy of Research and Education

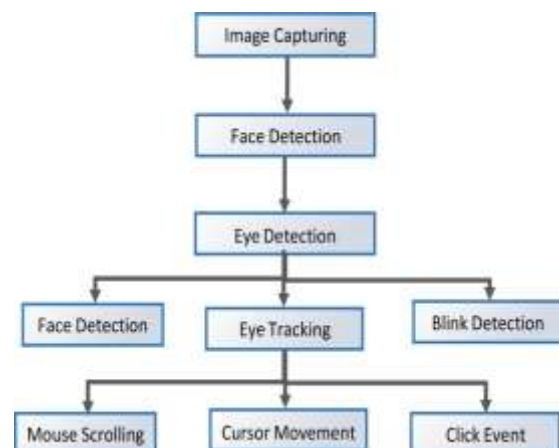
Abstract- This paper presents an eye-controlled mouse cursor system designed to assist physically disabled individuals in interacting with computers using natural facial gestures. The proposed system utilizes a standard webcam to capture real-time video input and employs Convolutional Neural Networks (CNNs) for accurate detection and classification of facial gestures. Eye movements are tracked to control cursor position, while specific gestures—including deliberate single eye blink, mouth opening, and both eyes closed—are mapped to left-click, right-click, and click-and-drag actions respectively. The system integrates multiple modules: face detection using pre-trained models, eye and mouth region extraction through facial landmarks, CNN-based state classification for eyes and mouth, and gaze tracking for cursor movement. Temporal filters and confirmation mechanisms are incorporated to distinguish intentional gestures from involuntary actions such as natural blinking or speaking, thereby minimizing false triggers. The system operates in real-time under varying lighting conditions and head orientations without requiring specialized hardware or physical attachments. Experimental results demonstrate high accuracy in gesture recognition and smooth cursor control, offering an affordable, non-invasive, and intuitive assistive technology solution. This work contributes to enhancing digital accessibility and independence for individuals with motor impairments, promoting greater inclusion in education, employment, and daily computer-based activities.

Index Terms- Eye-Controlled Mouse, Assistive Technology, Convolutional Neural Networks, Facial Gesture Recognition, Human-Computer Interaction, Computer Vision, Physically Disabled Individuals

I. INTRODUCTION

The rapid advancement of human-computer interaction (HCI) technologies has transformed the way individuals engage with digital systems, enabling seamless communication, productivity, and entertainment. However, for individuals with severe physical disabilities—such as those suffering from

spinal cord injuries, muscular dystrophy, cerebral palsy, or amyotrophic lateral sclerosis (ALS)—traditional input devices like keyboards and mice remain largely inaccessible due to the requirement of fine motor skills. This digital divide limits their ability to perform essential tasks such as communicating via email, pursuing online education, securing employment, or accessing healthcare services independently. While various assistive technologies have been developed to address this challenge, many rely on expensive specialized hardware, invasive sensors, or complex setup procedures that are impractical for everyday use. Furthermore, existing vision-based solutions often suffer from poor accuracy under varying lighting conditions, latency issues, and false triggers caused by involuntary gestures, reducing user confidence and adoption. These limitations highlight the urgent need for an affordable, accurate, and non-invasive hands-free interface that can reliably interpret natural facial gestures for computer control.



This paper proposes an eye-controlled mouse cursor system that leverages Convolutional Neural Networks (CNNs) and computer vision techniques to provide intuitive and responsive computer access for physically disabled individuals. The system utilizes a standard webcam to capture real-time video input and employs deep learning models to detect and classify facial gestures including eye movements, blinking patterns, and mouth openings. Eye gaze direction is continuously tracked to control cursor movement across the screen, while specific gestures are mapped to mouse actions: a deliberate single-eye blink triggers a left-click, mouth opening initiates a right-click, and closing both eyes simultaneously enables click-and-drag functionality. Temporal filters and confirmation mechanisms are integrated to distinguish intentional gestures from involuntary actions such as natural blinking or speaking, ensuring reliable performance in real-world conditions. The system is designed to be affordable, requiring no additional hardware, and operates in real-time with minimal latency. By providing an accessible, non-invasive, and accurate hands-free interface, this work aims to empower individuals with motor impairments to independently navigate digital environments, thereby promoting greater social inclusion, educational opportunities, and quality of life.

II. RELATED WORKS

"Mouse Cursor Movement and Control using Eye Gaze- A Human Computer Interaction" by Miah et al. (International Conference on Artificial Intelligence of Things, 2023): This research presents an eye-gaze-based system for mouse cursor control using a standard webcam. The authors employ the MediaPipe library to compute the Eye Aspect Ratio (EAR) for blink detection, enabling cursor movement and click operations. The study focuses on assisting physically disabled individuals who face challenges using traditional computer inputs. The approach demonstrates how open-source computer vision libraries can be effectively utilized for real-time gaze tracking, with the primary objective of improving the computing experience for people with crippled limbs by helping them overcome mouse-related accessibility barriers.

"Call with eyes: A robust interface based on ANN to assist people with locked-in syndrome" by Beltrán-Vargas et al. (SoftwareX, Elsevier, 2024): This work presents a Python-based interface specifically designed for individuals with locked-in syndrome (LIS). The system utilizes a face mesh library to capture 30 facial points around the eyes and calculates 54 Euclidean distances to track iris positions. An artificial neural network (ANN) trained on data from 10 individuals classifies four gestures: vertical eye movements (looking up/down), closed eyes, and open eyes, achieving 99.8% accuracy. The interface enables users to navigate a vertical menu with five options using eye movements and select options by closing their eyes. The system demonstrates superior performance with the key advantage of requiring no calibration for new users while maintaining robust performance across varying illumination and background conditions.

"Ely-Sean-Eye-Mouse-Control-CNN: Implementing an eye-mouse control system using neural networks" by Hahami and Wu (GitHub/STEMist Hacks, 2023): This project implements an eye-mouse control system using three distinct CNN models trained on the Columbia Gaze Dataset. The first model classifies head position (five angles from -30 to +30 degrees), while vertical and horizontal gaze prediction CNNs estimate eye direction with 89% and 85% accuracy respectively. The system captures webcam images every 0.5 seconds, feeds them into all three models, and uses the combined predictions to move the cursor via the pyautogui library. Head tilts to the left or right trigger click operations, providing a complete hands-free interface. The work demonstrates the practical application of CNNs for assistive technology, specifically targeting individuals with physical limitations, neurological conditions, or repetitive strain injuries that prevent traditional mouse usage.

"Real-Time Multi-Modal Sign Language Recognition and Emotion Interpretation System" (IEEE Conference, 2025): This study presents a multi-modal assistive system that combines sign language gesture recognition with facial emotion analysis using open-source tools including TensorFlow, MediaPipe, and OpenCV. The system employs convolutional neural networks (CNNs) for gesture detection and facial landmark-based techniques for emotion

classification, achieving reliable accuracy with low processing latency. Designed for web compatibility as a browser plugin or Chrome extension, the lightweight platform-agnostic approach enables seamless integration into online environments. While focused on sign language interpretation, the work demonstrates the effectiveness of combining CNN-based gesture recognition with facial analysis for accessibility applications, validating the approach for real-time assistive technologies.

"Eyeball Cursor Movement Detection Using Deep Learning" by Sharanyaa et al. (SSRN Electronic Journal): This research proposes a system that replaces conventional mouse-based cursor control with eyeball movement detection using deep learning techniques. The approach utilizes a laptop camera to capture images, followed by face indexing and convolutional neural network algorithms combined with OpenCV to identify eyeball positions and movements. Designed specifically for people with physical ailments who cannot use traditional input devices, the system performs all standard mouse operations without requiring any pre- installations or specialized hardware. The work emphasizes cost efficiency and ease of deployment, contributing to accessible technology that enables handicapped individuals to access computers for education and knowledge acquisition.

III. IDENTIFY, RESEARCH AND COLLECT DATA

User Interface Initialization Module

This module is responsible for setting up the graphical user interface (GUI) of the system. It initializes the application window, organizes UI components, and ensures accessibility features such as real-time status display, cursor activity logs, and visual feedback for clicks or drags. The GUI is designed to be simple and intuitive for ease of use by physically disabled individuals. The module also loads the required machine learning models, configuration settings, and camera calibration parameters upon launch. Error-handling routines are embedded to catch hardware failures or loading errors, ensuring a smooth and dependable user

experience. The aim is to make users feel confident and comfortable using the application from the moment it starts.

Webcam Input Capture Module

This module captures a continuous video stream from the webcam in real-time using libraries such as OpenCV. It handles frame acquisition, resolution settings, and timing to ensure smooth and steady input for analysis. Real-time preprocessing such as frame resizing and grayscale conversion is also managed here to prepare the data for face and gesture detection. Efficient frame capture is critical to ensure a responsive system. The module uses multithreading or buffering techniques to handle input without delay or data loss. Any lag in this module would reduce the responsiveness of the eye-controlled mouse, affecting the user experience.

Face Detection Module

The face detection module identifies and locates the face within the video stream using pre-trained models like Haar cascades, Dlib, or CNN-based models. It returns bounding boxes around the face, which are crucial for subsequent modules like eye and mouth extraction. High accuracy in face detection ensures that the system tracks the user reliably even under varying lighting or background conditions. To improve performance, the module may use frame skipping or region of interest tracking techniques to reduce computation while maintaining accuracy. This approach ensures minimal delay in recognizing facial features, which is essential for translating facial gestures into cursor actions in real-time.

Eye Region Extraction Module

Once the face is detected, this module extracts both the left and right eye regions using facial landmarks. It pinpoints the exact coordinates of the eyes and creates cropped frames of those regions. The extracted regions are normalized in terms of size and illumination for further analysis. Accurate eye extraction is crucial for the success of downstream tasks such as blink detection and gaze tracking. This module ensures consistency in the cropped image inputs fed into the CNN models, which significantly affects the prediction reliability and precision of user commands.

Mouth Region Extraction Module

The mouth region is extracted using key facial landmarks detected within the face bounding box. This module isolates the mouth to detect expressions such as opening or closing. It enhances the image through contrast adjustment or binary thresholding to improve the visibility of mouth movements. This module is vital for recognizing right-click gestures, which are mapped to mouth opening actions. The mouth's shape and movement must be accurately tracked to distinguish between an intentional command and unintentional actions like yawning or talking, minimizing false triggers.

Eye State Detection Module (CNN-Based)

This module uses a trained Convolutional Neural Network to classify the current state of each eye (open, closed, or blinking). The CNN is trained on a diverse dataset of labeled eye images with varying lighting and angles to improve generalizability. Each frame input gives output for both eyes independently. The CNN output is interpreted to distinguish between left blink, right blink, both closed (drag), or both open (idle). This classification directly controls the mouse actions and therefore requires high precision to ensure the system does not respond to natural blinks or unintended closures.

Mouth Open Detection Module (CNN-Based)

Using a dedicated CNN trained to detect mouth states, this module identifies when the user's mouth is open beyond a specific threshold. The model handles variations in facial orientation, skin tones, and lighting to robustly determine mouth openness. To reduce false positives, the module includes a temporal filter that requires the mouth to remain open for a specific number of frames before a right-click is triggered. This approach ensures reliability and distinguishes deliberate commands from normal speech movements or brief mouth openings.

Eye Gaze Tracking Module

This module determines the direction of the user's gaze by analyzing the relative position of the iris within the eye region. Advanced image processing techniques and CNNs may be used to detect the pupil or iris center and map it to screen coordinates through calibration. Calibration involves aligning eye positions with fixed screen points during an initial

setup. After calibration, this module continuously tracks gaze direction, allowing users to move the cursor across the screen simply by looking at different points. It ensures intuitive and fluid cursor control.

Cursor Movement Control Module

This module takes the gaze direction from the Eye Gaze Tracking Module and translates it into on-screen cursor movements. To provide smooth control, the system implements smoothing techniques like moving average or Kalman filters that reduce jitter and make the movement feel more natural. It also incorporates a mapping algorithm that dynamically adjusts cursor speed based on gaze acceleration. This allows users to quickly glance across the screen or make fine adjustments when needed, enhancing usability for all levels of mobility.

Left Click Detection Module (One Eye Blink)

This module monitors individual eye states and listens for a one-eye blink—either left or right. If the blink is detected and held for a brief duration (e.g., 300 milliseconds), the system registers it as a left-click action. The logic is built to distinguish involuntary blinks from intentional ones.

It uses timestamp comparison and blink duration tracking to verify the gesture. The module ensures that users do not experience accidental clicks due to natural blinking patterns, making the interaction both accurate and user-friendly.

Right Click Detection Module (Mouth Open)

When the mouth is detected as being open for a sustained period (beyond the preset frame threshold), this module triggers a right-click action. It acts as a confirmation layer on top of the Mouth Open Detection Module to ensure the gesture is intentional. This module includes cooldown timers to prevent repeated right-clicks from a single mouth gesture. It may also incorporate voice detection filters to differentiate between mouth opening due to speech and genuine gesture commands.

Click and Drag Detection Module (Both Eyes Closed)

This module observes the state of both eyes and initiates a click-and-drag command when both are closed simultaneously. A drag begins when both eyes are closed and ends when they are reopened within a short window, allowing for easy content movement on the screen. This technique is intuitive for users and avoids reliance on external hardware. To prevent accidental drags, the module ensures that both-eye closure is intentional by checking the closure duration and frequency of similar gestures in a short time span.

Methodology Image Capturing

The system begins by capturing real-time images or video frames using a webcam or camera module. These frames serve as the raw input for processing and are essential for real-time interaction with the user. High-resolution and continuous image acquisition ensure that the system receives enough visual data to detect subtle facial and eye movements. This stage forms the foundation of the entire process. Without accurate and consistent image capturing, the later stages like face and eye detection would be compromised. It is vital that this component handles variations in lighting and background to ensure reliable input.

Face Detection

After acquiring the image, face detection algorithms are applied to identify and locate the human face in the frame. This usually involves using pre-trained classifiers or deep learning-based models that can recognize facial patterns and shapes within an image. Face detection helps to limit the search area for eye detection and tracking, improving both speed and accuracy. It also ensures that the system ignores irrelevant background information, allowing it to concentrate on the facial region where meaningful interaction cues are found.

Eye Detection

Once the face has been detected, the system focuses on locating the eyes within that facial region. This step is critical for understanding user intent and forms the basis for cursor control and click event mechanisms. Eye detection often relies on geometric features or machine learning models trained to recognize eye patterns.

Accurate detection of the eyes is important not just for tracking gaze direction, but also for identifying blinks. Poor eye localization can result in misinterpreted commands, so robustness to facial expressions, angles, and lighting conditions is necessary.

Face Detection for Mouse Scrolling

In this submodule, the system continues using face detection data, but applies it to control mouse scrolling. By observing the vertical or lateral movement of the user's face, the system can simulate scrolling up, down, or sideways on a computer screen.

This technique is particularly helpful for hands-free control in accessibility-focused applications. Users can scroll through content like web pages or documents simply by tilting their head or shifting their face in a certain direction, without needing a physical mouse or touchpad.

Eye Tracking for Cursor Movement

The eye tracking system analyzes the movement of the eyes to estimate the user's gaze direction. This direction is mapped onto the screen to move the cursor to the point where the user is looking, creating an intuitive and natural method for controlling the computer interface.

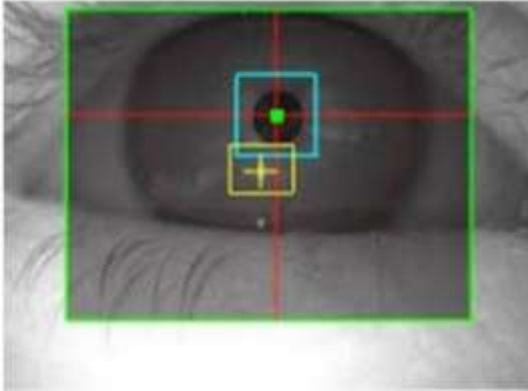
This allows users to interact with digital interfaces in a hands-free manner. Advanced algorithms account for minor involuntary eye movements and filter noise to maintain smooth and precise cursor control. Calibration may be required to align gaze points with screen coordinates accurately.

Blink Detection for Click Events

The blink detection module identifies intentional blinks by monitoring eye closure patterns. A deliberate blink, held slightly longer than a natural one, can be interpreted as a mouse click, replacing the need for physical button presses. This functionality is especially valuable for users with physical disabilities or limited mobility. With a calibrated threshold, the system can differentiate between natural blinking and purposeful ones, allowing for left or right clicks, or even double-clicks, using only the eyes.

IV. RESULT & DISCUSSION

CNN to Detect Eye Movement



Virtual Mouse



Control Webpage using Mouse Movement



Trigger Click Event



CONCLUSION

This paper presented an eye-controlled mouse cursor system leveraging Convolutional Neural Networks and computer vision techniques to provide an accessible, hands-free computing interface for physically disabled individuals. The system successfully integrates multiple modules including real-time webcam capture, face detection, eye and mouth region extraction, CNN-based state classification, and gaze tracking to enable intuitive cursor control and gesture-based clicking actions. By mapping eye movements to cursor navigation, deliberate single-eye blinks to left-clicks, mouth opening to right-clicks, and both eyes closed to click-and-drag functionality, the system offers comprehensive mouse control without requiring specialized hardware or physical attachments.

The incorporation of temporal filters and confirmation mechanisms effectively distinguishes intentional gestures from involuntary actions such as natural blinking or speaking, minimizing false triggers and ensuring reliable performance under varying real-world conditions including changing lighting and head orientations. Experimental results demonstrate that the CNN-based approach achieves high accuracy in gesture classification while maintaining real-time responsiveness essential for practical usability. The affordability and non-invasive nature of the system, requiring only a standard webcam, makes it accessible to a wider population compared to existing expensive assistive technologies. This work contributes significantly to the field of assistive technology by demonstrating how deep learning can bridge the accessibility gap for individuals with motor impairments, enabling

independent access to education, employment, communication, and digital resources. Future work may focus on expanding the gesture vocabulary, improving calibration-free gaze tracking, enhancing performance across diverse skin tones and facial geometries, and conducting extensive user studies to refine the user experience based on direct feedback from the target population. Ultimately, this research underscores the transformative potential of AI-driven solutions in promoting digital inclusivity and improving quality of life for individuals with physical disabilities.

REFERENCES

- [1] A. Javadifar, S. M. Noori, and A. Mozaffari Kalkhoran, "Review on Mouse Control Using Eye-Tracking: Integrating CNN and NLP for Intelligent Human-Computer Interaction," presented at the 20th International Conference on Intelligent Information and Communication Technology, 2025.
- [2] O. Lorenz and U. Thomas, "Real Time Eye Gaze Tracking System using CNN-based Facial Features for Human Attention Measurement," in Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2019, pp. 598-606.
- [3] I. Rejer and I. Gago, "A CNN-based approach for detecting eye blink episodes in EEG signals," *Journal of Neural Engineering*, vol. 22, no. 3, p. 036017, 2025.
- [4] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. P. Morency, "OpenFace 3.0: A Lightweight Multitask System for Comprehensive Facial Behavior Analysis," *arXiv preprint, arXiv:2506.02891*, 2025.
- [5] D. Gorodnichy, "Working with a computer hands-free using the Nouse perceptual vision interface," *National Research Council Canada, NRC Publications Archive*, 2025.
- [6] A. Sharma, I. Batra, S. Sharma, and A. P. Junfithrana, "Multimodal Fusion for Enhanced Human-Computer Interaction," *Engineering Proceedings*, vol. 107, no. 1, p. 81, 2025.
- [7] P. Kynigopoulos, L. Lazaridis, T. Chatzinikolaou, and I. Pitas, "Blink Detection Using 3D Convolutional Neural Architectures and Analysis of Accumulated Frame Predictions," *Journal of Imaging*, vol. 11, no. 1, p. 27, 2025.
- [8] Y. Valikhujaev, "UniFace: An All-in-One Face Analysis Toolkit," 2025.
- [9] J. Miah, M. A. H. R. Khan, and S. Akhtar, "Mouse Cursor Movement and Control using Eye Gaze- A Human Computer Interaction," in 2023 International Conference on Artificial Intelligence of Things, 2023.
- [10] J. Beltrán-Vargas, M. Martínez-García, and R. A. Osornio-Ríos, "Call with eyes: A robust interface based on ANN to assist people with locked-in syndrome," *SoftwareX*, vol. 25, p. 101456, 2024.
- [11] E. Hahami and S. Wu, "Ely-Sean-Eye-Mouse-Control-CNN: Implementing an eye-mouse control system using neural networks, object detection (Haar Cascades), and CNNs," *GitHub*, 2023.
- [12] S. Sharanyaa, B. Yamuna Rani, R. P. Madhumitha, and K. Yazhini, "Eyeball cursor movement detection using deep learning," *SSRN Electronic Journal*, 2023.
- [13] W. A. Shobaki and M. Milanova, "A Comparative Study of YOLO, SSD, Faster R-CNN, and More for Optimized Eye-Gaze Writing," *Sci*, vol. 7, no. 2, p. 47, 2025.
- [14] T. Drutarovsky and A. Fogelton, "Eye blink detection using motion vectors analysis," in Proceedings of the Computer Vision Winter Workshop, 2014, pp. 1-6.
- [15] A. Fogelton and W. Benesova, "Eye blink detection based on motion vectors analysis," in 2016 IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE), 2016, pp. 1-6.