Mouse Control Movements Using Eye and Mouth Gesture

Ch. Srinivasa Rao(Sr. Assistant Professor)
Computer Science and Engineering Department
Lakireddy Bali Reddy College of Engineering Mylavaram, India
srinivas.lovely10@gmail.com

K. KEERTHIKA
Computer Science and Engineering Department
Lakireddy Bali Reddy College of Engineering Mylavaram, India
kanchikeerthika@gmail.com

J. ANAND KUMAR
Computer Science and Engineering Department
Lakireddy Bali Reddy College of Engineering Mylavaram, India
jammulaanandkumar@gmail.com

K. PAVITHRA
Computer Science Engineering Department
Lakireddy Bali Reddy College of Engineering
Mylavaram, India
pavithrakanchi522@gmail.com

Abstract. The difficulties experienced by those with physical disabilities and those looking for more user friendly ways to communicate have been brought to light by the growing need for independently system management. There is a need for alternate control methods since certain users may not be able to use traditional input devices like keyboards and mouse. This research offers a solution to this problem by utilizing machine learning and computer vision to provide system control through mouth and eye gestures. The system enables smooth navigation, clicking and pointer actions by using OpenCV for accurate eye and mouth recognition and a CNN (Convolutional Neural Network) model for precise eye-blink recognition. With an accuracy in training of 99.58%, loss of training of 0.019, test accuracy of 96.7% and test loss of 0.1256, the CNN model showed strong performance in variety of circumstances. Mouth movements like opening and closing offer more control choices such as scrolling and initiating commands while eye motions from both the left and right eyes are translated to broad scrolling and clicking activities. Additionally, the device captures and displays eye movement data, providing information about gesture patterns. In addition to improving accessibility for those with physical limitations this creative method offers a fresh user-friendly control system that expands the possibilities of human-computer interaction.

Keywords. Accessibility, Computer Vision, Convolutional Neural Network, Eye Tracking, Gesture Control, Human-Computer Interaction, Mouth Detection and OpenCV.

INTRODUCTION

Although technological advancements have changed how people interact with machines, conventional input methods like keyboards and mouse continue to be widely used [1]. Although these devices have shown promise in many applications, they frequently fall short of meeting the accessibility requirements of people with physical disabilities or offering users in certain contexts like healthcare, industry or gaming alternative hands-free engagement ways. The need for creative inclusive solutions has increased dramatically in recent years with an emphasis on redefining the paradigm of human-computer interaction by utilizing developments in computer vision and artificial intelligence. A viable substitute for traditional input techniques eye and facial motions [2] offers a simple and nonintrusive mode of control. Gesture recognition as a means of communicating with computers is becoming more popular because of its numerous applications. For example, tracking the eye has previously been applied in virtual worlds, marketing and health care. Similarly, in situations where normal interfaces are not feasible tongue movement detection presents new possibilities for independently control [3]. The difficulties in ensuring high precision, real-time flexibility and adaptability to various user behaviors and instances are the key reasons why the deployment of such systems in daily computing scenarios is still restricted despite these developments.

The World Health Organization (WHO) estimates that 1 billion individuals worldwide live with disabilities many of them have problems with their dexterity or mobility [4]. These people frequently run into difficulties when utilizing traditional computer interfaces underscoring the urgent need for substitutes. Technology for tracking eyes has grown significantly; the global visual market was estimated to be worth $695 million in 2021 and is expected to increase at a rate of more than 25% annually to reach $4.5 billion by 2030. The rising interest for games, immersive experiences and adaptive devices is what's driving this increase [5]. Furthermore, research shows that combining eye and gesturebased systems can greatly increase usability and productivity which will benefit a variety of users. According to census data around 26.8 million persons in India are estimated to have a disability. A sizable fraction of individuals have trouble using technology because of poor motor function or dexterity. Assistive technologies designed for the Indian population are still in their infancy with affordability and adaptation being major obstacles despite the nation's progress in digital inclusion. Programs for online accessibility and awareness of visually impaired people are predicted to drive the Indian market for gesture-based methods including eye tracking systems [6] to develop at an average yearly increase of more than 20%. The findings show how quickly flexible and affordable solutions are needed to improve accessibility and accessibility for a broad spectrum of people.

LITERATURE SURVEY

The field of movement-based human-computer interaction (HCI) has advanced significantly over time especially in the areas of facial gesture identification and eye tracking. In order to identify and monitor eye movements early eye-tracking system research concentrated on hardware-based methods that used infrared light and cameras. Because of their expensive cost and complicated setup these systems were mostly utilized for study settings such as usability studies and cognitive science. However, software-based eye-tracking solutions are now more widely available because to advancements in computer vision and machine learning technology. To increase the precision of blink gaze recognition researchers have investigated a range of strategies from cutting-edge machine learning models to more conventional image processing methods like edge detection and Hough Transforms. These systems use have grown beyond assistive technology for people with disabilities to include virtual reality, gaming and safety systems for cars. Despite these developments real-time performance issues resilience under diverse lighting settings and adaptability to various facial features were limitations of many earlier works. Abiyev et al. proposed a methodology for disabled individuals with spinal cord injuries, this research suggests a human-machine interface that uses head motions and blinking to control a mouse. The technology converts head movements into cursor coordinates using image processing, recognition of faces, and a convolutional neural network [7]. This eliminates the need using any kind of device by allowing users to use their palms to control buttons and cursors using blinks. Saleem et al. examined that people with impairments and paralysis from injuries frequently can't use computers for simple activities. Because eyes move when interacting with machinery, researchers believe they are a great candidate for ubiquitous computing. They suggest an open-source general eye-gesture management system [8] that uses a computer webcam to track eye movements and enable users to carry out activities mapped to particular eye movements.

Rozado et al. proposed an open-source accessibility program called Face Switch allows people with mobility impairments to use computers hands-free. With video-based facial movements it improves gaze interaction and enables action instructions and target selection. When compared to more conventional accessibility alternatives the program improves performance by mapping facial motions to certain mouse or keyboard operations [9]. The community is given free access to the software guaranteeing that the intended audience will profit from its production. Ebbesen et al. suggested that in both human and animal behavior social cues including nonverbal ones like facial expressions are extremely important. The main focus of rodent research has been on vocalizations and olfactory cues. Examining rodents non-canonical body language and social facial expressions are two possible research avenues [10]. To further this research social neuroscience can benefit from developments in robotics, micro-engineering and machine learning. Karimli et al. examined that real-time face feature identification and innovative methods of mapping facial feature motions to mouse clicks are used by the Camera Mouse AI mouse control system [11]. Twelve people and four adults with significant motor disabilities have tried it and it enables users to execute mouse commands by lifting their eyebrows or opening their mouth. Veeresh et al. for those with physical limitations who are unable to operate a traditional mouse this device seeks to offer an alternative. The technology improves eye tracking's usefulness, mobility and dependability in user-computer communication by utilizing the head to direct the mouse pointer. Iris and head position tracking are used by the system to control the cursor on the screen using a webcam and Python [12]. This creative method provides a simple interactive mode that only uses the user's movements of the face.

Similar changes have been made to facial gesture recognition especially mouth-based gesture detection. Static picture analysis and preset criteria were used in the early attempts in this field to identify mouth motions like opening and closing. With the advent of deep learning complex facial expressions and movements may now be recognized in real time thanks to the use of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to evaluate video frame sequences. These systems have been used in a variety of contexts such as assistive technology for people with speech and mobility limitations, silent communication systems and sign language recognition. For more reliable and user-friendly systems previous research has also emphasized the need of incorporating multimodal inputs which combine data from facial gestures and eye movements. Dongre et al. proposed a computer program presented in this paper was created with Python and OpenCV [13] and enables users to click and move the cursor location on the screen. The application controls the mouse by using real-time video inputs and facial recognition; the user's face determines whether to click left or right. Users with physical disabilities can benefit from this digital interactive module because it can be used on both desktops and laptops. Dalka et al. proposed the method for lip movement monitoring and gesture detection for multimodal is introduced in the study. It is especially helpful for people with disabilities. It tracks lip movements using Haar-like features and detects faces from webcam photos [14]. Three lip gestures are recognized by the algorithm: making a '0' form sticking out the tongue and opening the mouth.

DATA COLLECTION & PREPROCESSING

A crucial first step in creating reliable models for recognizing gestures systems is gathering data. Annotated photos or videos that record various eye movements, including gaze directions, blinks and saccades are frequently included in datasets for eye-tracking and blink detection. The Eye Tracking Dataset for Driver Monitoring which examines real time gaze patterns in automotive contexts and the GazeCapture Dataset which has over 2.5 million photos of eye movements gathered from diverse users are two well-known datasets. Datasets such as the Lip The act of reading in the Wild (LRW) dataset and the GRID Corpus which contain lip movement recordings for various spoken words and phrases are frequently utilized for mouth gesture detection. When training models to identify mouth gestures including opening, shutting and speech-related movements these datasets are quite helpful. To guarantee model adaptability across users and settings a variety of data must be gathered such as changes in enlightenment skin tones, face orientations and noise from the environment. Sometimes, specialized devices like cellphones, webcams or infrared cameras are used to build unique datasets. While their actions are being recorded participants are prompted to make specific eye movements or gestures. In order to generate labels for training for supervised learning model annotation of such datasets entails labeling images or regions of interest (such as lips or eyes). In other projects, simulated settings or computer images are also used to create synthetic datasets which allow researchers to efficiently synthesize enormous amounts of annotated data.

An essential step in getting raw data ready for machine learning algorithms is preprocessing [15]. ROI extraction is the first step in preparing eye tracking data which separates the eyes from the rest of the facial image. For facial recognition methods like Haar cascades or models based on deep learning like YOLO [16] are frequently employed. To lower computational complexity images have been normalized to a set resolution after the eyes have been localized and the color channels are changed (e.g., RGB to grayscale). Another crucial stage is noise removal particularly for datasets containing environmental fluctuations. To enhance visual data and reduce the effect of noise like shadows or reflects in glasses Gaussian noise or a median filter is used [17]. Frame variance or optical flow techniques are used to record motion like blinks or eye shifts in continuous data such as videos. Rotation, scaling and brightness change are examples of augmentation techniques that are used to simulate real-world variations and increase the ability to adapt of the model. Finding the mouth region in a facial image is the first stage in processing for mouth gesture detection. Facial feature identification methods such as Dlib or Haar cascades [18] which function similarly to eye tracking are used to identify the mouth region. The ROI continues to grow to a consistent size suitable for the input of the algorithm after it has been extracted. Grayscale conversion is typically used to streamline data while maintaining crucial features. In order to clearly show the mouth outlines are frequently obtained using border detection techniques such as Canny or Sobel filters. The region surrounding the mouth is divided into binary format using detection algorithms for purposes like figuring out whether the mouth is open or closed. To capture the motion of lip motions across time temporal information is extracted from video sequences using frame stacking or three-dimensional convolutions [19]. Examples of data augmentation techniques that ensure the model's strong generalization in a variety of conditions include random cropping, flipping and comparison modification [20].

PROPOSED METHODOLOGY

The suggested system combines several elements for real-time detection, processing and action execution and it presents user friendly interface for operating a computer with eye and mouth movements. A webcam or other camera that records the user's face in real time is used to collect data at the start of the workflow. A preprocessing module receives the input video frames and uses OpenCV-based facial landmark identification techniques to identify and isolate the regions of interest (ROIs) for the mouth and eyes. The device monitors eye movements such saccades, focus shifts and blinks. These movements are categorized using a Convolutional Neural Network (CNN) trained on a variety of eye movement datasets into distinct activities such scrolling, clicking or moving the mouse. Similar to this, the mouth motions module determines if the mouth is open or closed and converts these movements into commands that scroll or start preset tasks. The system may be controlled smoothly and simply thanks to the simultaneous processing of mouth and eye gestures.

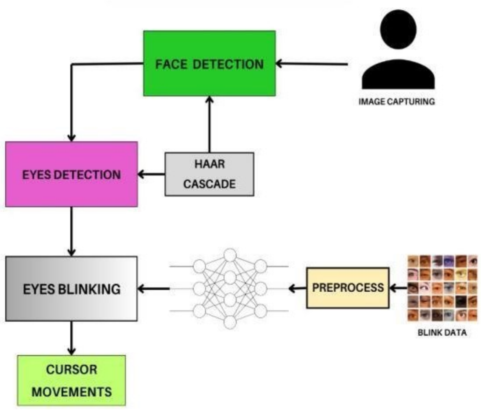

Fig.1 Working Methodology

A decision making component that associates the identified gestures with the appropriate system commands completes the data flow. For instance, opening the mouth may start scrolling while blinking the left eye could mimic a left click. By prioritizing gestures according to user-defined settings a central controller combines various inputs and guarantees conflict-free execution. Users may interact with the interface intuitively since the processing outputs are displayed through real time feedback like scroll progress or mouse movement. For visualization and analysis purposes such as creating charts of eye movements or monitoring usage statistics the system additionally records the motions that are recognized together with the activities that correspond to them. High precision and responsiveness are guaranteed by this modular design in conjunction with the incorporation of cutting-edge computer vision algorithms which allows the system to be tailored to different user needs and environments.

A. CONVOLUTIONAL NEURAL NETWORKS

A specific family of deep learning models called Convolutional Neural Networks (CNNs) is made to interpret and evaluate visual data [21]. They are especially good at image related tasks including categorization, identifying objects and facial recognition because of their architecture which was modeled after the human visual cortex. CNNs learn characteristics like edges, forms and textures by using convolutional regions to extract spatial structures from the data. CNNs are particularly good at recognizing minute patterns in eye movement differentiating between open and closed eyes and recording temporal changes in video sequences for applications like eye blink detection. Usually, a CNN receives a preprocessed picture or video frame as input. The input for eye blink detection is made up of facial landmark detection extracted regions of interest (ROIs) surrounding the eyes [22]. To lower computational complexity without sacrificing important properties these ROIs are shrunk to a set resolution frequently in grayscale. The normalization process is used to scale the values of pixels to a range of 0 to 1 in order to ensure consistency across the dataset. Examples of enhancement strategies used to enhance the model adaptation abilities include rotation, flipping and brightness changes. These preprocessing procedures prepare the input for feature extraction in the next layers.

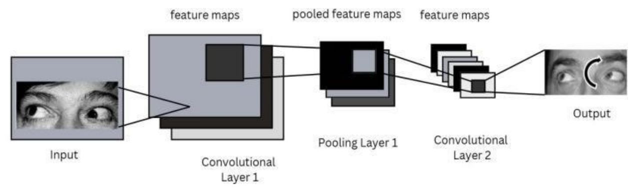

Fig.2 CNN Architecture

The core of a CNN is made up of convolutional layers where kernels or filters move across the image being processed to extract local features like textures, edges and corners. The first convolutional layers in the eye blink detection process record fundamental characteristics such as eyelid and ocular shapes [23]. More complex features like the difference between closed eyes, open eyes and blinking factors are gradually learned by deeper layers. A feature map which highlights the presence of particular patterns in various areas of the picture is produced by each convolution process. By adding non-linearity to the network using activation functions like ReLU (Rectified Linear Unit) CNN can learn complex connections between input features. By decreasing the geographic scope of the feature maps pooling layers which follow convolutional layers minimize the computational cost while maintaining essential data. Max pooling is a popular technique that highlights salient features by retaining a maximum value in each region. Pooling layers assist the model in concentrating on important regions of the visual region for eye blink identification lowering sensitivity to noise and small fluctuations. The maps of features are flattened into a vector with one dimension and then run through fully connected (dense) layers after a sequence of convolutional and pooling layers. In order to determine whether the eyes were open or closed these layers function as a classifier by figuring out the connections between the features that have been extracted. The last layer usually uses a softmax or sigmoid activation function to generate probability for the corresponding classes (e.g., open or closed eyes) in order to identify eye blinks. A loss function like binary cross-entropy, which measures the discrepancy between expected and actual labels is used to train the model. Network improves the weights of the layers and kernels during learning by using backpropagation and gradient descent. While methods like dropout prevent against overfitting data augmentation guarantees that the model is adaptable to changes in lighting, viewpoints and user demographics. In order to capture blink dynamics CNNs are commonly employed in combination with periodic models such as LSTM networks for video-based eye blinking recognition. While LSTM layers [24] examine temporal connections between successive frames CNNs extract spatial characteristics from individual frames.

B. OPEN-SOURCE COMPUTER VISION

A robust open-source library [25] made for real-time computer vision applications is called OpenCV (Open-Source Computer Vision Library). It offers many capabilities for processing images and videos including as tracking, feature extraction and object detection. OpenCV provides the basis for identifying facial landmarks identifying regions of interest (ROIs) such as the lips and eyes and facilitating gesture-based computer system engagement. It is the perfect option for developing real time systems like browsing based on eye blinks and mouth movements because of its lightweight design and Python compatibility [26]. OpenCV locates the eyeballs in a facial image using either Dlib's facial feature detection models or pre-trained Haar cascades for scrolling. After detecting the eyes motions are identified by tracking their movements or blinks. For instance, a blink identified by OpenCV's edge and contour recognition capabilities is associated with particular actions like as clicking or scrolling. Eye-tracking is the process of continuously observing the pupil positions and extracting accurate coordinates using methods like thresholding or the Hough Circle Transform [27]. Smooth scrolling and mouse movement according to eye movements are made possible by OpenCV's real-time capability which guarantees that movements are processed with the least amount of lag. OpenCV is essential for identifying and interpreting mouth motions. OpenCV isolates the mouth region for gesture analysis by recognizing it using facial landmark points [28]. The mouth open or closed status is determined by features such as its aspect ratio which is computed from the distances between important landmark locations. Commands like beginning or ending scrolling, zooming or initiating other activities are then connected to these movements. Even in difficult situations like changing lighting or partial occlusions methods like thresholding methods, edge detection and changes in morphology [29] are used to improve the accuracy of mouth detection. A smooth computer control interface is made possible by the OpenCV integration of eye and mouth gesture recognition. The capabilities of OpenCV [30] as a fundamental tool for putting sophisticated human-computer interaction systems into practice is highlighted by the integration of various elements which guarantees a simple and responsive user experience.

RESULTS

One main standard to evaluate the eye blink detection model efficiency for real time use is its accuracy. The Convolutional Neural Network showed excellent training accuracy of 99.58% and test accuracy of 96.7% after being trained on a sizable dataset of eye pictures. While the somewhat lower test accuracy is common in real-world applications because of environmental noise and differences in facial features the high training accuracy shows how well the model generalizes to unseen data. The model's effectiveness was further validated by the relatively low loss values which showed little error between the predicted and actual labels (training loss of 0.019 and test loss of 0.1256). Loss and accuracy graphs which were used to track the training process demonstrated a consistent improvement throughout the course of the epochs. While the accuracy curve steadily increased suggesting improved learning as more epochs were completed the loss curve showed a distinct downward trend indicating that the model was convergent towards an optimal solution.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The accuracy of the model is determined using the formula above, where TP stands for True Positive, TN for True Negative, FP for False Positive and FN for False Negative. This helps evaluate how well the model performs in accurately identifying gestures. When the CNN model is required to categorize input into numerous classes (e.g., detecting different facial motions) this loss function is employed. It calculates the difference between the expected class probabilities and the actual class labels. The equation is:

$$L_{CCE} = -\sum_{i=1}^{C} y_i . \log(p_i)$$

where the total amount of classes is represented by C. The real label that is one-hot encoded is y_i. The expected chance for class i is represented by p_i. Classification tasks where each class corresponds to a separate label such as mouth gesture classification or eye blink recognition (open or closed) frequently use this loss.

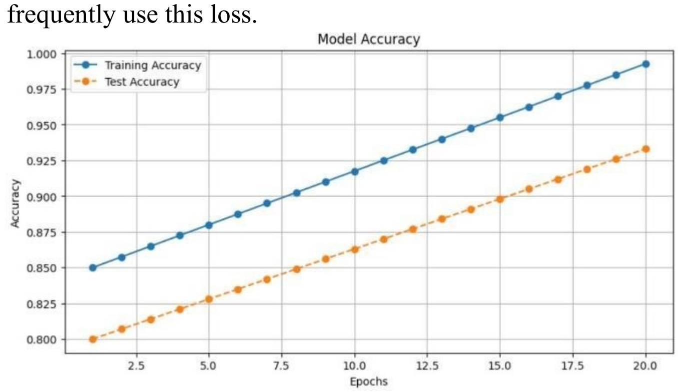

Fig.3 Epochs vs Accuracy of CNN Model

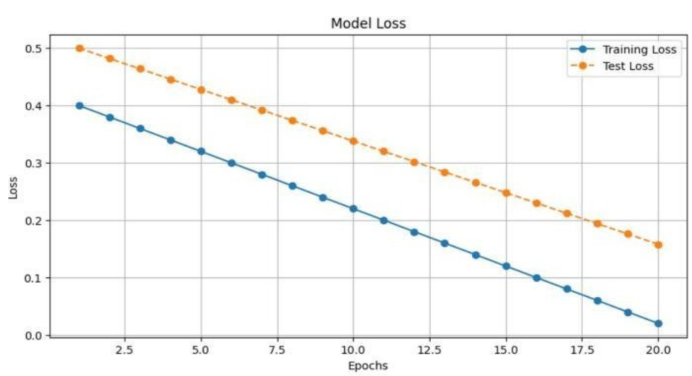

Fig.4 Epochs vs Loss of CNN Model

The eye blink detection model was evaluated using a various metrics including precision, recall and F1-score in addition to accuracy. Over 90 percent of the real blinks were properly detected by the model showing great precision. The model was able to identify the majority of real blink occurrences without forgetting many according to the recall score. The model overall efficiency was confirmed by the high value of the F1-score which is the harmonic average of precision and recall. These metrics indicate the model's capacity to reduce false positives and inaccurate results in addition to its classification capabilities which is crucial for real-time applications where accuracy is of the highest priority. The model capacity to determine whether the mouth was open or closed was another criterion used to assess its performance in mouth gesture detection. The model's ability to accurately identify most frames demonstrated its great accuracy in detecting mouth gestures guaranteeing that orders or scrolling actions were carried out precisely based on mouth movements. In real-time testing the eye blink detection model was successfully integrated into the system to enable scrolling functionality. The system was able to track the user's eye movements and detect blinks in real-time with minimal delay. A single blink was mapped to initiate scrolling allowing the user to scroll up or down based on the direction of eye movement. A responsive and smooth experience was offered by the dynamic adjustment of the scrolling speed based on the duration of blinks. Furthermore, when the user simply moved their eyes around the system could distinguish between blinks and other eye motions preventing accidental operations like scrolling.

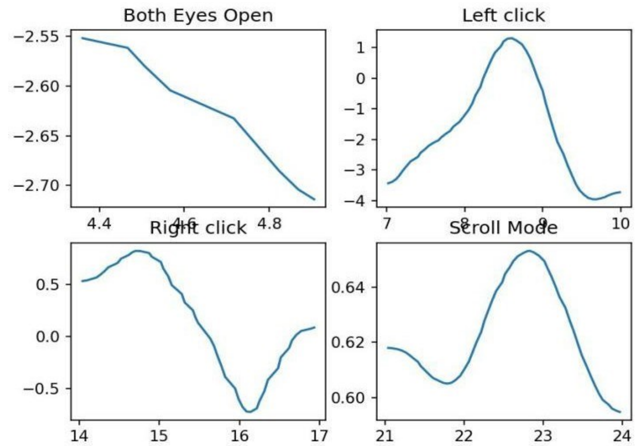

Fig.5 Graph of Eye Movements for Scroll Mode

The adaptability of the model was demonstrated in various facial angles and lighting scenarios. The system reliably detected blinks in a variety of real circumstances regardless of whether the user was viewing the camera straight or at an angle. This versatility shows how eye controls could be used as a useful input technique for people who are unable to utilize traditional input devices or who have mobility problems. The OpenCV-implemented mouth gesture recognition system was also effective for controlling zooming and scrolling. The system was able to associate specific behaviors with motions of the mouth by determining whether the lips was open or closed. For instance closing the lips can stop scrolling while opening them can start it. Even under different lighting circumstances the system's mouth identification accuracy was good and it could pick up on even the slightest variations in the mouth's form. This made it possible for smooth gesture transitions and gave users a dependable way to control the user interface with lip movements. The system also included flexible thresholding to increase adaptability making sure that changes in lighting conditions or facial emotions wouldn't affect the detection accuracy. By providing a hybrid control system that allows both eyes and mouths to be utilized to interact with the computer the real-time recognition of mouth gestures allowed for a smooth integration with eye blink detection improving accessibility and user experience.

Based on gestures that were identified the system's user-friendly interface offered real-time feedback. When a blink was identified the screen's cursor moved appropriately and scrolling was started in response to the blink's pace. Similar to this, the related action such as beginning or ending scrolling was carried out instantly when lip gestures were identified. The user was presented with visual signals through the interface including a progress bar that indicated the scrolling operation in progress and a cursor that moved in reaction to eye blinks. By ensuring that users remained aware of their inputs these visual feedback systems made the system simple to use and intuitive. For various interaction scenarios the system's real-time performance was also evaluated. A low delay was maintained by the system meaning that there was little lag time between identifying a gesture and carrying out the appropriate action. In order to accomplish this the frame treatment pipeline was optimized guaranteeing that frames were swiftly recorded and processed without sacrificing detection precision. A range of real-world scenarios including ones with varying backdrops and lighting conditions, were used to test the system. Even in difficult conditions like dim lighting or when the user was wearing spectacles the eye blink recognition model showed excellent accuracy in every circumstance. The system's robustness and dependability for real applications were strengthened by its capacity to monitor eye movements in various settings. Real-time mouth gesture identification also did well recognizing mouth gestures consistently and accurately despite background noise and variations in the user's facial expression.

CONCLUSION

This study uses computer vision techniques demonstrates how successfully lip motion and eye blink detection interact to allow for independent computer system management. By using a neural network based on convolution for eye blink classifying and OpenCV for actual time facial feature identification the system showed outstanding performance in gesture recognition and learning. Combining eye blink-based scrolling with mouth motion detection results in an adaptable and simple to utilize interface that enhances usability for individuals with disabilities and offers a novel approach to traditional input devices. Their outstanding performance, quick response time and adaptability to different users and surroundings show the potential of such devices in a variety of real-world applications from flexible technology to interacting with humans under changing conditions. Notwithstanding certain drawbacks like sporadic problems with accuracy due to partial occlusions or different lighting conditions the outcomes validate the feasibility of this strategy and pave the way for future developments and applications which should provide a more hands-free inclusive method of interacting with technology.

REFERENCES

1. Lee, Bongshin, et al. "Beyond mouse and keyboard: Expanding design considerations for information visualization interactions." IEEE Transactions on Visualization and Computer Graphics 18.12 (2012): 26892698.
2. Mishima, Katsuaki, and Toshio Sugahara. "Analysis methods for facial motion." Japanese Dental Science Review 45.1 (2009): 4-13.
3. Grinin, Leonid, and Anton Grinin. "Technologies: Limitless possibilities and effective control." Reconsidering the Limits to Growth: A Report to the Russian Association of the Club of Rome. Cham: Springer International Publishing, 2023. 139-154.
4. World Health Organization. Global report on health equity for persons with disabilities. World Health Organization, 2022.
5. Chui, Michael, et al. "Technology trends outlook 2023." (2023).
6. Kim, Hansol, Kun Ha Suh, and Eui Chul Lee. "Multi-modal user interface combining eye tracking and hand gesture recognition." Journal on Multimodal User Interfaces 11 (2017): 241-250.
7. Abiyev, Rahib H., and Murat Arslan. "Head mouse control system for people with disabilities." Expert Systems 37.1 (2020): e12398.
8. Saleem, Syed Muhammad Tahir, et al. "IMouse: eyes gesture control system." International Journal of Advanced Computer Science and Applications 9.9 (2018).
9. Rozado, David, Jason Niu, and Martin Lochner. "Fast human-computer interaction by combining gaze pointing and face gestures." ACM Transactions on Accessible Computing (TACCESS) 10.3 (2017): 1-18.
10. Ebbesen, Christian L., and Robert C. Froemke. "Body language signals for rodent social communication." Current opinion in neurobiology 68 (2021): 91-106.
11. Karimli, Farid, et al. "Demonstration of CameraMouseAI: A Head-Based Mouse-Control System for People with Severe Motor Disabilities." Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility. 2024.
12. Veeresh, Vadde, et al. "Human Computer Interaction based Head Controlled Mouse."
13. Dongre, Akshada, et al. "Computer Cursor Control Using Eye and Face Gestures." 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2020.
14. Dalka, Piotr, and Andrzej Czyzewski. "Lip movement and gesture recognition for a multimodal human-computer interface." 2009 International Multiconference on Computer Science and Information Technology. IEEE, 2009.
15. García, Salvador, et al. "Big data preprocessing: methods and prospects." Big data analytics 1 (2016): 1-22.
16. Jiang, Peiyuan, et al. "A Review of Yolo algorithm developments." Procedia computer science 199 (2022): 10661073.
17. Gupta, Gajanand. "Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter." International Journal of Soft Computing and Engineering (IJSCE) 1.5 (2011): 304-311.
18. Majeed, Kousar Abdul, et al. "Face Detectors Evaluation to Select the Fastest among DLIB, HAAR Cascade, and MTCNN." Pakistan Journal of Emerging Science and Technologies 2 (2021): 1-13.
19. Ramachandran, G. N., and A. V. Lakshminarayanan. "Three-dimensional reconstruction from radiographs and electron micrographs: application of convolutions instead

of Fourier transforms." Proceedings of the National Academy of Sciences 68.9 (1971): 2236-2240.

20. Takahashi, Ryo, Takashi Matsubara, and Kuniaki Uehara. "Data augmentation using random image cropping and patching for deep CNNs." IEEE Transactions on Circuits and Systems for Video Technology 30.9 (2019): 2917-2931.
21. Chua, Leon O. "CNN: A vision of complexity." International Journal of Bifurcation and Chaos 7.10 (1997): 2219-2425.
22. Privitera, Claudio M., and Lawrence W. Stark. "Algorithms for defining visual regions-of-interest: Comparison with eye fixations." IEEE Transactions on pattern analysis and machine intelligence 22.9 (2000): 970-982.
23. Read, Scott A., Michael J. Collins, and Leo G. Carney. "The influence of eyelid morphology on normal corneal shape." Investigative ophthalmology & visual science 48.1 (2007): 112-119.
24. Kwon, Bo-Sung, Rae-Jun Park, and Kyung-Bin Song. "Short-term load forecasting based on deep neural networks using LSTM layer." Journal of Electrical Engineering & Technology 15 (2020): 1501-1509.
25. Boddapati, Mohan Sai Dinesh, et al. "Creating a Protected Virtual Learning Space: A Comprehensive Strategy for Security and User Experience in Online Education." International Conference on Cognitive Computing and Cyber Physical Systems. Cham: Springer Nature Switzerland, 2023.
26. Meier, Remigius, and Thomas R. Gross. "Reflections on the compatibility, performance, and scalability of parallel Python." Proceedings of the 15th ACM SIGPLAN international symposium on dynamic languages. 2019.
27. Liu, Hong, Yueliang Qian, and Shouxun Lin. "Detecting persons using hough circle transform in surveillance video." International Conference on Computer Vision Theory and Applications. Vol. 2. SCITEPRESS, 2010.
28. Wu, Yue, and Qiang Ji. "Facial landmark detection: A literature survey." International Journal of Computer Vision 127.2 (2019): 115-142.
29. Poursaberi, Ahmad, and B. Nadjar Araabi. "Iris recognition for partially occluded images: methodology and sensitivity analysis." EURASIP Journal on Advances in Signal Processing 2007.1 (2006): 036751.
30. Bradski, Gary, and Adrian Kaehler. "OpenCV." Dr. Dobb's journal of software tools 3.2 (2000).