IRO Journal on Sustainable Wireless Systems (ISSN: 2582-3167)
www.irojournals.com/irosws/

Oculang: Empowering Communication through Blink Language Detection

Suriya S.[1], Anubharathi V S.[2], Bharathi G.[3], Harshada R.[4], Vignesh Aditya R.[5], Sundarsree B G.[6]

[1]Associate Professor, [2-6]UG scholars,Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India.

E-mail: [1]suriyas84@gmail.com, [1]ss.cse@psgtech.ac.in, [2]anuvs2003@gmail.com, [3]bharathi1010ad@gmail.com, [4]harshadaravi@gmail.com, [5]vigneshaditya.r@gmail.com, [6]sundarsree3@gmail.com

Abstract

Oculang is a communication system that uses computer vision and machine learning techniques to enable individuals with neurodegenerative disorders to communicate using eye gestures. The application focuses on detecting facial landmarks, gaze estimation, and blink detection to analyze combinations of eye movements captured from video input and produce message outputs. Dlib's shape predictor (68 facial landmarks) and OpenCV-based image processing methods are used to extract and process the features of the eye region. A decision-making algorithm maps the detected gestures and predefined keywords for message generation. The application supports both real-time video capturing and uploading video through a Django-based user interface. Experimental evaluation on recorded datasets of eye movements demonstrated robust performance in accurately recognizing blinks, winks, and gaze directions, validating the system's reliability.

Keywords: Computer Vision, Human-Computer Interaction, Assistive Technology, Accessibility, Vision-Based Interaction, Facial Landmark Detection, Gaze Estimation, Thresholding Techniques, Geometric Ratios.

1. Introduction

Eye movement detection has become increasingly significant in human-computer interaction technologies. These systems have paved the way for innovative applications across accessibility and healthcare. By analyzing gaze patterns and eye closures, researchers can gain insights into cognitive states, task engagement, and even emotional well-being. Neurological disorders such as Parkinson's Disease (PD) affect over 8.5 million people globally, with its

DOI: https://doi.org/10.36548/jsws.2024.4.003
Received: 04.11.2024, received in revised form: 27.11.2024, accepted: 12.01.2025, published: 22-01-2025

prevalence doubling in the past 25 years. Amyotrophic Lateral Sclerosis (ALS), with nearly 30,000 cases in the U.S., is predicted to see a 69% rise in global incidence over the next 25 years. Spinal cord injury (SCI) affects over 15 million people, contributing to long-term disability and higher mortality rates in low-income countries. These diseases significantly impact communication abilities, preventing the patients from expressing their basic needs. Parkinson's disease (PD) affects speech and voice in up up to 90% of patients, leading to hypophonia, monotone speech, and a "masked face," while cognitive changes further challenge interactions. People with ALS have muscles that are so weak that they have trouble speaking and swallowing. In SCI, respiratory issues and limited mobility restrict verbal and non-verbal communication and the patient requires the help of assistive technologies. For these patients, the eyes can serve as a vital communication tool to express their needs.

Oculang revolves around detecting a unique language that has alphabets as basic eye movements, like right, left, up, down, blink, and wink. The intuitive combinations of these eye gestures form meaningful messages. This eye-based language is simple to learn, making it accessible for patients to use in daily communication and serving as a powerful tool for individuals with limited speech or motor abilities, enabling them to effectively express their needs and thoughts through a structured system of eye gestures.

Existing systems for eye-based communication often face several limitations, including the need for extensive training and calibration, limited vocabulary, and challenges in adapting to individual user needs. By integrating facial feature detection and eye masking techniques, Oculang enhances accuracy and reduces distractions, ensuring seamless communication even in complex environments. Additionally, Oculang processes gestures such as right and left winks, further widening the vocabulary and increasing the number of possible messages.

The core of the Oculang research lies in its sophisticated eye-tracking algorithms and machine learning models, which analyze the intricate patterns of eye movements to predict the intended communication in real-time. Through the implementation of Oculang, The research aspires to substitute inclusivity, enabling individuals with paralysis to engage in meaningful interactions and express themselves with ease.

2. Literature Survey

The research study provides a comprehensive review of gaze tracking advancements in computer vision, covering its evolution, applications, and techniques. It introduces a new taxonomy to classify gaze-tracking approaches, emphasizing the importance of considering the application scope and context when evaluating gaze estimation solutions. Key findings include the use of deep neural networks, PCCR techniques, and infrared eyeglasses for improved accuracy [1]. The survey concludes by exploring more adaptable and flexible gaze-tracking solutions to handle varying domains and applications. [2] offers a detailed overview of face recognition (FR) systems, focusing on the key stages of the process and the challenges that impact performance. The study reviews various techniques like Viola-Jones, PCA, ICA, SIFT, and others used for facial feature extraction and recognition. It highlights the importance of face recognition in areas such as security, healthcare, and banking, while also pointing out challenges like handling variations in pose, occlusions, and different age or resolution levels. The research [3] aims to develop a face recognition technique using facial landmarks, with area ratios and angles as feature vectors. The method extracts facial landmarks through the Haar cascade algorithm and uses them to calculate area triplets and angles for identification. The findings show high accuracy, with minimal error for the same person, and the technique reliably identifies individuals even with geometric facial transformations. The study in [4] presents two methods for automatic detection of visual attention to areas of interest (AOIs) using pre-trained deep learning models: Image Classification (IC) and Object Detection (OD). The methods performed well for AOIs with distinct concepts that align with the pre-trained model classes but faced limitations when AOIs lacked support in the model or had weak matches, resulting in poor performance or failures to detect attention. The study concludes that while these methods show promise, they require further refinement to handle unsupported or ambiguous AOIs and improve overall performance. The study [5] presents a robust webcam-based eye-gaze estimation system for human-computer interaction (HCI), utilizing deep learning and Mediapipe technology. The system achieves 99% validation accuracy in gaze prediction and 20 FPS in real-time applications, such as mouse pointer control and scrolling. By using a low-cost webcam and a multiclass prediction approach, the method reduces computational complexity while maintaining high accuracy. The proposed system is suitable for real-time HCI applications, offering a cost-effective and intuitive alternative to traditional wearable devices, with potential for widespread adoption.

The review challenges in handling blink-induced missing data in eye-tracking systems and proposes standardized solutions. Analyzing 81 studies, finding that 42.9% lacked sufficient reporting, 45.7% detailed methods like interpolation and imputation, and 11.4% removed affected data, it was found that interpolation is the most common method, though practices vary widely. Key recommendations include avoiding data removal to prevent interaction delays, inspecting variations in eye movement speed around blinks to address artefacts, using linear or cubic spline interpolation for infilling missing data, and adopting community-sourced guidelines for standardized processing [6]. This study explores how pupil size and blink rate can be used to measure task load types: cognitive, perceptual, physical, and communicative, and their relationship with working memory (WM) models. It finds that pupil size can indicate cognitive and communicative loads, while blink rate is more effective at distinguishing cognitive load levels and perceptual tasks [7]. The study shows that the DAM-WM model better explains the interaction between task load and WM components than the M-WM model. It concludes that these physiological measures can provide valuable insights into task loads, offering a more complete understanding of the human cognitive system and its application in more complex tasks. The primary objective of [8] is to provide a comprehensive guide and set of tools for accurately identifying blinks within eye-tracking data. Additionally, the study aims to demonstrate the utility of blink durations as a measurement not only of whether mind wandering occurs but also of the extent to which the mind is wandering. The research also offers practical guidance for extracting blinks from eye-tracking data reliably and consistently, contributing to a better understanding of the relationship between mind wandering and blink durations across different task modalities. The research [9] introduces a modified Eye Aspect Ratio (EAR) method for detecting eye blinks, which recalculates the EAR threshold based on individual eye sizes. By using facial landmark detection with Dlib's pre-trained model, the method consistently detects eye blinks more effectively, making it suitable for real-time applications like drowsiness detection. The objective is to improve blink detection accuracy, outperforming traditional EAR methods. Key findings show the modified method achieves high precision (99%) and AUC scores (up to 96.85%) across different datasets, demonstrating robustness against variations in eye sizes and facial expressions. The research uses smartphone apps as assistive tools for visually impaired individuals and aims to create a repository to improve access for clinicians and patients [10]. The study reviews the literature and conducts various surveys and trials to assess the effectiveness of these apps. It finds that smartphone apps are affordable, portable, and versatile, offering benefits like reading and navigation assistance. However, many apps lack scientific evaluation, making it tough for clinicians to recommend them. The research calls for more controlled studies to better understand the real-world impact of these apps in vision rehabilitation.

The research [11] introduces a real-time blink detection system using the Eye Aspect Ratio (EAR) for driver safety, focusing on unique challenges such as individuals with small eyes, wearing glasses, and driving cars. By utilizing facial landmarks and the EAR method, the system achieves an optimal EAR threshold value of 0.18, yielding a 96% accuracy on certain datasets. The research demonstrates the technique's applicability for drowsiness detection, making it a valuable tool for enhancing driver safety. The research [12] presents a graphical user interface (GUI) for preprocessing eye-tracking data from Gazepoint and FOVIO eye trackers. The tool addresses the challenges of analyzing raw eye-tracking data by implementing a standardized approach that leverages a Savitzky-Golay filter for noise reduction and the Nystrom and Holmqvist (NH) algorithm for event detection. Key findings highlight its reliability in detecting fixations and saccades and its successful application in multiple studies. The study [13] introduces a dataset of 10,000 annotated eye images, designed for training custom convolutional neural networks (CNNs) for eye-tracking tasks. The dataset was created using the dlib library for face and landmark detection, followed by resolution enhancements with OpenCV to scale images to 416 x 416 pixels. An experimentally derived equation enabled the identification of the iris in these images, allowing precise annotations. The dataset was validated on a CNN model, achieving a tracking error of three degrees, confirming its utility for training the initial layers of eye-tracking neural network models. The research study [14] introduces GazeCapture, a crowdsourced dataset for mobile eye tracking, and iTracker, a deep learning model that makes gaze prediction accessible on everyday smartphones and tablets without extra hardware. By utilizing diverse real-world data, iTracker achieves impressive accuracy, with gaze errors as low as 1.34 cm on phones and 2.12 cm on tablets after calibration. The study highlights the power of large-scale datasets and shows how iTracker's features generalize well to other datasets. The study [15] analyzes 31 studies on the impact of assistive technology (AT) in the education of students with disabilities, highlighting both its benefits and the challenges in its implementation. The review finds that AT significantly enhances inclusion, accessibility, and students' social skills, autonomy, and motivation, but barriers such as teacher training, lack of resources, and limited access remain.

3. Proposed Work

3.1 Methodology

The methodology used in Oculang integrates computer vision techniques and user interface to detect, analyze, and interpret eye movements for communication. The process is divided into the following stages:

- **Eye Movement Detection and Analysis:** Oculang uses computer vision algorithms to detect subtle eye movements captured via video inputs. These algorithms are designed to identify basic eye signs, such as blinks, gaze directions, and wink. The detected movements are mapped to a set of predefined keywords.
- **Message Generation:** The combination of detected eye movements is interpreted through a decision-making algorithm that matches them to corresponding keywords.
- **User Interaction Modes:** Oculang provides two modes of user interaction for different communication scenarios:
- **Pre-recorded Video Analysis:** Users can upload eye movement videos. These videos are processed and messages will be displayed based on the predefined mapping.
- **Real-Time Detection:** For real-time communication, Oculang provides a live detection feature. This will be dynamically processed and the message will be generated immediately.
- **Interface Design:** A user-friendly interface has been developed for seamless interaction. The design ensures ease of use for both user interaction modes.

3.2 System Architecture

Oculang is a cutting-edge communication system designed to empower individuals suffering from conditions such as ALS/MND, spinal cord injuries, strokes, and various neurodegenerative disorders to communicate effectively through eye gestures. The system architecture as shown in Figure 1 comprises multiple components seamlessly integrated to capture, process, and interpret eye movements and provide us with the actual message the patients want to convey, facilitating intuitive communication pathways.

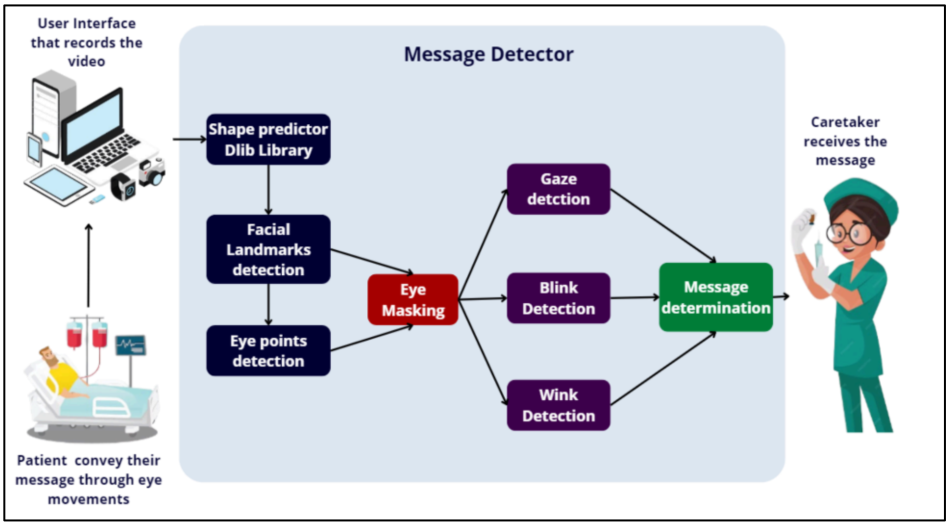


Figure 1. System Architecture

Figure 1 comprises multiple components seamlessly integrated to capture, process, and interpret eye movements and provide us with the actual message the patients want to convey, facilitating intuitive communication pathways.

The interaction between the user and the Oculang system begins with the user accessing the intuitive user interface (UI), which serves as the primary point of engagement. For individuals facing conditions such as ALS/MND, spinal cord injuries, strokes, and various neurodegenerative disorders, the UI provides an easy way to communicate with their caretakers or doctors effectively through their eye gestures. The user has two options either to upload pre-recorded videos so that the caretakers or doctors can use them accordingly or to capture live videos of the patient and display the messages that they wish to communicate.

The captured or uploaded video is processed frame by frame. The video undergoes comprehensive analysis using computer vision techniques. The first step involves employing the shape predictor functionality of the dlib library to detect facial features, including the 68 landmarks associated with the eyes. With these landmark detections, the system is implemented to focus the eye region by masking out. These masks serve as a crucial step in enhancing the accuracy of gesture detection by isolating the eye regions from surrounding facial features and distractions. Based on the facial landmark points and the distance between them, several ratios provide valuable insights into the orientation and alignment of the eyes. The calculations of those are mentioned in section 3.4. With these ratios and calculations, basic eye gestures are detected such as blinking, winking, looking up, down, left, and right. Oculang employs a sophisticated system to interpret the combination and sequence of these gestures to derive meaningful messages. The system utilizes a predefined mapping between gesture sequences and corresponding messages, allowing it to translate detected gestures into actionable communication signals. The UI now displays the communication messages to the caretakers or doctors, allowing them to understand the patient's needs.

3.3 Experimental Setup

3.3.1 Upload a Recorded Video or Capture a Video

A recorded video or a live video is captured and uploaded through the Django interface. This video is processed frame by frame. The home page of the platform is shown in Figure 2. The process begins with the user capturing or uploading a recorded video through the user interface. Once the video is uploaded, it appears on the platform as illustrated in Figure 3 and 4. Additionally, a live video can be captured, as demonstrated in Figure 6.

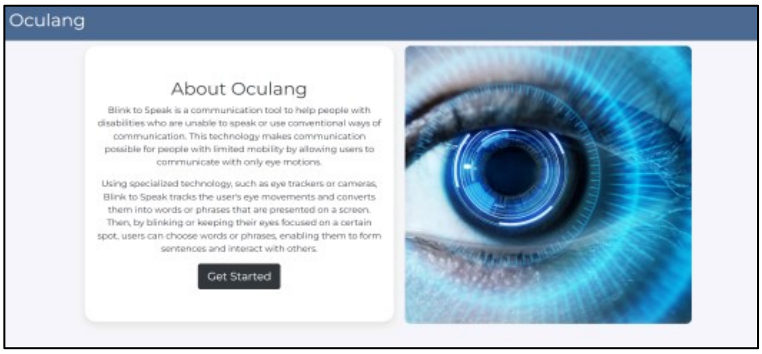
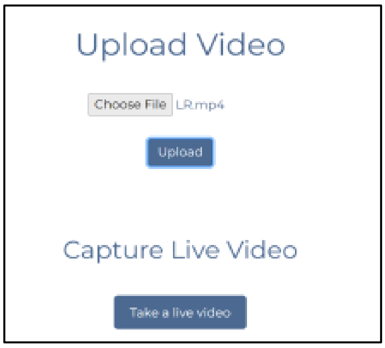


Figure 2. Home Page



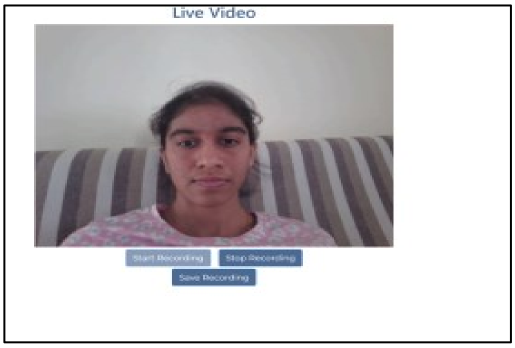


Figure 3. Video Uploaded

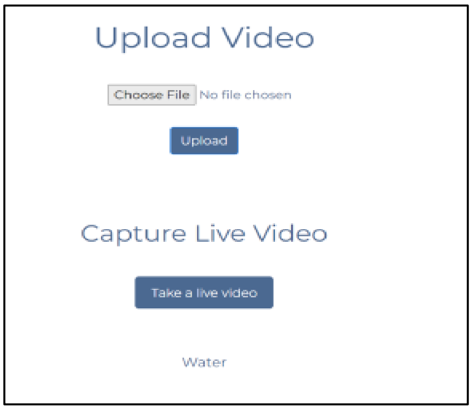
Figure 4. Live Video Captured

3.3.2 Processing

The video frames are analyzed using a facial landmark detection model to extract the eye regions and calculate various ratios. These ratios are used to determine the specific gesture.

3.3.3 Message Output

An array of gestures will be generated after the processing. The array is compared with the dictionary and the respective message is displayed to the user as shown in Figure 5 and 6.



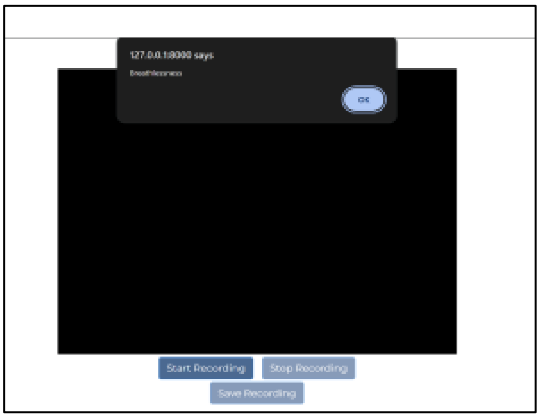


Figure 5. Upload Video Result

Figure 6. Live Video Result

3.4 Techniques Used

3.4.1 Facial Landmark Detection

The dlib library's pre-trained model (shape_predictor_68_face_landmarks.dat) is used to identify the 68 facial landmarks as shown in Figure 7 including the eye regions. Landmarks 36-41 are for the left eye and 42-47 are for the right eye. Each frame is passed through Dlib's face detector to locate the face and focus on the eyes (36-47)

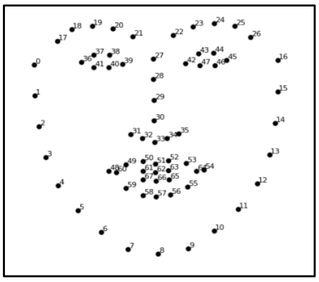


Figure 7. Facial Landmarks

3.4.2 Eye Points Detection

The six key points around each eye will be extracted. Left eye - 36,37,38,39,40,41 and Right eye - 42,43,44,45,46,47 points outline the upper and lower eyelids which are used to detect eye closure.

3.4.3 Eye Masking and Gaze Detection

OpenCV is used to generate a mask around the eye region. This is used to isolate the eye region from the rest of the face. This is important for analyzing eye movements, blinking, and winking. This application calculates horizontal and vertical gaze ratios as shown in Figure 8 and 9 based on the eye's position in the frame. The relative distances between key points on the eye are measured to determine whether the person is looking left, right, up, or down. The get_vertical_ratio function calculates the vertical ratio of the eye by determining the distance between the upper and lower eyelids relative to the vertical centerline of the eye. It computes the top and bottom points of the eye from facial landmarks, calculates the midpoint between the top two and bottom two points of the eye, and uses the Euclidean distance (hypotenuse) to derive the vertical line length. The ratio is computed by summing the distances from the centerline to the top and bottom points and dividing by the vertical line length, providing a measure of eye openness. The get_horizontal_ratio function estimates the horizontal gaze direction by analyzing the distribution of white pixels in a thresholded image of the eye region. A mask is created to isolate the eye area from the grayscale image, and the eye is divided into left and right halves. The white pixel count in each half is compared to determine gaze direction; if more white pixels are on the left, the user is looking right, and vice versa. Special conditions handle cases with no white pixels, assigning a default gaze ratio of 1 or 5 to avoid division errors.

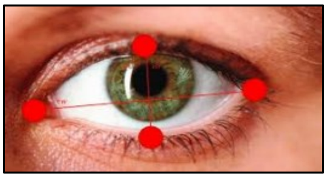


Figure 8. Horizontal Ratio

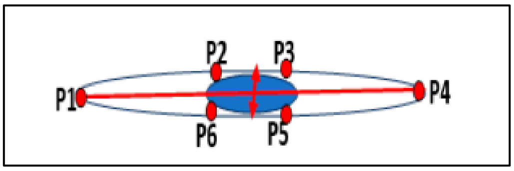


Figure 9. Vertical Ratio

3.4.5 Blink Detection

The ratio between the horizontal and the vertical distances of eyelids is calculated to detect a blink. The ratio decreases when the eyelids are closed which indicates a blink as shown in Figure 10. The get_blinking_ratio function determines the eye aspect ratio (EAR) by comparing the horizontal distance between the outer and inner corners of the eye to the vertical distance between the upper and lower eyelids. Facial landmarks mark key points of the eye, and the midpoint between the upper and lower eyelids is calculated. The function measures the horizontal and vertical distances and returns the ratio between them, with a lower ratio indicating eye closure, which can be used to detect blinks.

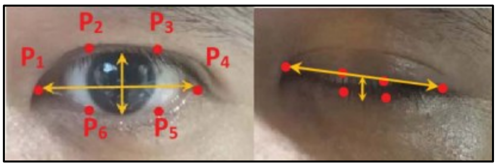


Figure 10. Blink Detection

3.4.6 Wink Detection

The ratios of the left and right eyes are compared to detect a wink. If one eye has a high blink ratio, then it is considered a wink. The get_winking_ratio function calculates the blinking ratio for both the left and right eyes by calling the get_blinking_ratio function separately for each eye. Using specific landmark points for each eye, it returns two values representing the eye aspect ratios, which can be compared to detect winks or uneven eye closures, with significant differences between left and right eye ratios indicating a possible wink.

3.4.7 Message Determination

The application matches gestures such as blink, wink or gaze direction with predefined messages using a dictionary mapping function.

3.5. Equipment Used

3.5.1 Hardware Components

The application runs on a local machine. A processor (i5 or higher), at least 8GB RAM, a minimum of 250GB storage, a stable internet connection, a webcam, or any video capturing device are essential for smooth video capturing, processing, and storing the video files. A dedicated GPU is optional to accelerate image processing tasks and speed up facial landmark detection.

3.5.2 Software Components

Operating System: Windows, Linux or macOS (64-bit) are all compatible with the application.

Python: Python 3.6 or higher is required to run this application.

Web Framework: Django 3.x or higher is used to build the user interface.

Libraries and Dependencies: OpenCV (version 4.x or higher) is used for video processing and image manipulation. Dlib (version 19.x or higher) is used for facial landmark detection. The pre-trained shape_predictor_68_face_landmarks.dat is essential for detecting and analyzing facial landmarks.

Database: A simple file-based database (local storage) is used for video storage.

Development Tools: Visual Studio Code for writing and debugging the code.

Browser: Any web browser such as Google Chrome, Mozilla Firefox can be used to interact with the user interface.

4. Results and Discussion

4.1 Basic Signs Detection

The system effectively detects various eye movements as shown in Figure 11 and 12.

Figure 11. Basic Signs Detection

4.2 Message Detection

The Figure 12 shows the message determination for the signs - Up and Down. Example; Message: I feel like eating.

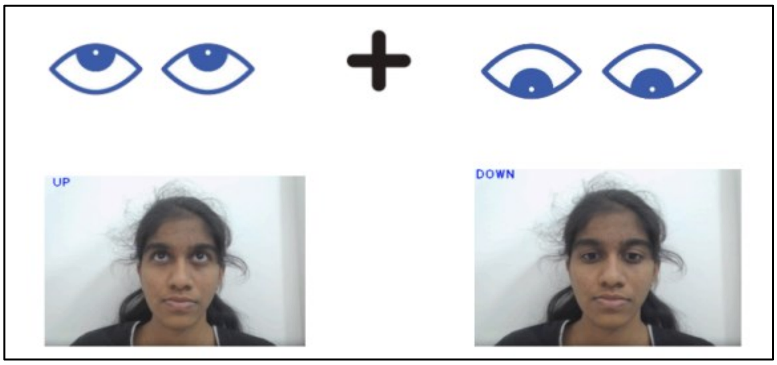


Figure 12. Message Determination for the Signs - Up + Down

Accuracy: 85.71%

5. Conclusion

Oculang provides a transformative communication tool for paralyzed individuals by translating eye movements into meaningful messages. Future improvements aim to reduce message detection delays, enhance responsiveness, and develop dedicated desktop and mobile apps for broader accessibility. Adding features like text-to-speech can further assist caretakers and improve usability for visually impaired users. Personalization options will also be explored to meet individual user needs, fostering a more inclusive and effective communication experience.

References

[1] Cazzato, Dario, Marco Leo, Cosimo Distante, and Holger Voos. "When I look into your eyes: A survey on computer vision contributions for human gaze estimation and tracking." Sensors 20, no. 13 (2020): 3739.

[2] Gururaj, H. L., B. C. Soundarya, S. Priya, J. Shreyas, and Francesco Flammini. "A Comprehensive Review of Face Recognition Techniques, Trends and Challenges." IEEE Access (2024).

[3] Juhong, Aniwat, and Chuchart Pintavirooj. "Face recognition based on facial landmark detection." In 2017 10th Biomedical Engineering International Conference (BMEiCON),Hokkaido, Japan. IEEE, 2017. 1-4.

[4] Barz, Michael, and Daniel Sonntag. "Automatic visual attention detection for mobile eye tracking using pre-trained computer vision models and human gaze." Sensors 21, no. 12 (2021): 4143.

[5] Roy, Koushik, and Dibaloke Chanda. "A robust webcam-based eye gaze estimation system for Human-Computer interaction." In 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET),Chittagong, Bangladesh. IEEE, 2022. 146-151.

[6] Grootjen, Jesse W., Henrike Weingärtner, and Sven Mayer. "Uncovering and Addressing Blink-Related Challenges in Using Eye Tracking for Interactive Systems." In Proceedings of the CHI Conference on Human Factors in Computing Systems, Canada. 2024. 1-23.

[7] Chen, Siyuan, Julien Epps, and Fred Paas. "Pupillometric and blink measures of diverse task loads: Implications for working memory models." British Journal of Educational Psychology 93 (2023): 318-338.

[8] Hollander, John, and Stephanie Huette. "Extracting blinks from continuous eye-tracking data in a mind wandering paradigm." Consciousness and Cognition 100 (2022): 103303.

[9] Dewi, Christine, Rung-Ching Chen, Xiaoyi Jiang, and Hui Yu. "Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks." PeerJ Computer Science 8 (2022): e943.

[10] Pundlik, Shrinivas, Prerana Shivshanker, and Gang Luo. "Impact of apps as assistive devices for visually impaired persons." Annual Review of Vision Science 9, no. 1 (2023): 111-130.

[11] Dewi, Christine, Rung-Ching Chen, Chun-Wei Chang, Shih-Hung Wu, Xiaoyi Jiang, and Hui Yu. "Eye aspect ratio for real-time drowsiness detection to improve driver safety." Electronics 11, no. 19 (2022): 3183.

[12] Atweh, Jad A., Ihab Tabbara, Ghady Nasrallah, and Sara L. Riggs. "Reducing Barriers in Analyzing Eye Tracking Data: The Development of a GUI to Preprocess Eye Tracking Data." In 2024 Systems and Information Engineering Design Symposium (SIEDS),Charlottesville, VA, USA. IEEE, 2024. 286-291.

[13] Rakhmatulin, Ildar. "Dataset for eye-tracking tasks." Rakhmatulin, I. Dataset for Eye-Tracking Tasks. Preprints 2020120047 (2020).

[14] Krafka, Kyle, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. "Eye tracking for everyone." In Proceedings of the IEEE conference on computer vision and pattern recognition, USA. 2016. 2176-2184.

[15] Fernández-Batanero, José María, Marta Montenegro-Rueda, José Fernández-Cerero, and Inmaculada García-Martínez. "Assistive technology for the inclusion of students with disabilities: a systematic review." Educational Technology Research and Development 70, no. 5 (2022): 1911-1930.