WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Trends and Techniques in Eye-Gaze Communication Systems

Dr. N. Sree Divya[1*], Nishitha Kanakanapuri[2] and Jahnavi Katta[3]

[1,2,3 *]Department of IT, Mahatma Gandhi Institute of Technology, Gandipet, Hyderabad, 500075, Telangana, India.
E-mail(s): nsreedivya_it@mgit.ac.in; nishithakanakanapuri@gmail.com; kattajahnavi27@gmail.com;
DOI : https://doi.org/10.55248/gengpi.6.0425.1417

ABSTRACT

Gaze-based communication systems have significantly advanced, offering trans- formative solutions for individuals with severe motor disabilities. This survey paper consolidates and compares various gaze-controlled communication tech- nologies, including blink-based interaction methods, virtual keyboards, and gaze estimation techniques. Key approaches reviewed include "NETRAVAAD: Inter- active Eye-Based Communication System for People with Speech Issues," which enhances accessibility through intuitive eye gestures, and "Translated Pattern- Based Eye-Writing Recognition using Dilated Causal Convolution Network," which leverages neural networks for eye-writing interpretation. Additionally, methodologies like "E-Gaze: Gaze Estimation with Event Camera" focus on pre- cision gaze tracking under dynamic conditions, while "Blink-To-Live Eye-Based Communication System" introduces innovative emergency alert systems for real-time interaction. Applications discussed span assistive technologies, human- computer interaction, and advanced predictive systems. The survey evaluates these methods based on accuracy, robustness, and user experience, identifying challenges such as calibration complexity and lighting variability. This study underscores the potential of gaze-based communication to redefine assistive technology, promoting inclusivity and enhancing quality of life.

Keywords: Gaze communication, Eye-tracking, Blink detection, Assistive technology, Virtual keyboards

1. Introduction

The rapid advancements in computer vision and assistive technologies have driven sig- nificant progress in gaze-based communication systems, enabling innovative solutions for individuals with severe physical disabilities. At the core of these developments is the field of eye tracking, which has transitioned from basic gaze-tracking techniques to advanced machine learning architectures capable of accurately interpreting eye move- ments and gestures in dynamic environments [1, 2, 5]. The ability to reliably estimate gaze direction, detect blinks, and map eye movements to communication commands has become essential for applications such as virtual keyboards, real-time assistive devices, and interactive systems for individuals with speech impairments [4, 10].

Traditional gaze-based communication systems relied heavily on hardware- intensive methods, such as infrared-based pupil tracking, to estimate gaze direction. These systems were effective in controlled settings but struggled to adapt to variations in lighting, head position, and user-specific differences [5, 7]. As applications expanded, research shifted toward data-driven and adaptive approaches, introducing algorithms capable of dynamically adjusting to contextual changes in gaze behavior and environ- mental conditions, thereby enhancing reliability and usability in real-world scenarios [2, 6].

The introduction of deep learning marked a turning point for gaze-based com- munication, with convolutional neural networks (CNNs) and other machine learning architectures significantly improving gaze estimation and interaction. CNNs facilitated the automatic learning of hierarchical features from complex input data, enabling pre- cise differentiation of gaze direction and blinks even in noisy or dynamic environments. Hybrid approaches that combined CNNs with recurrent networks, such as LSTMs, proved particularly effective for sequential gaze interpretation, allowing systems to accurately translate eye movements into typing or control commands [3, 6]. These innovations have empowered real-time systems to handle challenging conditions, such as head motion and variable lighting, with greater precision [5, 9].

Recent advancements focus on improving the accessibility and efficiency of gaze-based communication systems. Notable examples include NETRAVAAD, which integrates intuitive gaze gestures for effective communication, and Blink-To-Live, which emphasizes real-time emergency alert functionality for users with speech impair- ments [1, 4]. Similarly, predictive typing systems, such as Real-Time Human-Computer Interaction Using Eye Gazes, leverage natural language processing (NLP) models to reduce the number of inputs required for text generation, enhancing speed and usabil- ity [7]. Advances in gaze estimation using event cameras have further improved system performance under varying environmental conditions, offering solutions that are both robust and scalable [3, 8]. The paper by Zhou et al. (2022) reviews various gaze-based control methods for assistive devices, discussing their applications in enhancing com- munication and mobility for individuals with disabilities. It highlights the progress in gaze tracking technologies and their growing importance in assistive systems [11]. Smith and Huggins (2021) focus on blink-based eye gesture recognition, addressing the challenges of detecting blinks for communication systems, and exploring its integra- tion into assistive technologies [12]. Lee et al. (2020) examine the use of eye tracking for speech impairment recovery, discussing how eye movement can control communi- cation devices, offering a means of

interaction for those with speech disabilities [13]. Tanaka and Yamaguchi (2021) investigate gaze-controlled interfaces for real-time assis- tive communication, focusing on systems designed to help people with severe motor disabilities communicate using eye movements [14]. Ohashi and Nakamura (2022) explore eye gaze pattern recognition in augmented reality environments, emphasiz- ing its application in improving assistive technologies and dynamic user interaction in real-time settings [15].

Wang et al. (2024) explore real-time gaze tracking techniques for virtual reality applications, highlighting methods that enhance user experience by enabling more intuitive interactions in virtual environments [16]. Kim and Park (2023) focus on adaptive gaze-controlled communication systems for disabled individuals, presenting solutions that improve accessibility by adjusting to individual needs and environmental conditions [17]. Nguyen and Tran (2022) provide a comprehensive review of gaze- based interaction in augmented reality environments, discussing the challenges and advancements in using eye movements for interaction in AR systems [18].

In recent years, advancements in gaze-based technologies have opened new possi- bilities for assistive systems, offering enhanced accuracy, reduced latency, and robust real-time applicability. These systems have demonstrated significant potential in enabling seamless human-computer interaction, particularly for individuals with dis- abilities. By integrating precise gaze tracking, advanced algorithms, and adaptive interfaces, modern solutions continue to bridge the gap between accessibility and effi- ciency, paving the way for inclusive and innovative applications in communication, healthcare, and beyond.

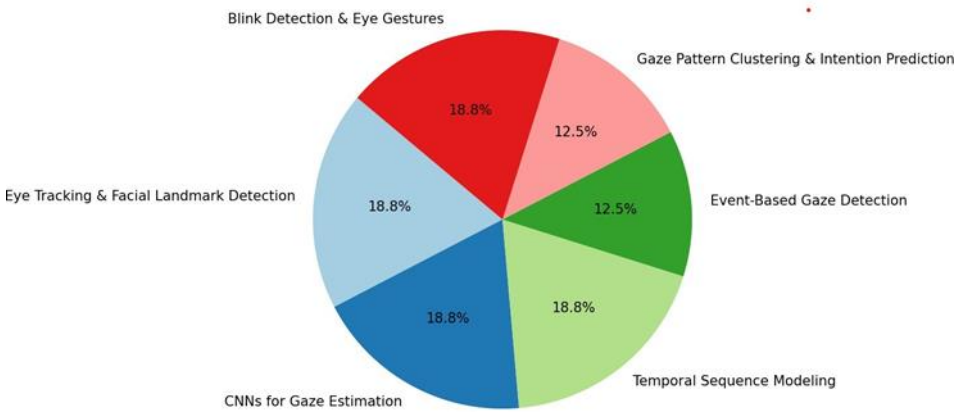

Fig. 1: Methodologies across various papers

Figure 1 illustrates the distribution of various gaze-based methodologies and tech- nologies. The majority share, approximately 18.8%, is attributed to Blink Detection & Eye Gestures, Eye Tracking & Facial Landmark Detection, CNNs for Gaze Estima- tion, and Temporal Sequence Modeling, indicating their prominence in gaze research. Event-Based Gaze Detection and Gaze Pattern Clustering & Intention Prediction hold smaller shares, at 12.5% each. This distribution highlights the diversity of approaches in gaze estimation systems, where both traditional and advanced methods contribute to innovations in eye-tracking, intention prediction, and gaze behavior analysis.

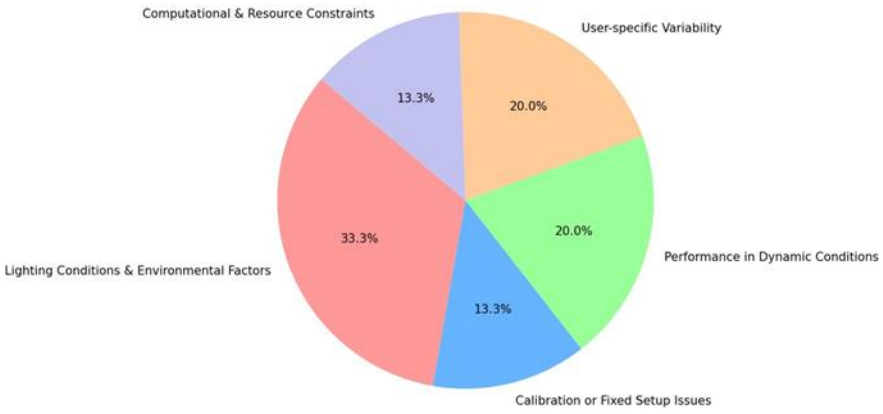

Fig. 2: Limitations across various papers

Figure 2 highlights the key challenges associated with gaze-based systems. Lighting Conditions & Environmental Factors represent the most significant issue, comprising 33.3% of the challenges, indicating the critical impact of external conditions on sys- tem accuracy. User-specific Variability and Performance in Dynamic Conditions each account for 20%, emphasizing the difficulty in adapting systems to individual differ- ences and real-world movement. Calibration or Fixed Setup Issues and Computational & Resource Constraints contribute 13.3% each, underscoring the need for precise sys- tem tuning and the limitations posed by hardware or processing power. Together, these challenges emphasize the importance of robust, adaptable, and resource-efficient solutions in gaze-based technology development.

1.1 Problem Statement

Developing effective gaze-based communication systems for individuals with severe motor impairments presents significant challenges due to the variability in user behaviors, environmental conditions, and the complexity of accurately detecting gaze patterns. Traditional eye-tracking methods, such as infrared-based pupil detection and simple gaze mapping, often struggle in real-world environments with fluctuat- ing lighting, head movements, and varying user physiological conditions. Additionally, while modern gaze-controlled systems utilizing machine learning have made significant progress, they are still limited by computational inefficiencies, the need for extensive calibration, and difficulties in ensuring high accuracy across diverse users and envi- ronments. These limitations underscore the need for a more adaptable, real-time, and efficient gaze-based communication system capable of providing reliable interaction and accurate input for users with severe disabilities, even in dynamic and complex settings.

1.2 Motivation

The growing need for assistive technologies in the communication space, especially for individuals with severe physical disabilities, calls for more efficient, adaptable, and real-time gaze-based communication systems. Existing gaze-controlled systems often face challenges when dealing with varied user behaviors, environmental conditions, and the complex task of accurately detecting eye movements. Traditional gaze-tracking methods struggle to provide consistent performance due to fluctuations in lighting, head positioning, and differences in individual users' eye features. Additionally, while modern systems that leverage deep learning techniques offer significant improvements, they are still limited by computational inefficiencies, the need for extensive user cali- bration, and difficulty ensuring accurate results in dynamic real-world settings. These limitations highlight the need for a more robust and scalable gaze-based communica- tion system capable of reliably addressing the complexities of interaction and ensuring usability for individuals with severe motor impairments.

The following are the contributions of this project:

- To evaluate the effectiveness of deep learning-based gaze estimation models, such as CNNs, in improving gaze tracking accuracy in dynamic environments.
- To identify key challenges in existing gaze-based systems, including issues with eye movement calibration, tracking accuracy, and real-time responsiveness, and propose potential solutions.
- To benchmark the developed system against existing gaze-controlled systems, measuring its performance in terms of accuracy, computational efficiency, and usability.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive literature review, discussing key developments and limitations in existing gaze-based communication systems and their applications for individuals with disabilities. Section 3 details the methodologies employed in the system, focusing on the techniques used for gaze tracking and eye movement recognition. Section 4 outlines the implementation details of the system, covering the hardware, software, and system architecture used to develop various methodologies discuddes in Section 3 . Section 5 presents and analyzes the evaluation results, comparing the system's performance against existing solutions in terms of accuracy, angular error, and latency. Finally, Section 6 concludes the paper by summarizing key findings and suggesting future research directions to advance gaze-based communication technologies.

2. Related Works

Over the years, significant progress has been made in the development of gaze- based communication systems, particularly in the context of assistive technologies for individuals with physical disabilities. Early systems relied on basic gaze-tracking techniques, such as infrared-based methods, to detect eye movements for controlling virtual interfaces. However, these systems often faced limitations in terms of accuracy, especially in dynamic environments with varying lighting and user behaviors. With the advent of machine learning and deep learning techniques, gaze estimation models have improved significantly, offering more precise tracking and interaction capabilities. Research in gaze-controlled virtual keyboards, such as those utilizing predictive text, has demonstrated promising results in enhancing communication efficiency for users with limited motor abilities. Furthermore, the integration of gaze tracking with other modalities, such as blink detection and eye gestures, has enabled more intuitive and hands-free communication systems. Despite these advancements, many existing sys- tems still struggle with challenges such as real-time processing, user calibration, and adaptability across diverse environmental conditions. The following section reviews key works in the field, highlighting the achievements, and limitations of existing gaze-based communication technologies.

Netravaad, an innovative eye-based communication system for individuals with speech impairments caused by conditions like ALS, cerebral palsy, or locked-in syndrome. Utilizing a unique eye-sign language called Netravaani and the Sarani algo- rithm, the system translates eye movements captured via a low-cost USB camera into spoken words or sentences. This device eliminates the need for interpreters, provides predictive text capabilities, and ensures portability with its adjustable stand and user- friendly GUI. Testing demonstrated high accuracy in detecting alphabets (91%), words (100%), and numbers (93%) among users aged 26–35, showcasing its potential as an effective augmentative and alternative communication (AAC) solution [1].

Despite its promise, Netravaad has several limitations. The system's accuracy and response time can be affected by lighting conditions, as it is highly light-sensitive. Additionally, the Sarani algorithm occasionally faces challenges in differentiating cer- tain eye signs, leading to detection delays. The GUI lacks advanced customization options, such as on-screen adjustments for screen size or aspect ratio. Furthermore, while predictive text improves

usability, the system's reliance on predefined patterns restricts flexibility for some users. Future improvements, such as integrating machine learning models, could enhance accuracy and adaptability, but these would require extensive datasets and further testing [1].

Translated Pattern-Based Eye-Writing Recognition Using Dilated Causal Convolu- tion Network, focuses on leveraging eye movements for character recognition, providing a novel approach to assist motor-impaired individuals. By employing root transla- tion and dilated causal convolutional (DCC) networks, the study addresses challenges like non-uniformity and long eye fixations in eye-writing patterns. Using a dataset of English letters and Arabic numerals from 20 participants, the proposed model achieved impressive accuracy (96.20%) and outperformed traditional methods on sev- eral benchmark datasets, making it a significant advancement in the field of assistive communication technologies [2].

The limitations of, Translated Pattern-Based Eye-Writing Recognition Using Dilated Causal Convolution Network, primarily stem from constraints in the deep learning architecture. The network struggles with gaze points that fall outside its receptive field, which can lead to misclassifications, especially for patterns with sig- nificant distortions or missing essential parts. While increasing the number of DCC layers could extend the receptive field and improve recognition, it would also result in significantly longer training times, posing a challenge for scalability. Additionally, the system has not yet been validated with real-world users, such as ALS patients, limiting its applicability and generalizability. Addressing these issues would require incorporating mechanisms to enhance spatial and temporal information capture, such as skip connections or alternative network designs [2].

E-Gaze, a novel system for gaze estimation utilizing event cameras, which capture asynchronous, motion-triggered data rather than traditional frame-based imagery. By leveraging the high temporal resolution, low latency, and sparse data format of event cameras, the system extracts pupil features in real time using spatiotemporal event distributions. These features feed into a recurrent neural network (RNN) that incor- porates a coordinate-to-angle loss function for accurate gaze direction estimation. The system achieves angular accuracy of 0.46° with sub-millisecond latency, making it suitable for extended reality (XR) applications [3].

Despite its innovation, the system has limitations. It requires calibration for each user session due to headset positioning inconsistencies, which can be time-intensive. Additionally, the system's reliance on a fixed eye-to-display distance for the angular loss function constrains its adaptability to various headset designs. The experiments were conducted in controlled laboratory conditions, meaning its robustness against real-world challenges like variable lighting, reflections, and occlusions from eyewear remains untested, potentially limiting its broader applicability [3].

Blink-To-Live system introduces a cost-efficient and accessible eye-tracking com- munication solution designed for individuals with speech impairments caused by conditions such as Amyotrophic Lateral Sclerosis (ALS). Using a smartphone cam- era, the system relies on computer vision to detect eye gestures—Blink, Left, Right, and Up—and translates them into over 60 predefined daily commands, which are dis- played and converted into speech in the user's native language. Unlike other expensive sensor-based systems, this innovative approach does not require specialized hardware, making it a practical solution for low-income settings. The application also minimizes training time and enhances usability, allowing users to communicate efficiently with fewer eye movements [4].

The Blink-To-Live system encounters challenges such as occasional delays in com- munication due to the extensive backend processing of video frames in real-time. Furthermore, the absence of downward eye gesture tracking limits its gesture set, and fast or inconsistent eye movements can result in unrecognized inputs. The system's speed and accuracy heavily depend on stable internet connectivity and the user's abil- ity to maintain precise eye control without head movement. Additionally, individuals with low educational levels or limited technological familiarity may require prolonged training periods to achieve proficiency [4].

Uncertainty-Aware Gaze Tracking for Assisted Living Environments, proposes a novel method for gaze tracking in multi-camera setups within assisted living facilities. It utilizes a neural network regressor that predicts gaze direction based on facial key- point positions and incorporates a unique feature—uncertainty estimation for each prediction. This uncertainty is leveraged within an angular Kalman filter framework to enhance temporal consistency and robustness. The method addresses challenges such as occlusions and unfavorable views through Confidence Gated Units (CGUs), which mitigate low-confidence keypoints. Evaluations on datasets like MPIIFaceGaze and Gaze360, alongside a specialized assisted living dataset (MoDiPro), demonstrate superior performance over state-of-the-art methods, offering accurate and temporally stable gaze predictions [6].

Despite its advancements, the approach has some limitations. The reliance on facial keypoints makes the system vulnerable to significant occlusions or extremely low lighting conditions, where pose estimation models may fail to detect keypoints. Addi- tionally, the system assumes a static 2D environment and does not fully exploit 3D gaze data, limiting its applicability in complex real-world scenarios requiring precise spatial understanding. Moreover, high computational requirements for real-time track- ing, particularly due to the dependency on pose estimation algorithms like OpenPose, might hinder deployment in resource-constrained environments [6].

Real-time human-computer interaction using eye gazes, introduces a real-time human-computer interaction (HCI) system utilizing eye gaze recognition. This sys- tem employs a standard RGB webcam to detect, track, and interpret four distinct eye gazes—looking straight, left, right, and blinking—using a Dlib facial landmark detector and a sclera-based region-of-interest method. It integrates eye gaze recogni- tion with a Mask R-CNN-based instance segmentation model to identify and segment tools and parts in images. The system is packaged into a user-friendly visual interface, allowing users to control and select objects through eye movements alone. Experimen- tal results demonstrate robust performance, with eye gaze recognition achieving high accuracy and instance segmentation maintaining high precision and recall values [7]. Despite its strengths, the system has limitations. Its accuracy declines with increased distance beyond 160 cm between the eyes and the webcam, restricting its effective operational range. The model's robustness in varying environmental condi- tions, such as low lighting or reflective surfaces, has not been thoroughly explored. Additionally, the instance segmentation model relies on a pre-defined set of objects,

limiting its generalizability to new or complex environments without retraining [7]. Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive

Technologies, presents an innovative solution for gaze estimation using state-of-the- art convolutional neural network (CNN) algorithms. It offers a non-intrusive and lightweight alternative to traditional eye-tracking methods, which often rely on intru- sive head-mounted devices or infrared systems. The proposed approach utilizes a simple webcam to estimate gaze and facial position in real time, achieving significant improvements in computation speed (up to a 91% reduction in time) while maintain- ing comparable accuracy. This solution is particularly aimed at enhancing assistive robotic arms for individuals with motor disabilities, making it viable for both indoor and outdoor environments without requiring user-specific calibration [8].

Non-Intrusive Real-Time Eye Tracking Using Facial Alignment, highlights several limitations. There is a lack of standardized benchmarks for comparing computational requirements of gaze estimation methods, leading to challenges in evaluating per- formance consistently across different systems. Additionally, appearance-based gaze estimation methods, such as the one proposed, may show reduced accuracy compared to infrared-based systems (e.g., PCCR). Training dataset biases, such as underrep- resentation of certain demographics, can also affect model accuracy for some users. Furthermore, while the method does not require person-specific calibration, this may limit its precision in scenarios where individual-specific adjustments could enhance performance. Finally, the approach may struggle with extreme variations in head pose and gaze angles that are not well-represented in the training data [8].

A data-driven framework for intention prediction via eye movement with appli- cations to assistive systems, introduces a data-driven framework to predict user intentions based on eye movement, targeting assistive technologies for individuals with limited motor or communication abilities. The framework uses spatial and temporal patterns of gaze, employing clustering techniques like DBSCAN to identify regions of interest (ROIs) and hidden Markov models (HMMs) to capture transition sequences between these ROIs. Through transfer learning with CNNs, it identifies objects in the displayed images and predicts user intentions during and after tasks. The framework achieves an impressive accuracy of 97.42% in predicting daily-life activities and incor- porates early prediction capabilities with a CNN-LSTM model, showcasing significant advancements over previous methods [9].

A data-driven framework for intention prediction via eye movement with applica- tions to assistive systems, framework's applicability is currently limited to controlled environments like kitchens, with tasks involving only 3–4 objects, and it has not been tested in more complex or diverse scenarios. The reliance on 2D images constrains its ability to differentiate objects aligned along the same axis, suggesting the need for 3D imagery. Its performance is highly dependent on accurate eye-tracking hardware, which may not be accessible to all users. Furthermore, the system cannot accommodate users with severe eye impairments, and its dependency on substantial experimental data might limit its scalability to new contexts or tasks [9].

Eye Gaze Controlled Virtual Keyboard, presents a system enabling text input using eye gaze and blinking. This innovative approach is tailored for individuals with physical disabilities, particularly those unable to use traditional input methods. The system uses a webcam to detect the user's face and eyes, employing advanced techniques like Histogram of Oriented Gradients (HoG) and 68-point facial landmarks for precise detection. Eye movements are tracked to select keyboard sections, and blinking is utilized for character selection. The system ensures accessibility, allowing users to type without the need for hands, making it a significant contribution to assistive technology [10].

While the system demonstrates notable potential, several limitations exist. One significant drawback is the reliance on sequential key highlighting, which requires users to blink precisely when the desired key is highlighted. Missing the opportunity to blink during this interval necessitates waiting for the entire cycle to repeat, leading to inefficiencies and potential frustration. Additionally, the system's accuracy in gaze

detection may degrade under poor lighting conditions or for users wearing glasses due to light reflections. Further optimization, such as improved timing control or advanced gaze tracking with higher resolution cameras, could address these issues [10].

Table 1: Literature Survey

S.no	**Title**	**Author(s)**	**Journal & Year**	**Methodologies**	**Key Findings**	**Gaps**
1.	Netravaad: Interac- tive Eye Based Commu- nication System for Peo- ple With	Farajesh Kannan Megalingam , Sak- thiprasad Manoha- ran , Gokul Riju,	*IEEE, 2024*	The Netravaad system uses a portable hardware con- figuration with a USB cam- era, mini-PC, touch display, and speaker.	The system achieved 91% accuracy for alphabets, 100% for words, and 93% for num- bers, with the highest	The system's sensitivity to lighting conditions occasion- ally reduced detection accuracy. The GUI lacks

	Speech Issues [1]	Sreekanth Makkal Mohandas		The Sarani algorithm processes real-time video to detect eye movements by calculating horizontal, vertical, and blinking ratios. A unique eye-sign language, Netravaani, encodes alphabets, words, and numbers. Calibration aligns the user's face for optimal detection accuracy, and tests across distances (60–80 cm) validated the system's performance. These elements provide a robust foundation for addressing speech impairment challenges.	performance observed at a 70 cm user-camera distance. Response time averaged 0.36 seconds, ensuring real-time usability. The system performed reliably across age groups, with the 26–35 age group achieving the best results. It significantly reduced the need for human interpreters, offering an independent communication solution.	customization options like screen size adjustments, limiting flexibility. The Sarani algorithm relies on predefined rules, lacking adaptability through machine learning. Calibration remains a manual process, posing challenges for new users. Limited testing on users with severe physical impairments restricts insights into its performance under complex real-world conditions.

S.no	**Title**	**Author(s)**	**Journal & Year**	**Methodologies**	**Key Findings**	**Gaps**
2.	Translated Pattern-Based Bau-	Zakariyya Abdullahi	*IEEE, 2024*	The study utilizes a Tobii eye tracker to	The proposed method achieved an	The method has limitations in

	Eye-Writing Recognition Using Dilated Causal Convolution Network [2]	ture and Sunusi Bala Abdullah		capture gaze data from participants performing eye-writing tasks. Root translation is applied to align gaze patterns by normalizing initial gaze points, ensuring uniformity across samples. A Temporal Convolutional Network (TCN) is employed, incorporating three Dilated Causal Convolution (DCC) layers to handle long-range temporal dependencies and skip long eye fixations.	accuracy of 96.20% on a custom dataset of English letters and Arabic numerals, and outperformed traditional methods with accuracies of 98.81%, 97.76%, and 93.51% on HideMyGaze, Complex Gaze Gesture, and Japanese Katakana datasets, respectively. The system demonstrated high precision, recall, and F1-scores, effectively minimizing misclassifications and handling non-uniform eye-writing patterns.	handling gaze points outside the receptive field of the network, leading to potential misclassifications. Expanding the receptive field by adding DCC layers increases training time. The system is yet to be validated with real-world ALS patients, and the lack of public datasets with similar character types limits comparability with other approaches.

S.no	Title	Author(s)	Journal & Year	Methodologies	Key Findings	Gaps
3.	E-Gaze: Gaze Estimation With Event Camera [3]	Nealson Li , Muya Chang , and Arijit Raychowdhury	*IEEE, 2024*	The system utilizes event-based data from near-eye cameras to extract pupil features using image processing and kernel density estimation. A recurrent neural network (RNN) with Long Short-Term Memory (LSTM) layers processes temporal sequences of eye movements. A custom angular loss function is employed to optimize gaze estimation accuracy by minimizing angular errors. Region-of-interest (ROI) optimization is used to reduce computational overhead, ensuring low latency.	Achieved angular accuracy of 0.46° and sub-millisecond latency of 1.025 ms, outperforming state-of-the-art frame-based systems in real-time gaze estimation. Robust performance demonstrated across diverse subjects, achieving a median Intersection over Union (IoU) score of 0.8 for pupil feature extraction. Suitable for extended reality (XR) applications due to its high precision and low latency.	Requires per-session calibration due to inconsistent device positioning, which limits ease of use. System performance is validated only in controlled environments with fixed lighting and stable setups, leaving its robustness in real-world, dynamic conditions untested. Dependence on a fixed eye-to-display distance for the angular loss function reduces adaptability to different headset designs.

S.no	Title	Author(s)	Journal & Year	Methodologies	Key Findings	Gaps
4.	Blink-To-Liv eye-based commu- nication system for users with speech impair- ments [4]	eMohamed Ezzat, Mohamed Maged, Youssef Gamal, MustafaAd el, Mohammed Alrah- mawy& Sara El-Metwally	*Scientific Reports, 2023*	The Blink-To-Live system employs a mobile-based application that integrates com- puter vision techniques for real-time eye gesture detection and communica- tion. Facial landmarks are identified using a combination of Histogram of Oriented Gradi- ents (HOG) and Support Vector Machine (SVM) models to detect and track eye movements. The system recognizes four primary ges- tures—Blink, Left, Right, and Up—which are encoded into sequences of three states. These sequences are matched to a predefined command dic- tionary, and therecognized commands are converted into text and synthe- sized into lifelike speech using a Text-to-Speech module.	The system demonstrated high com- munication accuracy across diverse users, with participants achieving consistent recognition of gestures after adequate training. Average com- munication speed ranged from 10 to 25 seconds per command, with simpler commands being exe- cuted faster. The system proved cost- efficient and accessible, requiring only a smartphone with a camera and avoiding the need for specialized hardware. User adapt- ability was high, espe- cially for those with greater tech- nological awareness and educational backgrounds.	The system experienced occasional delays in gesture recognition, particularly with rapid transitions or complex sequences. It lacks the capability to detect downward eye ges- tures, limiting the range of possible commands. Performance is also heav- ily dependent on stable internet con- nectivity and precise eye movements, which may pose chal- lenges for users with advanced motor impair- ments. Additionally, individuals with low technology familiar- ity required longer train- ing periods to use the system effectively.

S.no	Title	Author(s)	Journal & Year	Methodologies	Key Findings	Gaps
5.	Uncertainty-Aware Gaze Tracking for Assisted Living Environments [6]	Paris Her, Logan Manderle, Philipe A. Dias	*IEEE, 2023*	Utilized a neural network regressor with facial keypoints for gaze estimation, integrated with Confidence Gated Units (CGUs) to handle occlusions and uncertainties. An angular Kalman filter was applied for temporal consistency, using uncertainty predictions to adjust observations dynamically. The system was validated on multiple datasets, including MoDiPro, MPIIFaceGaze, and Gaze360.	Achieved state-of-the-art performance on MoDiPro with a mean angular error of 21.7°, outperforming other methods by up to 36°. Demonstrated high correlation between predicted uncertainties and actual angular errors. CGUs improved accuracy by up to 3.12° in low-confidence scenarios, and the Kalman filter further reduced errors by 1.5° while enhancing temporal stability.	Limited performance under extreme conditions such as severe occlusions or poor lighting where keypoints cannot be detected. The method is restricted to 2D gaze estimation, potentially limiting its utility in scenarios requiring precise 3D gaze tracking. Computationally expensive due to reliance on pose estimation models like OpenPose, which may hinder real-time performance in resource-constrained environments.

S.no	Title	Author(s)	Journal & Year	Methodologies	Key Findings	Gaps
6.	Real-Time Human-Computer Interac- tion Using Eye Gazes [7]	Haodong Chena, Niloofar Zende-hdela , Ming C. Leua , Zhaozheng Yin	*Elsevier, 2023*	Combines the Dlib 68-point facial landmark detector for pre- cise eye tracking and a sclera-region-based method for gaze recognition. Blink detection uses eyelash dis- tance metrics, while the Mask R-CNN model enables instance segmentation of tools and parts with high pre- cision. These components are integrated into a real-time, visual software inter- face that allows users to interact through gaze inputs. The sys- tem is designed to operate efficiently with mini- mal hardware requirements, making it scal- able for various applications.	The system achieves 99% accuracy for eye gaze recognition within the safe distance of 40–60 cm and maintains robustness up to 160 cm with minor performance reductions. The Mask R- CNN instance segmenta-tion model achieves an average pre-cision, recall, and F1-score exceed- ing 99%, with certain object classes like pliers and prisms achieving per- fect 100% accuracy. Processing time is under 0.001 seconds per frame, demonstrat- ing real-time capability and efficiency. The integrated system pro- vides seamless interaction through eye gestures for object selec-tion and interface control.	Evaluation under varying environmen-tal conditions (e.g., light- ing changes, reflections, shadows) remains lim- ited. The system's accuracy decreases at distances beyond 160 cm. Object recognition is restricted to prede-fined classes, requiring retraining for new or more complex envi-ronments. These gaps highlight the need for enhanced adaptability and testing in real-world scenarios, particularly in dynamic envi-ronments or human-robot collaboration tasks.

S.no	Title	Author(s)	Journal & Year	Methodologies	Key Findings	Gaps
7.	Non- Intrusive Real Time Eye Track- ing Using Facial Align- ment for Assistive Tech- nologies [8]	C. Leblond- Menard and S. Achiche	*IEEE Trans- actions On Neural Systems and Rehabil- itation Engi- neering, Vol. 31, 2023*	The paper employs a four-step methodology: face detection, head pose cor- rection, eye patch extrac- tion, and gaze estimation using a CNN-based model. The pro- cess involves detecting facial land- marks to locate eye regions, applying nor- malization to reduce pose variability, and estimating gaze angles in real time with- out requiring user-specific calibration.	Achieved state-of- the- art angular accuracy with errors of 4.5° (MPIIGaze), 3.9° (UTMul- tiview), and 3.3° (Gaze- Capture). Compu- tational efficiency improved sig- nificantly, reducing inference time by up to 91%. The lightweight model ensures real- time operation on mobile and embedded systems with low power requirements, maintaining over 20 FPS on a single CPU core.	Limited by train- ing dataset biases, which can affect perfor- mance across diverse demo- graphics. Appearance- based models may strug- gle with extreme head poses or gaze angles outside the train- ing domain. Additionally, the lack of standardized benchmarking for com- putational require- ments hinders consistent comparisons across dif- ferent gaze estimation techniques.

S.no	Title	Author(s)	Journal & Year	Methodologies	Key Findings	Gaps
8.	A Data-Driven Frame-work for Intention Prediction via Eye Movement With Applica- tions to Assistive Systems [9]	Fatemeh Koochaki and Laleh Najafizadeh	*IEEE, 2021*	The frame- work integrates DBSCAN for clustering gaze points into regions of interest (ROIs), HMMs for temporal gaze sequence modeling, and CNNs for object detection using trans- fer learning. A CNN-LSTM model is employed for early inten- tion prediction, enabling proac- tive task identification. All components are designed to handle spatial and tempo- ral gaze data efficiently.	The frame-work achieves a high clas- sification accuracy of 97.42% for task prediction and 84.24% for early intention pre- diction after identifying the first two objects in the sequence. Object detec- tion is robust, with a preci- sion exceeding 90% for most objects. The framework successfully models gaze behavior, demonstrat- ing consistent temporal pat- terns across tasks.	The frame-work's performance is primarily validated in controlled kitchen sce- narios with limited tasks (3–4 objects), and its scal- ability to diverse, real-world environments is unexplored. It relies on high-precision eye track- ers, which may not be accessible to all users, and cannot accommodate individu- als with severe eye impairments. Additionally, the use of 2D images limits depth perception, which could be addressed with 3D environments.

S.no	Title	Author(s)	Journal & Year	Methodologies	Key Findings	Gaps
9.	Eye Gaze Controlled Virtual Keyboard [10]	Partha Chakrabort y, Dipa Roy, Md. Zahidur Rahman, Saifur Rehman	*Internati onal Jour- nal of Recent Tech- nology and Engi- neering, 2019*	The system employs real- time face and eye detection using HoG descriptors and 68-point facial landmarks, gaze tracking by ana- lyzing eyeball position, and blink detection using the Eye Aspect Ratio (EAR). The virtual key- board highlights keys sequen- tially, enabling users to select keys through intentional blinking.	The imple- mentation achieved A typing accuracy of 90.13%, with reliable detec- tion of gaze direction and intentional blinks. Face and eye region detection performed effectively under most conditions, offering a hands-free and acces- sible typing solution.	Gaze detec- tion accuracy decreases for users wearing glasses due to light reflec- tion, and the sequential key highlighting introduces delays, mak- ing typing slower com- pared to traditional methods. Additionally, the system's performance is sensitive to variations inlighting conditions.

3. Methodologies

The methodology employed for the development of Netravaad involved a combination of hard- ware and software approaches. A portable and adjustable hardware setup, including a touch display, USB camera, and mini-PC, was designed to capture and process eye movements. The researchers developed the Sarani algorithm, leveraging image processing techniques for detecting eye movements and translating them into commands. To define a structured sys- tem for communication, they introduced a unique eye-sign language called Netravaani, which encodes alphabets, words, and numbers through specific eye gestures. The system was evalu- ated through a series of controlled tests with volunteers across different age groups, focusing on accuracy, precision, and recall for detecting eye signs and forming messages. Data from these tests informed refinements to the algorithm and system design. Calibration techniques were applied to optimize detection, and predictive text was integrated to enhance usability. Ethical considerations, including informed consent and training, ensured the participants' involvement adhered to research protocols [1].

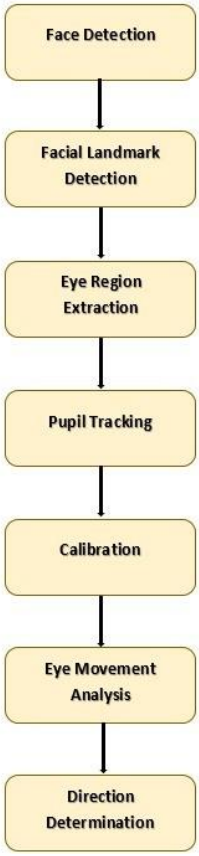

Fig. 3: Sarani Algorithm Steps

Figure 3 shows the steps involved in Sarani algorithm. Sarani algorithm is the backbone of the Netravaad system, designed to detect and interpret eye movements for communication. It preprocesses video frames from a USB camera, identifies facial landmarks, and tracks pupil positions to calculate horizontal (HR), vertical (VR), and blinking (BR) ratios. These ratios are mapped to predefined eye-sign patterns in the Netravaani language, enabling the formation of alphabets, words, and sentences. This algorithm provides a reliable and efficient solution for eye-based communication.

Algorithm 1 Pseudocode for Netravaad Methodology

1: **Input:** Video feed from camera

2: **Output:** Translated alphabets, words, or sentences

3: **Initialize:** Calibration parameters and Sarani algorithm

4: ▷ Calibration Phase

5: Adjust camera position to align the user's face within the frame.

6: Detect facial landmarks (eyes, nose, mouth).

7: Compute thresholds for pupil detection (horizontal and vertical ratios).

8: ▷ Eye Sign Detection

9: **while** System is active **do**

10: Capture video frames from the camera. 11: Preprocess frames (grayscale, resizing). 12: Detect eyes using facial landmarks.

13: Track pupil position to compute gaze ratios:

14: Horizontal Ratio (HR)

15: Vertical Ratio (VR)

16: Blinking Ratio (BR)

17: Determine gaze direction: *Center*, *Left*, *Right*, *Up*, *Down*.

18: **end while**

19: ▷ Pattern Matching and Recognition

20: **if** Eye sign sequence matches a predefined pattern **then**

21: Translate pattern to corresponding alphabet, word, or number.

22: Display output on screen and play audio.

23: **else**

24: Prompt user to retry or adjust input.

25: **end if**

26: ▷ Post-Processing

27: Allow user to correct errors or append additional characters.

28: Save the output and reset for the next input.

The Netravaad algorithm is designed to detect and interpret eye movements as a form of communication for individuals with speech disabilities. The process begins with the calibra- tion phase, where the system ensures proper alignment of the user's face within the camera's frame, followed by detecting facial landmarks, such as the eyes, nose, and mouth. The system then computes thresholds for accurate pupil detection using horizontal, vertical, and blinking ratios. During the eye sign detection phase, real-time video frames are captured, processed, and analyzed to track the user's pupil movements. Based on these movements, the system determines the direction of the gaze (e.g., left, right, up, down, or center). These movements are matched against predefined eye-sign patterns, which are then translated into correspond- ing letters, words, or numbers. The detected output is displayed on the screen and converted into speech.

Mathematical Functions

Netravaad algorithm uses several mathematical functions to track eye movements and calculate gaze directions. Below are the key functions used in the algorithm:

1. **Horizontal Eye Movement Ratio (HR)** This function calculates the horizontal position of the pupil relative to the center of the eye.

$$HR_{Left} = \frac{Pl_x}{E_{C_x}} \times 2^{-10}$$

$$HR_{Right} = \frac{Pr_x}{E_{C_x}} \times 2^{-10}$$

Where:

- Pl_x, Pr_x are the x-coordinates of the left and right pupils.
- E_{C_x} is the x-coordinate of the eye center.
- HR_{Left}, HR_{Right} are the horizontal ratios for the left and right eyes.

The combined horizontal ratio is:

$$HR = \frac{HRLeft + HRRight}{2}$$

2. **Vertical Eye Movement Ratio (VR)** This function calculates the vertical position of the pupil relative to the center of the eye.

$$V\,R_{Left} = \frac{Pl_y}{E_{C_y}} \times 2^{-10}$$

$$V\,R_{Right} = \frac{Pr_y}{E_{C_y}} \times 2^{-10}$$

Where:

- Pl_y, Pr_y are the y-coordinates of the left and right pupils.

- E_{C_y} is the y-coordinate of the eye center.
- $V\,R_{Left}$, $V\,R_{Right}$ are the vertical ratios for the left and right eyes.

The combined vertical ratio is:

$$V\,R = \frac{V\,RLeft + V\,RRight}{2}$$

3. **Blinking Ratio (BR)** The blinking ratio helps detect if the user's gaze is downward or if the eyes are closed. It is computed based on the eye width and height.

$$BR_{Left} = \frac{W_l}{H_l}$$
$$BR_{Right} = \frac{W_r}{H_r}$$

Where:

- W_l, H_l are the width and height of the left eye.
- W_r, H_r are the width and height of the right eye.
- BR_{Left}, BR_{Right} are the blinking ratios for the left and right eyes.

The combined blinking ratio is:

$$BR = \frac{BRLeft + BRRight}{2}$$

4. **Gaze Detection** The gaze direction is determined by comparing the calculated ratios to predefined thresholds for each direction (center, left, right, up, down). The system uses these ratios to classify the gaze direction, for example:
    - If HR is high and V R is centered, the user is gazing *right*.
    - If BR is high, the system detects a *blink* (indicating the "down" direction).

In the, Translated Pattern-based Eye-writing Recognition using Dilated Causal Convolu- tion Network, eye-writing patterns were captured using a Tobii eye tracker, which provided precise gaze coordinates from 20 participants writing 36 characters, including English letters and Arabic numerals. To address the non-uniformity in eye-writing caused by varied start- ing points and long eye fixations, a root translation technique was employed, shifting each gaze point to a uniform root. This translated pattern was then processed using a Tempo- ral Convolutional Network (TCN), enhanced with three stacked Dilated Causal Convolution (DCC) layers. The DCC layers, with varying dilation factors, allowed the model to capture long-range temporal dependencies and skip long eye fixations. The model was trained using the translated eye-writing data, and its performance was evaluated using accuracy, precision, recall, and F1-score on a newly designed dataset, as well as three existing public datasets. This approach aimed to improve recognition accuracy while handling non-uniform and complex eye-writing patterns [2].

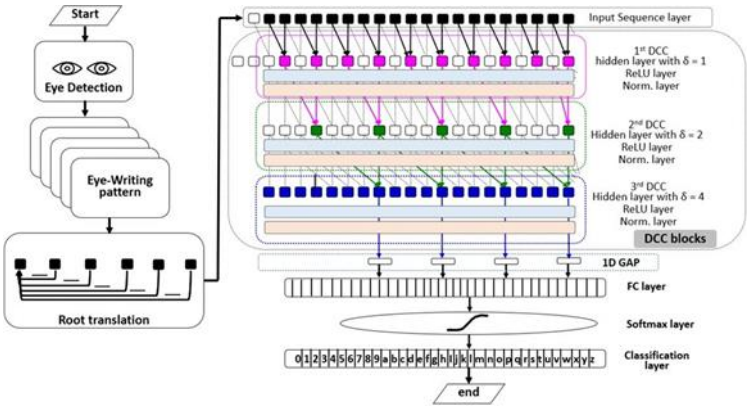

Fig. 4: Translated Pattern-Based Eye-Writing Architecture [2].

Figure 4 presents the architecture of the proposed eye-writing recognition system. It begins with gaze data collection using a Tobii eye tracker, followed by root translation to align gaze patterns to a uniform starting point. The translated data is processed by a Temporal Convolutional Network (TCN) with three Dilated Causal Convolution (DCC) layers, which capture long-range temporal dependencies and handle non-uniform patterns. The system employs

global average pooling and a fully connected layer, with softmax classification to predict the character class. This architecture efficiently addresses challenges like long eye fixations and pattern variations, enabling robust recognition.

Algorithm 2 Eye-Writing Recognition with Root Translation and Dilated Causal Convolution Network

1: **Input:** Eye gaze data from Tobii eye tracker (G_x, G_y) for each character

2: **Output:** Predicted character class

3:

4: **Step 1: Data Collection**

5: **for** each participant **do**

6: Capture eye gaze points for 36 characters (26 English letters + 10 Arabic numerals)

7: Store gaze points in (G_x, G_y) where G_x and G_y represent horizontal and vertical gaze coordinates

8: **end for**

9:

10: **Step 2: Root Translation**

11: **for** each eye-writing sample **do**

12: Let $G_0 = (G_{x0}, G_{y0})$ be the first gaze point

13: **for** each gaze point (G_{xi}, G_{yi}) in the sequence **do**

14: Translate gaze point: $\lambda_{xi} = G_{xi} - G_{x0}, \quad \lambda_{yi} = G_{yi} - G_{y0}$

15: **end for**

16: Store the translated gaze points as λ_x and λ_y

17: **end for**

18:

19: **Step 3: Temporal Convolutional Network with Dilated Causal Convo- lution**

20: Initialize Temporal Convolutional Network (TCN) with three stacked Dilated Causal Convolution (DCC) layers

21: **for** each translated eye-writing pattern (λ_x, λ_y) **do**

22: Pass the translated gaze points to the TCN

23: **for** each DCC layer **do**

24: Apply causal convolution with dilation factor δ

25: Skip long eye fixations using dilated receptive fields

26: Apply ReLU activation and normalization

27: **end for**

28: **end for**

29:

30: **Step 4: Classification**

31: After the final DCC block, apply global average pooling

32: Pass the pooled features to a fully connected (FC) layer

33: Apply softmax to predict the character class

34: Output the predicted character

The above algorithm outlines the methodology for eye-writing recognition using root translation and a Temporal Convolutional Network (TCN) with Dilated Causal Convolution (DCC) layers. The algorithm starts by collecting eye gaze data from participants using a Tobii eye tracker, capturing gaze

points for 36 characters, including both English letters and Arabic numerals. To address non-uniformity in eye-writing patterns, root translation is applied, where each gaze point is shifted relative to the first gaze point of the sequence, ensuring consistent starting positions. The translated gaze points are then passed through a TCN architecture consisting of three stacked DCC layers. These layers use causal convolutions with varying dilation factors to capture long-range temporal dependencies in the gaze data, while skipping long eye fixations. After processing through the DCC layers, the features are pooled, passed through a fully connected layer, and a softmax function is applied for classification. The output of this process is the predicted character corresponding to the eye-writing pattern.

Mathematical Functions

The algorithm uses several mathematical functions to track eye movements and calculate gaze directions. Below are the key functions used in the algorithm:

1. **Root Translation of Eye-Writing Patterns** This function adjusts the gaze coordinates to align with the root gaze point to ensure uniformity across eye-writing patterns.

$\lambda xi = Gxi - Gx0, \qquad \lambda yi = Gyi - Gy0$

Where:

- G_{xi}, G_{yi} are the horizontal and vertical gaze coordinates for the i-th gaze point.
- G_{x0}, G_{y0} are the coordinates of the initial gaze point (the root gaze point).
- λ_{xi}, λ_{yi} are the translated gaze points aligned with the root gaze point.

2. **Dilated Causal Convolution** The dilated causal convolution function is used to capture long-range temporal dependencies by skipping over certain gaze points, based on the dilation factor δ.

$$DCC(\lambda \star \delta k)(t) = \sum_{i=0}^{i} k(i) \cdot \lambda_{t-\delta \cdot i}$$

Where:

- λ represents the translated eye-writing pattern.
- k(i) is the convolution filter applied to the gaze points.
- δ is the dilation factor that determines how much the receptive field skips between gaze points.
- The sum accumulates the values of gaze points at different steps defined by the dilation factor, capturing long-range dependencies.

3. **Receptive Field of Dilated Convolution** The receptive field of a dilated convolution layer is calculated based on the number of layers ζ and the filter size e.

$r = 2\zeta - 1 \cdot e$

Where:

- r is the receptive field, which defines how many previous gaze points are considered during convolution.
- ζ is the number of layers in the convolutional network.
- e is the filter size used in the convolution operation.

The methodology of the E-Gaze system involves two primary stages: eye tracking and gaze estimation. First, the system processes the event-based data stream from a near-eye event camera to extract pupil features. Events are accumulated asynchronously based on their spatial and temporal distributions, capturing rapid eye movements such as saccades and blinks. These features, represented as ellipses, are then fed into a recurrent neural network (RNN) that uses Long Short-Term Memory (LSTM) layers to analyze pupil motion over time. To improve gaze estimation, the system utilizes a custom angular loss function that minimizes the angular error between the estimated and actual gaze directions. The system operates entirely on event data, without relying on additional frame-based or infrared data, achieving high accuracy (0.46°) and low latency (under 1 ms) [3].

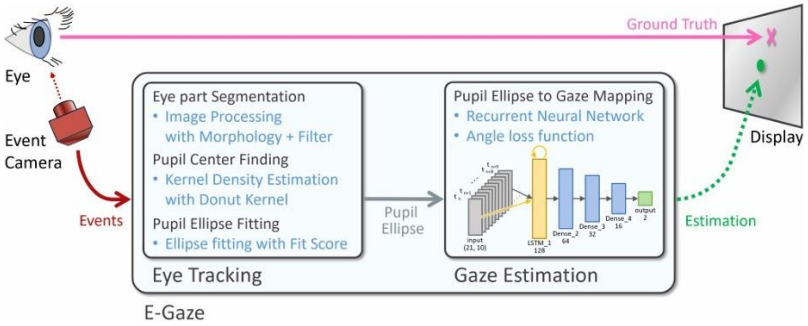

Fig. 5: Overview of the E-Gaze system architecture [3].

Figure 5 illustrates the architecture of the E-Gaze system, which is designed for real- time gaze estimation using event cameras. The system begins with the Eye Tracking Block, which processes the asynchronous event data generated by a near-eye event camera to extract pupil features. These features are represented as ellipses, encoding spatial and temporal information about the pupil's motion. The extracted sequence of ellipses is then fed into the Gaze Estimation Block, which employs a recurrent neural network (RNN) architecture to infer gaze direction. By analyzing temporal patterns in the pupil motion, the RNN predicts the display coordinates corresponding to the user's point of gaze. This modular architecture effectively combines high-temporal-resolution data processing with advanced neural network- based inference, ensuring precise and low-latency gaze estimation.

Algorithm 3 E-Gaze System Methodology

Input: Event data stream $\{e_i\}_{i=1}^{n}$
Output: Gaze direction (x_d, y_d)
Step 1: *Eye Tracking Block*
  Accumulate event sets $E_{set} = \{e_i, e_{i+1}, \ldots, e_{i+2000}\}$
  Extract pupil features F_{pupil} using event distributions and spatial-temporal characteristics
  Fit pupil ellipses to extract center coordinates (x_p, y_p), height h_p, width w_p, and rotation ϕ_p
  Repeat for each event set in the stream
Step 2: *Gaze Estimation Block*
  For each sequence of pupil features $\{F_{pupil}\}$, feed to RNN with LSTM layers
  Process sequence of N pupil features $\{F_{pupil}^{1}, F_{pupil}^{2}, \ldots, F_{pupil}^{N}\}$

  For each time step t:
    Update RNN hidden state $h_t = \text{LSTM}(h_{t-1}, F_{pupil}^{t})$
  Use fully connected layers to output gaze coordinates (x_d, y_d)
  Step 3: *Loss Function Optimization*
  Minimize angular loss function:

$$L_{angular} = \frac{1}{n} \sum_{i=1}^{n} \theta_{error}(u_i, v_i)^2$$

where $\theta_{error}(u_i, v_i)$ is the angular error between the estimated gaze direction vector

u_i and the ground truth vector v_i.

Repeat until convergence

The above algorithm outlines the methodology of the E-Gaze system, which is designed for real-time gaze estimation using event-based data. The process begins with the Eye Tracking Block, where the system accumulates event data into sets and extracts pupil features by analyzing the spatial and temporal distributions of the events. These features are used to fit ellipses that represent the pupil's position and shape. Next, in the Gaze Estimation Block, the sequence of pupil features is fed into a recurrent neural network (RNN) with Long Short- Term Memory (LSTM) layers. The RNN processes the pupil motion over time to predict gaze coordinates on the display. The network is trained by minimizing an angular loss function, which computes the angular error between the predicted and ground truth gaze directions. The algorithm iterates through these steps, refining its gaze predictions based on the input event stream, and optimizing the system's accuracy using the angular loss function. This method ensures that the system can provide precise gaze estimates with minimal latency, suitable for applications like extended reality (XR).

Mathematical Functions

The E-Gaze algorithm uses several mathematical functions to track eye movements and estimate gaze directions. Below are the key functions used in the algorithm:

1. **Angular Error Function (Angle between Gaze Vectors)** This function calculates the angular error between the estimated gaze direction and the ground truth gaze direction.

$$\theta_{error}(u_i, v_i) = \arccos\left(\frac{u_i \cdot v_i}{\|u_i\|\|v_i\|}\right)$$

- u_i is the estimated gaze direction vector (from the eye to the estimated gaze point on the display).
- v_i is the ground truth gaze direction vector (from the eye to the actual gaze point on the display).
- $\|u_i\|$ and $\|v_i\|$ are the magnitudes of the gaze vectors.
- $u_i \cdot v_i$ is the dot product of the estimated and ground truth gaze direction vectors.

2. **Angular Loss Function** The angular loss function is used to minimize the angular error between the estimated and actual gaze directions over multiple samples.

$$L_{angular} = \frac{1}{n}\sum_{i=1}^{n} \theta_{error}(u_i, v_i)^2$$

Where:

- n is the number of gaze estimation samples.
- $\theta_{error}(u_i, v_i)$ is the angular error between the estimated and ground truth gaze direction vectors for the i-th sample.

The goal of this function is to minimize the mean squared angular error to improve gaze estimation accuracy.

The methodology of the Blink-To-Live system revolves around a mobile-based applica- tion leveraging computer vision techniques to enable communication for individuals with speech impairments. The system employs a smartphone camera to capture real-time video frames, which are sent to a backend for processing. Key modules include facial landmark detection, eye tracking, and gesture recognition, where four eye gestures—Blink, Left, Right, and Up—serve as the foundational alphabets. These gestures are combined into sequences of three states to encode over 60 daily life commands. The backend uses Histogram of Ori- ented Gradients (HOG) with a Support Vector Machine (SVM) for facial detection and the Eye Aspect Ratio (EAR) for blinking recognition. Recognized commands are translated into text and synthesized into lifelike speech in the user's native language. The system prioritizes simplicity and cost-efficiency, ensuring accessibility by eliminating the need for specialized hardware while delivering flexible, real-time communication [4].

Figure 6 illustrates the architecture of the Blink-To-Live communication system, show- casing its key components and workflow. The system consists of two main modules: a mobile application built using the Flutter framework and a Python-based backend for real-time image analysis. The process begins with a caregiver activating the smartphone camera to capture video frames, which are sent to the backend via a web socket for processing. The backend performs facial landmark detection to localize the eyes and track movements based on pre- defined gestures—Blink, Left, Right, and Up. These gestures are encoded into sequences of three states, which are matched to a command dictionary. The recognized commands are translated into the user's native language and converted to lifelike speech through a Text-to- Speech module. The resulting text and audio are displayed on the mobile application, enabling seamless communication. This architecture ensures real-time performance, flexibility, and cost efficiency by eliminating the need for specialized hardware.

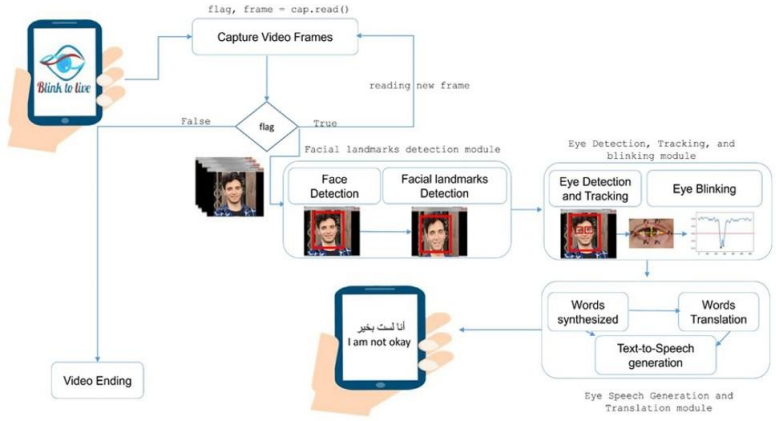

Fig. 6: Blink-To-live Communication System Architecture [4].

Algorithm 4 Blink-To-Live Eye-Tracking Communication System

1: **Input:** Real-time video frames captured by smartphone camera

2: **Output:** Translated text command and synthesized speech

3: Initialize the mobile application

4: Start the camera to capture video frames in real-time

5: **while** frames are being captured **do**

6: **Step 1: Facial Landmark Detection**

7: Detect face using Histogram of Oriented Gradients (HOG) + SVM

8: Extract 68 facial landmarks (eyes, mouth, nose, etc.)

9: **Step 2: Eye Detection and Gesture Recognition**

10: Localize and track left and right eyes using facial landmarks

11: Compute Eye Aspect Ratio (EAR) to detect blinking

12: Recognize eye gestures (Left, Right, Up, Blink)

13: **Step 3: Command Encoding**

14: Store recognized gestures in a sequence of three states

15: **if** sequence matches predefined dictionary **then**

16: Retrieve corresponding command from dictionary

17: **else**

18: Prompt for correction or retry

19: **end if**

20: **Step 4: Translation and Speech Synthesis** 21: Translate command into user's native language

22: Synthesize speech using Text-to-Speech module

23: Display text and play synthesized speech on screen

24: **end while**

25: End application

The Blink-To-Live algorithm enables real-time communication by processing video frames captured through a smartphone camera. Facial landmarks are detected using the HOG + SVM model, and eye gestures—Blink, Left, Right, and Up—are identified through Eye Aspect Ratio (EAR) analysis. These gestures are recorded in sequences of three states and matched to a predefined dictionary of commands. Recognized commands are translated into the user's native language and synthesized into speech using a Text-to-Speech module. The output is displayed and played on the device, providing a simple and cost-efficient communication solution.

Mathematical Functions

The Blink-To-Live algorithm uses several mathematical functions to track eye movements and detect blinking. Below is the key function used to calculate the Eye Aspect Ratio (EAR):

1. **Eye Aspect Ratio (EAR)** This function calculates the ratio of eye aspect to detect blinking by analyzing the distance between specific eye landmarks.

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Where:

- $p_1, p_2, p_3, p_4, p_5, p_6$ represent the coordinates of six key landmarks around the eye.
- $||p_i - p_j||$ is the Euclidean distance between two points.

The EAR value is constant when the eye is open, and it drops significantly when the eye blinks. A threshold value t (e.g., 0.2) is used to determine whether the eye is open or closed.

In the, Uncertainty-aware gaze tracking for assisted living environments,gaze tracking approach leverages facial keypoint detection from a human pose estimation model to estimate gaze direction. A neural network regressor uses the relative positions of keypoints (eyes, nose, and ears) to predict the gaze direction in a 2D image plane. To handle uncertainties arising from occlusions or low-confidence detections, Confidence Gated Units (CGUs) are integrated into the network, reducing the impact of unreliable keypoint predictions. The model also provides an estimate of uncertainty for each gaze prediction, which is used in an angular Kalman filter to improve temporal consistency and tracking accuracy. The Kalman filter combines past gaze estimates with new predictions, adjusting the influence of each based on the uncertainty values. This methodology is evaluated on real-world datasets from assisted living environments, as well as publicly available datasets, demonstrating its robustness and ability to accurately track gaze over time [6].

Algorithm 5 Uncertainty-Aware Gaze Tracking with Kalman Filter

Input: Video frames with facial keypoints (x_i, y_i, c_i), where i denotes facial keypoints, and their confidence scores c_i.

Output: Predicted gaze direction and uncertainty for each frame.

Initialize Kalman Filter with initial gaze state $s_0 = [\rho_0, \omega_0]$.

Initialize neural network model NN to predict gaze direction and uncertainty from facial keypoints.

for each frame in video **do**

Detect facial keypoints using pose estimation model.

Normalize the keypoints' positions to obtain relative coordinates.

Obtain the confidence levels of the detected keypoints.

Feed the keypoints into the neural network model NN to predict gaze direction g_j and uncertainty σ_j.

if confidence of keypoints is low **then**

Apply Confidence Gated Units (CGUs) to adjust the contribution of low- confidence keypoints.

end if

Use uncertainty σ_j to adjust the Kalman filter observation model:

Set observation covariance $\sigma_v = e^{-\sigma}j$ or $\sigma_v = 1/\sigma_j$.

Predict the new gaze state $\hat{s}_t$ using the Kalman filter:

$\hat{s}_t = F \cdot s_{t-1} + w_t$ where F is the state transition matrix and w_t is process noise.

Update the gaze prediction using the Kalman filter:

$s_t = s_{t-1} + K_t \cdot (z_t - H \cdot s_{t-1})$ where K_t is the Kalman gain.

Update the Kalman filter state with the new gaze direction g_t and angular velocity ω_t.

Store the updated gaze prediction and uncertainty for the current frame.

end for

Return: Sequence of gaze directions and uncertainties for all frames.

The algorithm presented above outlines the process of uncertainty-aware gaze tracking using a neural network and an angular Kalman filter for temporal integration. First, for each frame in a video, facial keypoints (such as eyes, nose, and ears) are detected using a pose estimation model. The relative positions of these keypoints are normalized, and their associ- ated confidence levels are computed. The neural network then predicts the gaze direction and uncertainty based on these keypoints. If the confidence in the keypoints is low due to occlu- sions or poor visibility, Confidence Gated Units (CGUs) are applied to adjust the influence of these unreliable keypoints. The uncertainty from the neural network is then used to adjust the observation model of the Kalman filter by modifying the observation covariance. The Kalman filter predicts the gaze state, which includes the gaze direction and angular velocity, by combining the new predictions with the previous state, with adjustments made based on the uncertainty. This process ensures that the gaze predictions are temporally consistent and robust. The final output is a sequence of gaze directions and their associated uncertainties for each frame in the video, providing an accurate and stable gaze tracking solution.

Mathematical Functions

The proposed gaze tracking algorithm uses several mathematical functions to track gaze direction and estimate uncertainty. Below are the key functions used in the algorithm:

1. **Facial Keypoint Normalization** The keypoints of the face are normalized relative to the head centroid:

$$\hat{x}_{j,k,s} = \frac{x_{j,k,s} - x_{j,h}}{x_{j,m}}, \quad \hat{y}_{j,k,s} = \frac{y_{j,k,s} - y_{j,h}}{y_{j,m}}$$

- $x_{j,k,s}$, $y_{j,k,s}$ are the original coordinates of the k-th keypoint of the j-th subject.
- $x_{j,h}$, $y_{j,h}$ are the coordinates of the head centroid.
- $x_{j,m}$, $y_{j,m}$ are the distances from the head centroid to the furthest keypoint.

2. **Gaze Direction Estimation** The neural network estimates the gaze direction in the 2D image plane:

$$g_{jx} = \sin(\rho_j), \quad g_{jy} = \cos(\rho_j)$$

Where:

- g_{jx}, g_{jy} are the components of the gaze direction vector.
- ρ_j is the apparent gaze angle with respect to the horizontal axis.

3. **Kalman Filter State Prediction** The Kalman filter predicts the current gaze state based on the previous state:

$$\hat{s}_t = F \cdot s_{t-1} + w_t$$

Where:

- $\hat{s}_t$ is the predicted state (gaze direction and angular velocity).
- F is the state transition matrix.
- w_t is the process noise, assumed to be normally distributed.

4. **Kalman Filter Update** The Kalman filter updates the state s_t based on the new gaze prediction z_t:

$$s_t = s_{t-1} + K_t \cdot (z_t - H \cdot s_{t-1})$$

Where:

- s_t is the updated state (gaze direction and angular velocity).
- K_t is the Kalman gain.
- z_t is the new gaze prediction.
- H is the observation matrix.

The Kalman gain K_t is computed as:

$$K_t = \frac{P_{t-1}}{P_{t-1} + R_t}$$

Where:

- P_{t-1} is the error covariance from the previous state.

- R_t is the observation noise covariance.

5. **Observation Covariance Adjustment** The uncertainty σj is used to adjust the observation covariance in the Kalman filter:

$$\sigma_v = \frac{1}{\sigma_j} \quad \text{or} \quad \sigma_v = e^{-\sigma_j}$$

Where:

- σ_v is the observation covariance.
- σ_j is the predicted uncertainty for the gaze direction.

Real-Time Human-Computer Interaction Using Eye Gazes, describes a webcam-based system that facilitates real-time eye gaze recognition and object segmentation. The system employs the Dlib 68-point facial landmark detector to track and identify eye regions, focusing on the sclera to determine gaze directions (straight, left, right) and blinking through sclera ratios and eyelash distance metrics. For object segmentation, a Mask Region-Based Convolutional Neural Network (Mask R-CNN) is trained using annotated datasets with polygon masks to recognize and segment tools and parts. These elements are integrated into a user- friendly visual interface, enabling seamless interaction where users select segmented objects using their eye gazes. The system is designed for real-time operation, ensuring high accuracy and robustness with standard RGB camera hardware [7].

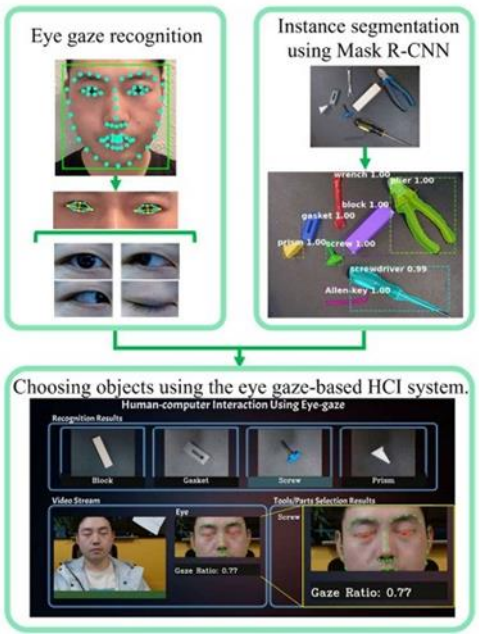

Fig. 7: Real-Time Human-Computer Interaction Using Eye Gazes System Overview [7].

Figure 7 illustrates the architecture of the real-time eye-gaze-based human-computer interaction (HCI) system. The system integrates three core components: real-time eye track- ing and gaze recognition, object recognition using instance segmentation, and a visual software interface for interaction. The input from a standard RGB webcam captures facial landmarks and eye regions, which are analyzed to detect and track gaze directions (left, right, straight) and blinks. Simultaneously, the Mask R-CNN model processes image data to iden- tify and segment tools and parts with high accuracy. These components are combined into a seamless software interface, allowing users to interact by selecting objects using eye gestures. The system's design emphasizes efficiency and scalability, making it suitable for practical real-time applications.

Algorithm 6 Real-Time Eye Gaze Recognition and Object Segmentation

1: **Input:** RGB Webcam Frames

2: **Output:** Recognized Eye Gaze and Segmented Objects

3: **Step 1: Eye Detection and Tracking**

4: Load the Dlib 68-point Facial Landmark Detector.

5: Detect facial landmarks and extract eye regions (landmarks 37–48).

6: Define Regions of Interest (ROI) based on sclera and eyelash positions.

7: **Step 2: Eye Gaze Recognition**

8: Convert ROI to grayscale and binary scale.

9: Calculate sclera area ratios to determine gaze direction:

10: **if** $\phi < 0.70$ **then**

11: Gaze is to the left.

12: **else if** $0.70 \leq \phi \leq 1.20$ **then**

13: Gaze is straight ahead.

14: **else**

15: Gaze is to the right.

16: **end if**

17: Detect blinking by calculating eyelash distances:

18: **if** $\psi \geq 5.50$ for at least 15 frames **then**

19: Gaze is a blink.

20: **end if**

21: **Step 3: Instance Segmentation**

22: Train a Mask R-CNN model with annotated datasets of tools and parts.

23: Input webcam frame to the model for segmentation.

24: Output segmented objects with bounding boxes, masks, and class labels.

25: **Step 4: Integration into Software Interface**

26: Visualize real-time gaze recognition and segmentation.

27: Allow user interaction via gaze (e.g., blink to select segmented object).

28: **Step 5: Output Results**

29: Display selected tools and parts based on user gaze input.

The above algorithm for real-time eye gaze recognition and object segmentation integrates eye tracking, gaze recognition, and instance segmentation into a cohesive process. First, the Dlib 68-point facial landmark detector is utilized to identify facial features and extract eye regions. These regions are processed to calculate sclera area ratios, which determine the direction of gaze—left, straight, or right—based on predefined thresholds. Blinking is detected by analyzing the relative distances between eyelash landmarks over consecutive frames. For object segmentation, a Mask R-CNN model is trained using annotated datasets, enabling accurate segmentation of tools and parts in the input frame. The recognized gaze directions and segmented objects are integrated into a visual software interface, allowing users to interact with the system through gaze inputs. For instance, users can select segmented objects by blinking. The algorithm ensures real-time processing and robust performance, even with minimal hardware requirements, enabling natural and efficient human-computer interaction.

Mathematical Functions

Real-Time Human-Computer Interaction Using Eye Gazes algorithm uses several mathe- matical functions to track eye movements and calculate gaze directions. Below are the key functions used in the algorithm:

1. **Gaze Direction Ratio (ϕ)** This function calculates the horizontal direction of the gaze based on the visible sclera areas in the left and right parts of the eye.

$$\phi = \frac{\text{Left Sclera Pixels}}{\text{Right Sclera Pixels}}$$

Where:

- Left Sclera Pixels: The number of white pixels on the left side of the eye.
- Right Sclera Pixels: The number of white pixels on the right side of the eye.
- ϕ is the gaze ratio that determines the gaze direction.

The gaze direction is determined as follows:

- If $\phi < 0.70$, the gaze is considered to be looking left.
- If $0.70 \leq \phi \leq 1.20$, the gaze is considered straight ahead.
- If $\phi > 1.20$, the gaze is considered to be looking right.

2. **Blink Detection Ratio (ψ)** This function detects blinking by comparing the length of the vertical and horizontal lines connecting eye landmarks.

$$\psi = \frac{\text{Horizontal Line Length}}{\text{Vertical Line Length}}$$

Where:

- Horizontal Line Length: The distance between the landmarks on the left and right sides of the eye.
- Vertical Line Length: The distance between the upper and lower eyelash landmarks.
- ψ is the ratio that helps detect blinking.

A blink is detected if $\psi \geq 5.50$ for at least 15 frames (0.5 seconds).

Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive Technologies employs a four-step methodology for gaze estimation: face detection, head pose correction, eye patch extraction, and iris detection for gaze estimation. The process begins with face detection to locate the user's face in the image, followed by extracting facial landmarks to identify the eye positions. Head pose correction is then applied to normalize the eye images by removing the roll component, ensuring consistent gaze estimation input. Eye patches are extracted from the normalized images and passed through a CNN-based gaze estimation model. The model detects the iris and estimates gaze angles (pitch and yaw), providing the gaze origin. This method delivers real-time gaze estimation without requiring person-specific calibration, making it computationally efficient and suitable for both indoor and outdoor environments [8].

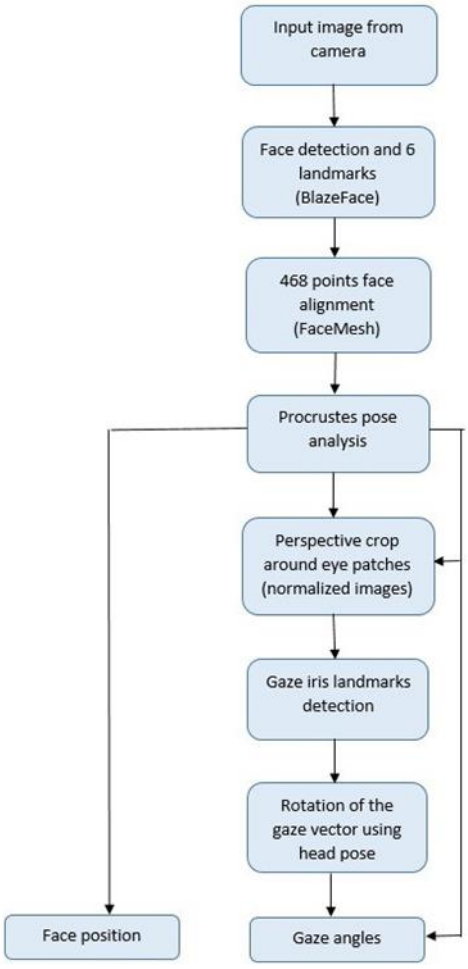

Fig. 8: Workflow of Non-Intrusive Real Time Eye Tracking.

Figure 8 illustrates the sequential workflow employed in Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive Technologies. The process begins with face detection, identifying the user's face and extracting key facial landmarks. These landmarks are used to estimate the head pose, which plays a critical role in normalizing the eye patch images. This normalization involves correcting the roll component of the head pose to ensure consistent input for gaze estimation. The normalized eye patches are then fed into the gaze estimation model, which calculates gaze angles and provides the corrected gaze direction.

The figure emphasizes the systematic progression of tasks, highlighting how each step builds upon the previous one to achieve real-time and accurate gaze tracking.

Algorithm 7 Non-Intrusive Real-Time Eye Tracking Methodology

1: **Input:** Image containing user's face

2: **Output:** Gaze angles (pitch, yaw), gaze origin

3:

4: **Step 1: Face Detection**

5: Detect the face in the input image

6: Extract facial landmarks (eyes, nose, mouth, ears)

7:

8: **Step 2: Head Pose Correction**

9: Calculate head pose using facial landmarks

10: Normalize the eye region by removing the roll component of the head pose

11:

12: **Step 3: Eye Patch Extraction**

13: Extract eye patches from the image based on normalized facial landmarks

14: Crop and resize the eye patches for input to gaze estimation model

15:

16: **Step 4: Iris Detection and Gaze Estimation**

17: Apply CNN-based gaze estimation model to detect the iris

18: Estimate the gaze direction using the iris position and facial landmarks

19: Output the gaze angles (pitch, yaw) and gaze origin

The above algorithm outlines the step-by-step methodology for Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive Technologies. It begins by detecting the user's face within the input image and extracting key facial landmarks such as the eyes, nose, mouth, and ears. These landmarks are then utilized to calculate the head pose, followed by normalization to remove the roll component, ensuring consistency across the input data. The next step involves extracting and resizing eye patches based on the normalized facial landmarks, preparing them for gaze estimation. Finally, a CNN-based model detects the iris and estimates the gaze direction, outputting accurate gaze angles (pitch and yaw) and the gaze origin. The sequential nature of the algorithm ensures efficiency and real-time processing, making it suitable for practical applications in assistive technologies.

Mathematical Functions

Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive Technologies uses several mathematical functions to track eye movements and calculate gaze directions. Below are the key functions used in the algorithm:

1. Head Pose Correction This function normalizes the eye image by removing the roll component of the head pose.

$\vec{g} = H \cdot \vec{g_e}$

Where:

- $\vec{g}$ is the world coordinates gaze direction.
- H is the head pose matrix.
- $\vec{g}_e$ is the eye gaze direction with respect to the face.

2. **Eye Position Calculation** This function calculates the eye position in world coordinates using the head pose and the eye center from the canonical model.

$$\vec{e} = H \cdot \vec{e_c}$$

Where:

- $\vec{e}$ is the eye center position with respect to the camera.
- H is the head pose matrix.
- $\vec{e_c}$ is the eye center in the canonical model.

3. **Perspective Transformation** This function applies a perspective transformation to correct the image, removing head pose and camera parameter variations.

$$W = \frac{\|\vec{e}\|}{\|\vec{e}^*\|} \cdot \frac{l^*}{l}$$

Where:

- W is the perspective transformation matrix.
- $\vec{e}$ is the adjusted eye position.
- l^* and l are the focal lengths used for transformation.
- $\vec{e}^*$ is the reference eye position.

4. **Gaze Angle Error** This function computes the angular error between the real gaze direction and the estimated gaze direction using cosine similarity.

$$e = \cos^{-1}\left(\frac{\vec{g} \cdot \vec{g}_e}{\|\vec{g}\|\|\vec{g}_e\|}\right)$$

Where:

- $\vec{g}$ is the real gaze direction.
- $\vec{g}_e$ is the estimated gaze direction.
- e is the angular error between the real and estimated gaze directions.

A Data-Driven Framework for Intention Prediction via Eye Movement With Applications to Assistive Systems, employs a data-driven approach to predict user intentions based on eye movement patterns. Eye tracking is used to capture the gaze points of users on displayed images, which are processed using the DBSCAN clustering algorithm to identify regions of interest (ROIs) based on gaze density. Temporal patterns of gaze are analyzed using hidden Markov models (HMMs) to determine the sequence of object selections. Transfer learning is utilized with pre-trained convolutional neural networks (CNNs) to identify objects within the ROIs. The methodology includes two main phases: task classification for predicting intended versus unintended tasks and early intention prediction that uses both spatial and temporal gaze data to forecast the user's task in advance. This approach ensures high accuracy and adaptability for real-world applications, particularly in assistive technologies [9].

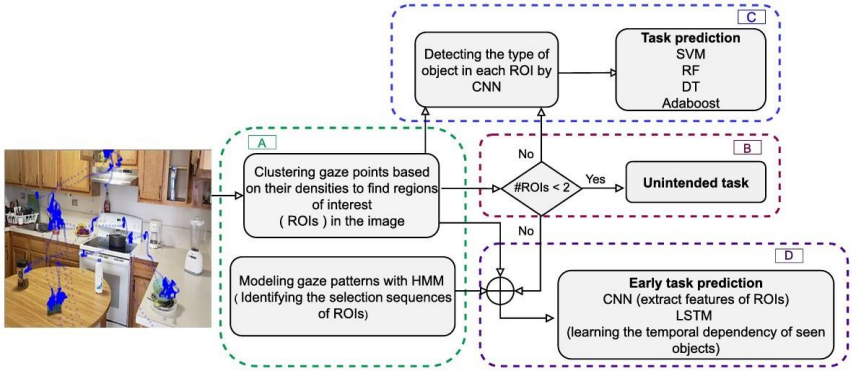

Fig. 9: Workflow of Non-Intrusive Real Time Eye Tracking.

Figure 9 illustrates the proposed framework for predicting user intention by leverag- ing spatial and temporal patterns of eye movement. The framework is composed of four primary modules, each addressing a critical aspect of the prediction process. The first mod- ule clusters gaze points using DBSCAN

to identify regions of interest (ROIs) and analyzes the sequence of these regions via Hidden Markov Models (HMMs). The second module dis- tinguishes intended tasks from unintended tasks based on the number of detected ROIs. The third module predicts the type of intended task using a CNN-based object detection mechanism to classify objects within the ROIs. Lastly, the fourth module focuses on early task prediction by combining spatial and temporal gaze data with a CNN-LSTM model, enabling proactive intention forecasting. This modular approach ensures high accuracy and adaptability for assistive technology applications.

Mathematical Functions

A Data-Driven Framework for Intention Prediction via Eye Movement With Applications to Assistive Systems uses several mathematical functions to analyze eye movement and predict user intention. Below are the key functions used in the framework:

1. **DBSCAN Clustering for Eye Movement Data** This function clusters gaze points based on spatial density to identify regions of interest (ROIs).

$$\text{Cluster}(P) = \{q \mid \text{Dist}(P, q) \leq \epsilon \text{ and } \text{MinPts}(P) \geq \text{minP ts}\}$$

Where:

- P, q are gaze points.
- ϵ is the radius of the neighborhood for clustering.
- MinPts(P) is the minimum number of points required to form a cluster.
- Dist(P, q) is the distance between points P and q.

2. **Hidden Markov Model (HMM) for Temporal Gaze Sequences** This function models the temporal sequence of gaze points by representing gaze transitions as hidden states in the HMM.

$$P(O|\lambda) = \sum_{q_1, q_2, \ldots, q_T} P(O, q_1, q_2, \ldots, q_T |\lambda)$$

Where:

- O represents the observed gaze data.
- λ represents the model parameters (transition and emission probabilities).
- $q_1, q_2, \ldots, q_T$ are the hidden states (ROIs).

The forward algorithm for computing the probability of observing a sequence is:

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, q_t = i|\lambda)$$

Where:

- $\alpha_t(i)$ is the probability of observing the sequence up to time t and being in state I at time t.

Algorithm 8 Data-Driven Framework for Intention Prediction

1: **Input:** Eye movement data (gaze points, timestamps), displayed images 2: **Output:** Predicted user intention (task classification or early prediction) 3: **Step 1: Clustering Eye Movement Data**

4: Use DBSCAN to cluster gaze points based on spatial density

5: Identify Regions of Interest (ROIs) for each trial

6: **Output:** ROIs for each gaze pattern

7: **Step 2: Temporal Sequence Analysis**

8: Use Hidden Markov Models (HMM) to analyze gaze sequence

9: Model transitions between ROIs to capture temporal patterns

10: **Output:** Temporal sequence of object selection

11: **Step 3: Object Identification with CNN**

12: Apply pre-trained CNN (ResNet50) to identify objects within ROIs

13: Fine-tune the model with trial-specific images for better object detection

14: **Output:** Identified objects in each ROI

15: **Step 4: Task Classification**

16: Classify tasks as intended or unintended based on number of ROIs

17: **if** Number of ROIs ¿ 2 **then**

18: Predict task as intended

19: **else**

20: Predict task as unintended

21: **end if**

22: **Step 5: Early Intention Prediction (CNN-LSTM)**

23: Combine CNN features with Long Short-Term Memory (LSTM) for early task prediction

24: Use sequence of ROI images to predict task intention at early stages

25: **Output:** Early prediction of user intention

The avove algorithm outlines a systematic approach to analyze eye movement data and predict user intention. The framework integrates multiple techniques to ensure robust pre- dictions. Initially, gaze points are clustered using DBSCAN to identify Regions of Interest (ROIs), which represent areas of visual attention. Temporal patterns within these ROIs are then modeled using Hidden Markov Models (HMM) to capture transitions and sequences in gaze behavior. To identify specific objects within the ROIs, a Convolutional Neural Network (CNN), such as a fine-tuned ResNet50, is employed, enhancing object detection accuracy by leveraging trial-specific image training. For task classification, the framework analyzes the number of ROIs: tasks with multiple ROIs are classified as intended, while those with fewer are deemed unintended. Finally, CNN-LSTM models are used for early intention prediction, combining spatial features from CNNs with temporal dependencies captured by LSTMs, enabling the system to anticipate user intentions in real-time. This multi-step framework effectively bridges eye movement data and predictive user intention modeling.

The eye gaze-controlled virtual keyboard system, comprises several key steps: face detec- tion, eye detection, gaze tracking, blinking detection, and virtual keyboard interaction. Initially, a webcam captures real-time video, and the system uses Histogram of Oriented Gradients (HoG) with dlib-based 68-point facial landmarks for accurate face and eye detec- tion. Eye gaze is tracked by analyzing the position of the eyeballs, determining whether the user is looking left, right, or center, to select sections of the virtual keyboard. Blinking is detected by monitoring the closure of the eyelids through vertical and horizontal line inter- section techniques, which differentiate between involuntary blinks and intentional blinks for key selection. A virtual keyboard with sequential key highlighting enables the user to type by blinking when the desired key is lit. Each component is optimized to work in real-time, providing a hands-free text input method tailored for individuals with physical disabilities [10].

Algorithm 9 Eye Gaze Controlled Virtual Keyboard

1: **Input:** Live video stream from webcam

2: **Output:** Typed text using eye gaze and blink detection

3: **procedure** EyeGazeKeyboard

4: Initialize video capture from webcam

5: **while** Video stream is active **do**

6: **Step 1: Face Detection**

7: Detect face using HoG + dlib 68-point landmarks

8: **if** Face Detected **then**

9: Extract eye region (left and right eye)

10: **Step 2: Eye Gaze Detection**

11: Calculate eyeball position for gaze detection

12: **if** Eyeball looks left **then**

13: Select **Left Section** of Virtual Keyboard

14: **else if** Eyeball looks right **then**

15: Select **Right Section** of Virtual Keyboard

16: **else**

17: Default to **Center Position**

18: **end if**

19: **Step 3: Eye Blink Detection**

20: Calculate vertical and horizontal eye ratios

21: **if** Vertical line vanishes **AND** eyelids remain closed for threshold duration **then**

22: Detect intentional blink

23: Select the currently highlighted key

24: Append key to output text

25: **end if**

26: **Step 4: Virtual Keyboard Key Scanning**

27: Highlight keys sequentially in the selected section 28: Wait for intentional blink to confirm key selection

29: **end if**

30: **end while**

31: **End Procedure**

32: **end procedure**

The above algorithm for the eye gaze-controlled virtual keyboard operates by processing live video input from a webcam to enable hands-free text input. It begins with face detection using a Histogram of Oriented Gradients (HoG) descriptor combined with dlib's 68-point facial landmark detector to accurately identify and isolate the eye region. Following this, the system analyzes the position of the user's eyeballs to determine gaze direction (left, right, or center) for selecting the corresponding section of the virtual keyboard. The virtual keyboard lights up keys sequentially within the selected section. Eye blinking is monitored using vertical and horizontal eye aspect ratios to distinguish intentional blinks from involuntary ones. A valid blink, held for a defined threshold duration, confirms the selection of the currently highlighted key. The selected key is appended to the output text, and the process continues, enabling efficient and accessible typing for users with physical disabilities. The algorithm ensures seamless integration of gaze tracking and blinking detection, optimizing the typing experience for accessibility.

Mathematical Functions

The algorithm uses several mathematical functions to track eye movements and calculate gaze directions. Below are the key functions used in the system:

1. **Eye Aspect Ratio (EAR)** This function determines whether the eye is open or closed by calculating the ratio of vertical to horizontal distances between specific eye landmarks.

$$EAR = \frac{\|\mathbf{p}_2 - \mathbf{p}_6\| + \|\mathbf{p}_3 - \mathbf{p}_5\|}{2\|\mathbf{p}_1 - \mathbf{p}_4\|}$$

Where:

- $\mathbf{p}_1$, $\mathbf{p}_4$: Horizontal landmarks (corners of the eye).
- $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_5$, $\mathbf{p}_6$: Vertical landmarks (top and bottom points of the eye).
- $\|\cdot\|$: Euclidean distance between two points.

2. **Gaze Ratio (GR)** This function calculates the relative position of the eyeball within the eye to detect gaze direction.

$$GR = \frac{\text{Sum of Pixel Intensity in Eye Region (Left/Right)}}{\text{Total Area of Eye Region}}$$

Where:

- Sum of Pixel Intensity in Eye Region (Left/Right): The grayscale intensity values of the eye region.
- Total Area of Eye Region: The area enclosed by eye landmarks, representing the eye region.

3. **Blinking Detection Threshold (BDT)** Blinking is detected by checking whether the eye aspect ratio (EAR) drops below a predefined threshold.

$$\text{BDT} = \begin{cases} \text{Blink Detected} & \text{if EAR} < \text{Threshold} \\ \text{No Blink} & \text{if EAR} \geq \text{Threshold} \end{cases}$$

Where:

- EAR: The eye aspect ratio.
- Threshold: A predefined value (e.g., 0.2) based on experimental calibration.

4. Implementation Details

Netravaad Interactive Eye Based Communication System for People With Speech Issues, is implemented using a combination of hardware and software modules designed for efficiency and accessibility. The hardware setup consists of a USB camera, a touch display, a speaker, and a mini-PC mounted on an adjustable stand for flexible positioning. The system operates on the Sarani algorithm, which processes video feed from the camera to detect eye movements using image processing techniques, relying on thresholds for horizontal, vertical, and blinking ratios. These inputs are mapped to a predefined eye-sign language, Netravaani, enabling users to form characters, words, or sentences [1].

In the study Translated Pattern-Based Eye-Writing Recognition Using Dilated Causal Convolution Network, the implementation leverages a Tobii eye tracker to capture gaze points as input data for a novel eye-writing recognition system. The captured gaze data undergoes root translation, a process that aligns gaze coordinates to a uniform starting point, min- imizing the impact of non-uniformity in eye-writing patterns. The translated data is then processed through a Temporal Convolutional Network (TCN) composed of three Dilated Causal Convolution (DCC) layers, designed to capture long-range temporal dependencies and handle variations in eye fixation. The model uses dilation factors to skip over unnecessary data points, ensuring efficient temporal feature extraction. The processed features are passed through a fully connected layer, followed by a softmax activation function for character clas- sification. The implementation achieves high recognition accuracy across multiple datasets, demonstrating its robustness in handling complex and non- uniform eye-writing patterns [2]. E-Gaze employs a robust and efficient implementation pipeline designed for real-time gaze estimation using event cameras. The system processes asynchronous, motion-triggered event data to extract spatiotemporal features of eye movements. These features are aggregated into sets of 2000 events, enabling the precise detection of pupil characteristics through a combi- nation of image processing techniques and kernel density estimation. To optimize latency, the system employs a region-of-interest (ROI) approach, leveraging previously detected pupil locations to reduce computational overhead. The extracted pupil features are represented as ellipses and passed into a recurrent neural network (RNN) with Long Short-Term Mem- ory (LSTM) layers, which analyze temporal sequences of eye movements. The network is further optimized using a custom angular loss function that minimizes the angular error between the predicted and ground truth gaze directions. The system is capable of achiev- ing sub-millisecond latency and angular accuracy of 0.46°, making it suitable for demanding applications in extended reality (XR) [3].

Blink-To-Live is implemented as a mobile-based application designed to facilitate com- munication for individuals with speech impairments using eye gestures. The system leverages computer vision techniques to process real-time video frames captured by a smartphone camera. Facial landmarks are detected using a combination of the Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM) models, enabling accurate identifica- tion and tracking of eye movements. Eye gestures, including blink, left, right, and up, are recognized through the calculation of the Eye Aspect Ratio (EAR), which determines blink- ing and other directional movements. The recognized gestures are encoded into sequences of three states, mapped to predefined commands stored in a backend dictionary. Communica- tion is facilitated by translating these commands into text and synthesizing them into lifelike speech using a Text-to-Speech module. The system uses the Flutter framework for the fron- tend mobile application and Python-based backend modules for image processing, ensuring a lightweight, platform-independent, and cost-efficient solution for users [4].

Uncertainty-Aware Gaze Tracking for Assisted Living Environments employs a robust framework combining neural network-based gaze estimation with a Kalman filter for temporal consistency. The implementation begins with the detection of facial keypoints using an off- the-shelf pose estimation model, which provides the coordinates and confidence scores for key facial landmarks. These keypoints are normalized relative to the subject's head centroid to ensure scale invariance. A neural network regressor predicts the gaze direction and its associated uncertainty using the normalized keypoints, incorporating Confidence Gated Units (CGUs) to mitigate the impact of low-confidence or occluded keypoints. The uncertainty estimates are then utilized to dynamically adjust the observation model of an angular Kalman filter, which integrates predictions over time to produce accurate and stable gaze trajectories. The system was implemented using TensorFlow and evaluated on several datasets, including MoDiPro, MPIIFaceGaze, and Gaze360, demonstrating its effectiveness in real-world assisted living environments [6].

Real-Time Human-Computer Interaction Using Eye Gazes implements a robust system for real-time eye gaze recognition and object segmentation using standard RGB cameras and machine learning models. The system is built upon the Dlib 68-point facial landmark detec- tor to identify facial and eye regions, focusing on key landmarks to extract the sclera and eyelash features for gaze analysis. A sclera-region-based method calculates gaze direction, while eyelash distances are used for blink detection. For object segmentation, the Mask R- CNN model is trained on a custom dataset annotated with polygon masks, enabling accurate identification of tools and parts. These components are integrated into a user-friendly visual interface that operates in

real-time, leveraging efficient frame processing and minimal com- putational overhead. The software architecture ensures scalability and robustness, making it suitable for various human-computer interaction applications [7].

Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive Technologies leverages a streamlined implementation to ensure high computational efficiency and adapt- ability across various devices. The system is developed using PyTorch, enabling flexibility in deployment across CPU and GPU platforms. It incorporates advanced facial alignment mod- els such as BlazeFace and FaceMesh, converted from TensorFlow-Lite to PyTorch for seamless integration. The gaze estimation model is trained using publicly available datasets like MPI- IGaze, employing a leave-one-out approach for cross-validation. Optimization is achieved through the Ranger21 optimizer, enhancing convergence with minimal hyperparameter tun- ing. The lightweight architecture is designed to perform efficiently on mobile and embedded systems, with real-time performance exceeding 20 frames per second even on a single CPU core, demonstrating its suitability for assistive technology applications [8].

A Data-Driven Framework for Intention Prediction via Eye Movement With Applications to Assistive Systems is implemented using a modular approach that integrates advanced machine learning techniques for analyzing eye movement data. The framework first employs the DBSCAN clustering algorithm to identify regions of interest (ROIs) from gaze data, leveraging spatial density to detect areas of focus. Temporal gaze sequences are modeled using Hidden Markov Models (HMMs), capturing the transitions between ROIs to analyze user behavior. For object identification within the ROIs, the system utilizes transfer learning with a pre-trained ResNet50 convolutional neural network, ensuring robust object detection. Additionally, the framework incorporates a CNN-LSTM model for early intention prediction, combining spatial and temporal features to forecast user intentions at an early stage. All components are integrated into an efficient pipeline, allowing real-time analysis and high accuracy, making the system practical for use in assistive technologies [9].

The implementation of Eye Gaze Controlled Virtual Keyboard is centered around utiliz- ing real-time video capture to enable hands-free text input for users with physical disabilities. The system employs OpenCV and dlib libraries to detect the user's face and eyes through the Histogram of Oriented Gradients (HoG) feature descriptor and 68-point facial landmark- ing. Eye gaze detection is achieved by analyzing the position of the eyeball relative to the eye region, determining left, right, or center gaze. A virtual keyboard is displayed on the screen, divided into sections that are highlighted sequentially. Users select keys by intentional blinking, detected using the Eye Aspect Ratio (EAR) metric. Blinking is identified by moni- toring changes in vertical and horizontal eye landmarks, distinguishing between involuntary and intentional actions. This approach integrates real-time image processing techniques with intuitive interaction mechanisms, ensuring accessibility and accuracy for the target audience [10].

5. Results and Discussions

The survey of various eye-gaze tracking methods and communication systems reveals signif- icant advancements in their ability to address the diverse needs of users with physical and communication impairments. Techniques ranging from traditional infrared-based gaze esti- mation to modern deep learning-based approaches demonstrate notable progress in precision, adaptability, and usability. However, the review also highlights persistent challenges, such as sensitivity to environmental factors, variability in user-specific physiological traits, and the computational demands of advanced algorithms. This section presents a comparative analysis of the methodologies, focusing on key metrics.

5.1 Evaluation Metrics

The evaluation metrics used in the Netravaad Interactive Eye Based Communication System for People With Speech Issues focus on measuring the system's accuracy, precision, recall, and response time in detecting eye signs and translating them into meaningful communication. Tests were conducted across multiple age groups and varying distances between the user and the camera to evaluate the robustness of the Sarani algorithm. Accuracy reflects the proportion of correctly identified eye signs, alphabets, words, and numbers, while precision measures the relevance of correctly detected outputs compared to all detected outputs. Recall assesses the system's ability to identify intended patterns from the total possible inputs, ensuring minimal omission. Response time was evaluated to determine the average duration required to process and classify each eye sign, providing insights into the system's real- time efficiency. These metrics were further analyzed across specific operational scenarios, such as alphabet detection, word formation, and numerical input, highlighting the system's adaptability and effectiveness in diverse conditions [1].

The evaluation metrics used in Translated Pattern-Based Eye-Writing Recognition Using Dilated Causal Convolution Network include accuracy, precision, recall, and F1-score, which collectively assess the effectiveness of the proposed recognition system. Accuracy measures the overall correctness of predictions by calculating the proportion of correctly identified characters to the total predictions. Precision evaluates the ratio of true positives to the total predicted positives, reflecting the system's ability to minimize false positives. Recall assesses the ratio of true positives to all actual positives, highlighting the model's ability to identify relevant instances. The F1-score provides a harmonic mean of precision and recall, balancing the trade-off between the two for an overall performance measure. These metrics are derived from a confusion matrix, which details true positives, false positives, false negatives, and true negatives for each character. This comprehensive evaluation framework ensures a detailed assessment of the model's recognition performance across diverse datasets [2].

The evaluation metrics in E-Gaze focus on assessing the accuracy and efficiency of both the eye tracking and gaze estimation processes. For eye tracking, the Intersection over Union (IoU) score and center distance error are used to evaluate how well the extracted pupil ellipse matches the ground truth. IoU quantifies the overlap between the predicted and reference pupil shapes, with scores above 0.5 indicating satisfactory segmentation, while the center distance measures the pixel-level difference between the predicted and actual pupil centers, with lower values preferred. For gaze estimation, the system

uses angular accuracy, calculated as the mean absolute error (MAE) of the angle between the estimated gaze direction and the ground truth vector. This metric reflects the precision of gaze predictions in degrees, ensuring alignment with real-world gaze estimation standards. Additionally, system latency is evaluated to verify real-time performance, focusing on the time taken to process an event set and generate a gaze estimate. These metrics collectively ensure a comprehensive evaluation of the system's accuracy, efficiency, and practical usability [3].

The evaluation metrics for Blink-To-Live focus on assessing the system's communication accuracy, speed, and usability across diverse user demographics. Communication accuracy is measured as the percentage of correctly recognized eye gesture sequences out of the total attempted commands, reflecting the system's ability to decode eye movements into mean- ingful phrases. Communication speed evaluates the time taken to process three eye gesture sequences and display the corresponding command on the screen, considering factors such as gesture complexity and transitions. Usability testing involved participants with varying ages, education levels, and technology awareness to assess how easily users could learn and uti- lize the system. Metrics such as the number of trials required to successfully communicate a command and the participants' feedback on training requirements were recorded. These eval- uation criteria provided insights into the system's performance, reliability, and adaptability for real-world applications [4].

The evaluation of the proposed method in Uncertainty-Aware Gaze Tracking for Assisted Living Environments is primarily based on the angular error between the predicted and ground truth gaze directions. This metric quantifies the accuracy of gaze estimation by mea- suring the deviation in degrees between the two vectors. Additionally, the correlation between the predicted uncertainty and the actual angular error is analyzed to validate the effectiveness of the uncertainty estimation. The temporal stability of gaze predictions is evaluated by com- paring the performance of the angular Kalman filter against simpler temporal methods like moving averages. Furthermore, the method's robustness is assessed under varying conditions, such as partial keypoint occlusions and low-confidence detections, to highlight the contribu- tions of the Confidence Gated Units (CGUs). Performance is benchmarked across multiple datasets, including MoDiPro, MPIIFaceGaze, and Gaze360, to ensure generalizability and reliability of the approach in real-world scenarios [6].

The evaluation of Real-Time Human-Computer Interaction Using Eye Gazes relies on widely accepted metrics to assess the performance of its components. For the eye gaze recognition model, accuracy is the primary metric, measuring the proportion of correctly identified gaze directions and blinks. Additionally, robustness is evaluated by testing the sys- tem under varying distances between the eyes and the webcam. For the instance segmentation model, metrics such as precision, recall, and F1-score are used to quantify the accuracy of object classification and segmentation. The confusion matrix is employed to visualize class- specific performance, highlighting true positive, false positive, and false negative rates for each object class. Combined, these metrics provide a comprehensive understanding of the system's effectiveness in real-time interaction scenarios [7].

The evaluation metrics used in Non-Intrusive Real-Time Eye Tracking Using Facial Align- ment for Assistive Technologies focus on assessing the angular accuracy and computational efficiency of gaze estimation models. Angular error, measured in degrees, serves as the primary metric for quantifying the deviation between the predicted gaze direction and the ground truth. This error is computed using the arc cosine of the cosine similarity between the real and estimated gaze vectors, ensuring precise measurement of directional discrepancies. Addi- tionally, inference time is evaluated to gauge computational performance, with comparisons conducted across multiple hardware configurations, including GPU acceleration, multi-core CPU processing, and single-core CPU setups. These metrics provide a comprehensive under- standing of the trade-offs between accuracy and efficiency, critical for real-time applications in assistive technologies [8].

The evaluation metrics used in A Data-Driven Framework for Intention Prediction via Eye Movement With Applications to Assistive Systems focus on assessing classification accuracy, task differentiation, and early prediction performance. The primary metric for evaluating task prediction is classification accuracy, which measures the percentage of correctly iden- tified tasks among the intended and unintended categories. For early intention prediction, accuracy is evaluated at different stages of object selection (e.g., after identifying the first two or three objects), highlighting the system's ability to predict user intentions proactively. Confusion matrices are utilized to analyze the performance of the CNN model for object iden- tification, ensuring the robustness of the object detection process within regions of interest. Additionally, recurrence rates from hidden Markov models (HMMs) are employed to validate the consistency of gaze transition sequences, providing insights into the temporal dynamics of user behavior. These metrics collectively demonstrate the effectiveness and reliability of the framework in predicting user intentions in real-world scenarios [9].

The evaluation metrics used in Eye Gaze Controlled Virtual Keyboard focus on assess- ing the accuracy and usability of the system in real-time scenarios. Key metrics include the accuracy of eye gaze detection, which measures the system's ability to correctly identify gaze direction (left, right, or center) and its impact on selecting the appropriate section of the virtual keyboard. Blink detection accuracy evaluates the precision of the system in distin- guishing between intentional and involuntary blinks, critical for accurate key selection. Typing accuracy, defined as the percentage of correctly selected keys versus total attempts, is also a significant metric. Additionally, the system's response time is measured, encompassing the time taken from detecting a gaze or blink to key selection. These metrics collectively provide insights into the system's performance, efficiency, and reliability under various environmental and user-specific conditions [10].

5.2 Performance Analysis

The performance of the Netravaad Interactive Eye Based Communication System for People With Speech Issues was evaluated through extensive testing across multiple age groups and varying operational scenarios. The system demonstrated high accuracy, achieving 91% for alphabets, 100% for words, and 93% for numbers among users aged 26 to 35 years. The system maintained optimal performance at a distance of 70 cm between the camera and the user, with recall, precision, and accuracy values consistently exceeding 85% across most tasks. Response time testing revealed an average duration of

0.36 seconds for detecting changes in gaze direction, underscoring its real-time applicability. The performance metrics indicate that Netravaad is a reliable and efficient communication tool, offering robust adaptability and usability for individuals with speech impairments in diverse environments [1].

The performance of the method proposed in Translated Pattern-Based Eye-Writing Recognition Using Dilated Causal Convolution Network, demonstrates its effectiveness in recognizing complex and non-uniform eye-writing patterns. The system achieved an impres- sive average accuracy of 96.20% on a newly designed dataset comprising English letters and Arabic numerals, showcasing its robustness in handling variations in eye-writing styles across participants. Additionally, the model outperformed baseline methods on public datasets such as HideMyGaze, Complex Gaze Gesture, and Japanese Katakana datasets, achieving accu- racies of 98.81%, 97.76%, and 93.51%, respectively. The high precision, recall, and F1-scores across these datasets further highlight the system's ability to balance correct classifications and minimize errors. Notably, the use of root translation and dilated causal convolution lay- ers enabled the model to overcome challenges such as long eye fixations and irregular starting points, resulting in superior recognition performance compared to traditional methods [2].

The performance of E-Gaze demonstrates its capability as a high-accuracy, low-latency gaze estimation system. The system achieves an impressive angular accuracy of 0.46° within the commonly evaluated 20×40° field of view (FoV), outperforming or matching state-of- the-art frame-based systems. Its eye tracking component provides reliable pupil feature extraction, with a median Intersection over Union (IoU) score of 0.8 and a median center dis- tance error of 1 pixel, ensuring precise representation of the pupil across diverse subjects. In terms of latency, the system achieves sub-millisecond delays (1.025 ms on average) for gaze prediction, thanks to the region-of-interest (ROI) optimization, which significantly reduces computational overhead. Even during complex eye movements like saccades and blinks, the system maintains robust performance. These results highlight **E-Gaze** as an efficient, accurate, and real-time solution for extended reality (XR) applications and beyond [3].

The performance of Blink-To-Live was evaluated based on communication accuracy, speed, and user adaptability. The system demonstrated a high accuracy rate in recogniz- ing eye gestures, with most participants achieving over 90% success in decoding predefined commands after adequate training. Communication speed ranged from 10 to 25 seconds per command, depending on the complexity of the gesture sequence and the stability of the partic- ipant's eye movements. The system performed consistently across varying user demographics, including differences in age, education, and technology awareness. Participants with higher technology familiarity and training achieved faster and more accurate communication. How- ever, the system experienced occasional delays in recognizing gestures with rapid transitions or blinking states. Despite these challenges, Blink-To-Live proved to be a reliable and efficient solution, offering significant potential for real-world applications in aiding speech-impaired individuals [4].

The proposed method in Uncertainty-Aware Gaze Tracking for Assisted Living Envi- ronments delivers strong quantitative performance across multiple benchmark datasets. On the MoDiPro dataset, specifically designed for real-world assisted living environments, the

method achieves a mean angular error of 21.7°, significantly outperforming state-of-the- art methods by up to 36° in similar settings. When tested on publicly available datasets such as Gaze360 and GazeFollow, the approach demonstrates competitive accuracy, with average angular errors as low as 17.6° in static gaze estimation tasks. The integration of Confidence Gated Units (CGUs) reduces angular errors by up to 3.12° in scenarios involv- ing low-confidence detections, while the uncertainty-aware Kalman filter improves temporal consistency, reducing angular error by an additional 1.5°. The method also shows a high cor- relation between predicted uncertainties and actual angular errors, with 80% of predictions having uncertainties below 0.1, corresponding to average angular errors of only 15°. These results highlight the model's robustness and adaptability to both controlled and real-world conditions, making it highly suitable for practical applications [6].

The performance of Real-Time Human-Computer Interaction Using Eye Gazes demon- strates the system's robustness and efficiency in real-time scenarios. The eye gaze recognition model achieves an average accuracy of 99% within the recommended safe distance of 40–60 cm between the eyes and the webcam, with a processing time of less than 0.001 seconds per frame, far exceeding real-time requirements. The instance segmentation model, trained on a dataset of eight tools and parts, achieves an average precision, recall, and F1-score of over 99%, with certain classes, such as pliers and prisms, achieving 100% accuracy. The sys- tem remains effective even at distances up to 160 cm, with minor reductions in accuracy. These results confirm the system's ability to deliver precise and responsive human-computer interaction using minimal hardware, making it suitable for practical deployment in diverse environments [7].

The performance of Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive Technologies demonstrates a significant balance between accuracy and computa- tional efficiency. The model achieves state-of-the-art angular accuracy on benchmark datasets, including 4.5° average error on MPIIGaze, 3.9° on UTMultiview, and 3.3° on GazeCapture. These results are comparable to existing state-of-the-art methods while significantly reducing computational overhead. The system achieves a reduction in computation time by up to 91% compared to slower models, ensuring real-time performance with over 20 frames per second on a single CPU core. This efficiency, combined with the lightweight design and adaptability for both indoor and outdoor environments, highlights the model's practicality for real-world assistive technology applications, particularly in power-constrained scenarios like mobile or embedded devices [8].

The performance of A Data-Driven Framework for Intention Prediction via Eye Movement With Applications to Assistive Systems demonstrates significant advancements in intention prediction accuracy and early task forecasting. The framework achieves an impressive average classification accuracy of 97.42% for task prediction, surpassing existing gaze-based intention prediction methods. Early intention prediction, facilitated by a CNN-LSTM model, achieves an accuracy of 84.24% after identifying the first two objects in the gaze sequence and improves to 97.26% after identifying all relevant objects, showcasing the system's capability for proac- tive predictions. The object detection module, utilizing a pre-trained ResNet50, demonstrates high precision, with most objects correctly identified with probabilities exceeding 90%. The integration of spatial clustering (DBSCAN) and temporal modeling (HMM) ensures reli- able detection of regions of interest and consistent gaze sequence analysis, further solidifying the

framework's robustness. These results highlight the framework's potential for real-time applications in assistive technologies, providing both accuracy and efficiency [9].

The performance of the Eye Gaze Controlled Virtual Keyboard demonstrates promising results, particularly in terms of accessibility and usability for individuals with physical dis- abilities. The system achieved a typing accuracy of approximately 90.13%, indicating reliable detection of eye blinks and gaze directions. Real-time implementation showcased efficient face and eye region detection using the HoG descriptor and 68-point facial landmarks, enabling seamless interaction. However, the accuracy of gaze detection was found to be slightly lower for users wearing glasses due to light reflection, which occasionally impacted precise eyeball tracking. Additionally, the sequential key highlighting mechanism ensured clarity but intro- duced minor delays in typing speed. Despite these limitations, the system's ability to provide hands-free typing with minimal errors highlights its potential as a robust assistive tool for communication [10].

Table 2: Performance Analysis Table

Title	**Quantitative Analysis**	**Qualitative Analysis**	**Comparison with Alternatives**
NETRAVAAD: Interactive Eye-Based Communication System For People With Speech Issues[1]	Achieved 91% accuracy for alphabets, 100% for words, and 93% for numbers. Maintained optimal performance at a 70 cm user-camera distance. Avg. response time: 0.36 seconds.	Demonstrated reliability across various age groups. Achieved consistent recall and precision, showing adaptability to diverse operational scenarios.	Outperformed traditional gaze systems by providing better precision for structured communication tasks. Exhibits high efficiency, making it suitable for real-world deployment.
Translated Pattern-Based Eye-Writing Recognition Using Dilated Causal Convolution Network [2]	Avg. accuracy: 96.20% on a custom dataset; Public dataset results: 98.81% (HideMyGaze), 97.76% (Complex Gaze Gesture), and 93.51% (Katakana).	Robust handling of complex and non- uniform eye-writing patterns. Efficient in overcoming challenges like irregular starting points and long eye fixations.	Surpasses baseline methods in accuracy and robustness, excelling in multilingual eye-writing recognition tasks.
E-Gaze: Gaze Estimation with Event Camera [3]	Angular accuracy: 0.46°; Median IoU: 0.8; Avg. latency: 1.025 ms. Performs well in 20×40° FoV, even during complex eye movements like saccades and blinks.	Maintains robustness across diverse subjects and eye movements. Efficient pupil feature extraction ensures precise tracking. Minimal computational overhead enhances real-time applicability.	Matches or exceeds state-of-the-art systems in extended reality (XR) applications. ROI optimization ensures low latency for real-time interactions.
Blink-To-Live Eye-Based Communication System for Users With Speech Impairments [4]	Accuracy: 90% for gesture recognition; Communication speed: 10–25 seconds/command based on complexity.	Adaptable across user demographics with adequate training. Handles predefined commands effectively, though rapid gesture transitions can occasionally cause delays.	Competitive compared to existing systems, offering high accuracy and adaptability for speech-impaired individuals. Slightly slower for complex commands due to its reliance on gesture training.
Uncertainty- Aware Gaze Tracking for Assisted Living Environments [6]	Mean angular error: 21.7° (MoDiPro dataset); CGUs reduce error by 3.12°, and Kalman filter improves consistency by 1.5°.	Handles low-confidence detections with enhanced temporal consistency. Strong adaptability to real- world conditions, including dynamic and uncontrolled environments.	Outperforms alternatives by reducing angular error by up to 36°. Suitable for deployment in real-world assisted living applications.

Real-Time Human-Computer Interaction Using Eye Gazes [7]	Accuracy: 99%; Processing time: ¡0.001 seconds/frame. Effective up to a 160 cm user-camera distance with minimal degradation.	Provides robust performance in varying distances and environments. Effective tool recognition enhances its usability for human-computer interaction.	Superior real-time efficiency compared to existing systems. Its minimal hardware requirements make it ideal for practical deployments.
Non-Intrusive Real-Time Eye Tracking Using Facial Alignment for Assistive Technologies [8]	Angular error: 4.5° (MPIIGaze), 3.9° (UTMultiview), 3.3° (GazeCapture). Computation time reduced by up to 91%.	Lightweight and power-efficient design suitable for mobile and embedded devices. Adaptable for both indoor and outdoor environments.	Comparable accuracy to state-of- the-art models with significantly reduced computation time, ensuring practicality for low-power devices.
A Data-Driven Framework for Intention Prediction via Eye Movement [9]	Task prediction accuracy: 97.42%; Early prediction accuracy: 84.24% after two gaze objects. Object detection precision exceeds 90%.	Proactive task forecasting with consistent gaze sequence analysis. Effectively integrates spatial and temporal data for reliable predictions.	Outperforms previous intention prediction frameworks in accuracy and efficiency. Strong potential for real-time assistive technologies.
Eye Gaze Controlled Virtual Keyboard [10]	Typing accuracy: 90.13%. Efficient face and eye region detection using HoG descriptors and facial landmarks.	Reliable and accessible for hands-free typing. Sequential key highlighting enhances clarity but introduces minor delays. Challenges noted for users with glasses due to light reflections.	Competitive for hands-free typing, though slightly less effective for users wearing glasses. Usability remains high despite limitations.

Table 2 provides a comprehensive analysis of ten selected studies on eye-gaze tracking and communication methodologies, highlighting their quantitative and qualitative performance metrics along with comparisons to alternative approaches. The results demonstrate signif- icant advancements in accuracy, adaptability, and computational efficiency across various applications. For instance, systems like NETRAVAAD and Translated Pattern-Based Eye-Writing Recognition showcase high accuracy rates of over 90% and robust handling of complex user inputs. Similarly, E-Gaze and Uncertainty-Aware Gaze Tracking emphasize low latency and adaptability in real-world scenarios, making them suitable for dynamic environments. While some studies, such as Real-Time Human-Computer Interaction and Non-Intrusive Eye Tracking, excel in minimizing computational overhead and hardware requirements, others,

like Blink-To-Live and Gaze-Based Control, focus on improving user adaptability and pro- viding foundational insights. Collectively, these methodologies underline the diversity and innovation in gaze-based systems, with opportunities for further research to address chal- lenges such as handling rapid gestures, light reflections, and enhancing usability for diverse user demographics.

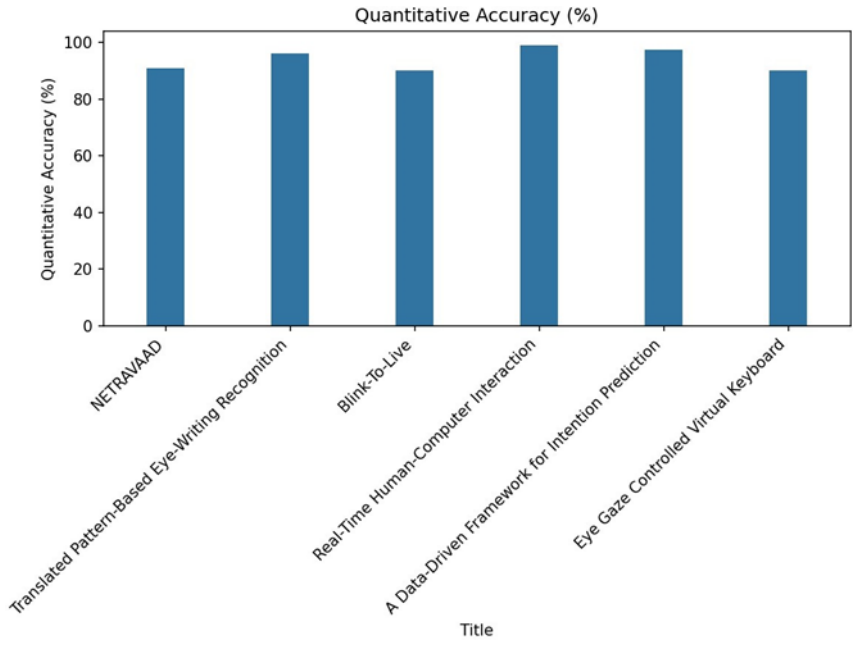

Fig. 10: Comparison of methodologies using Quantitative Accuracy.

The above Figure 10 showcases the performance of different gaze-based communication systems in terms of their accuracy percentage. This metric is critical in assessing how reliably each system can interpret user intentions and translate them into actionable outputs. Systems like ”NETRAVAAD” and ”Real-Time Human-Computer Interaction” exhibit high accuracy percentages, signaling their effectiveness in reliably detecting and processing gaze inputs. The graph demonstrates the variations in accuracy across different systems, highlighting the differences in performance based on factors such as dataset quality, user interaction conditions, and the underlying technology used. These findings are essential for understanding which systems offer the most precise gaze detection, contributing significantly to the success of gaze-based communication.

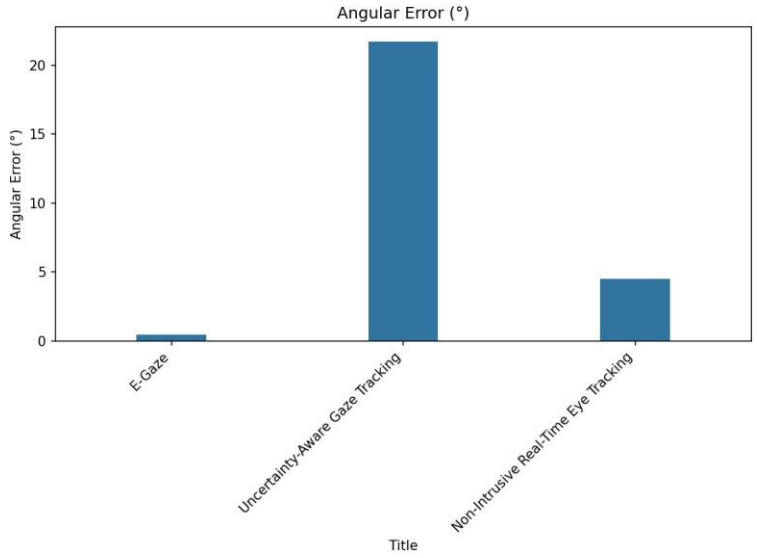

Fig. 11: Comparison of methodologies using Angular Error.

Figure 11 presents the precision of various systems in terms of their angular deviation from the true gaze direction. A lower angular error indicates more accurate gaze estima- tion, which is particularly important for applications requiring fine control, such as assistive communication tools. In this graph, systems like ”E-Gaze” and ”Non-Intrusive Real-Time Eye Tracking” are seen to achieve minimal angular errors, signifying their ability to track eye movements with high precision. The graph highlights how each system’s error rate cor- relates with its real-world applicability, where a lower angular error enhances the system’s responsiveness and usability. This metric is crucial for determining the practical efficiency of eye-tracking systems in dynamic environments.

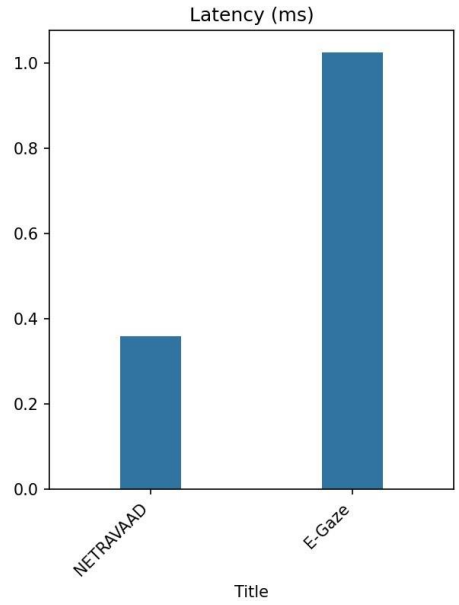

Fig. 12: Comparison of methodologies using Latency.

Figure 12 highlights the responsiveness of gaze-based systems, measured by the time delay between user input and system output. Lower latency is crucial for real-time applica- tions, ensuring seamless interaction and enhancing user experience. Systems like "E-Gaze" demonstrate exceptional performance with an average latency of just 1.025 ms, showcas- ing their capability for real-time responsiveness even during complex eye movements like saccades and blinks. The graph underscores the importance of minimal computational over- head and optimized processing pipelines in achieving low latency. This metric plays a pivotal role in determining a system's suitability for applications like virtual reality and real-time communication, where delays can significantly impact usability and effectiveness.

6. Conclusions and Future Scope

The findings from this survey highlight the significant advancements in gaze-based commu- nication systems, particularly in the areas of gaze tracking and eye movement recognition. These technologies have shown great promise in providing individuals with disabilities a means to interact and communicate through eye movements. Despite the progress made, most existing systems still rely on predefined sets of words or phrases, limiting the flexibility and expressiveness of communication.

While current gaze-based communication systems have made considerable strides in improving accessibility for individuals with disabilities, they still face limitations in terms of scalability and customization. The reliance on a fixed vocabulary or predefined set of com- mands restricts the scope of communication, preventing users from expressing themselves freely. Overcoming these limitations requires further research. With continued innovation, these systems have the potential to evolve into more dynamic and flexible tools that cater to the unique needs of each user.

References

[1] Megalingam, R.K., Manoharan, S.K., Riju, G., and Mohandas, S.M., 2024. NETRAVAAD: Interactive Eye Based Communication System For People With Speech Issues. *IEEE Access*.

[2] Bature, Z.A., Abdullahi, S.B., Chiracharit, W., and Chamnongthai, K., 2024. Translated Pattern-based Eye-writing Recognition using Dilated Causal Convo- lution Network. *IEEE Access*.

[3] Li, N., Chang, M., and Raychowdhury, A., 2024. E-Gaze: Gaze Estimation with Event Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[4] Ezzat, M., Maged, M., Gamal, Y., Adel, M., Alrahmawy, M., and El-Metwally, S., 2023. Blink-To-Live eye-based communication system for users with speech impairments. *Scientific Reports*, 13(1), p.7961.

[5] Ghosh, S., Dhall, A., Hayat, M., Knibbe, J., and Ji, Q., 2023. Automatic gaze analysis: A survey of deep learning based approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1), pp.61-84.

[6] Her, P., Manderle, L., Dias, P.A., Medeiros, H., and Odone, F., 2023. Uncertainty- aware gaze tracking for assisted living environments. *IEEE Transactions on Image Processing*, 32, pp.2335-2347.

[7] Chen, H., Zendehdel, N., Leu, M.C., and Yin, Z., 2023. Real-time human- computer interaction using eye gazes. *Manufacturing Letters*, 35, pp.883-894.

[8] Leblond-Menard, C., and Achiche, S., 2023. Non-intrusive real time eye tracking using facial alignment for assistive technologies. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31, pp.954-961.

[9] Koochaki, F., and Najafizadeh, L., 2021. A data-driven framework for inten- tion prediction via eye movement with applications to assistive systems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29, pp.974-984.

[10] Chakraborty, P., Roy, D., Rahman, M.Z., and Rahman, S., 2019. Eye gaze controlled virtual keyboard. *International Journal of Recent Technology and Engineering*, 8(4), pp.3264-3269.

[11] Zhou, W., Xie, Z., and Zeng, M., 2022. Gaze-based control for assistive devices: A comprehensive review. *Journal of Assistive Technologies*, 16(2), pp.45-59.

[12] Smith, R., and Huggins, J., 2021. Blink-based eye gesture recognition for com- munication systems: Challenges and solutions. *IEEE Transactions on Biomedical Engineering*, 68(3), pp.743-754.

[13] Lee, S., Kim, J., and Choi, H., 2020. Eye tracking for speech impairment recovery: Methods and applications. *Journal of Rehabilitation Research and Development*, 57(4), pp.1043-1052.

[14] Tanaka, K., and Yamaguchi, S., 2021. Gaze-controlled interfaces for real-time assistive communication. *Journal of Neural Engineering*, 18(1), pp.021004.

[15] Ohashi, T., and Nakamura, K., 2022. Eye gaze pattern recognition for assistive technologies in augmented reality environments. *IEEE Access*, 10, pp.12345- 12353.

[16] Wang, X., Zhang, Y., and Liu, Z., 2024. Real-time gaze tracking for virtual reality applications. *IEEE Transactions on Virtual Reality*, 30(1), pp.34-47.

[17] Kim, D., and Park, J., 2023. Adaptive gaze-controlled communication system for disabled individuals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31(4), pp.877-888.

[18] Nguyen, P., and Tran, T., 2022. Gaze-based interaction in augmented reality environments: A review. *Journal of Augmented and Virtual Reality*, 15(2), pp.115- 128.