International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

An Intelligent Eye Gaze–Driven E-Book Reader using MediaPipe and OpenCV

Gunalini R[1], Muhammad Hamthan S[2] , Ms. K. Sangeetha[3]
[1],[2]UG - Department of Information Technology, B. S. Abdur Rahman Crescent Institute of Science and Technology, Vandalur, Chennai, Tamil Nadu, India.
[3]Assistant Professor, Information Technology, B. S. Abdur Rahman Crescent Institute of Science and Technology, Vandalur, Chennai, Tamil Nadu, India
Email ID: *gunaliniraj@gmail.com[1], muhammadhamthan660@gmail.com[2], sangeethak@crescent.education[3]*

Abstract
The fast switching to digital documents in educational and professional life has raised the number of screen-based reading enormously, but it is accompanied by a notable decrease in concentration, constant-distraction, and the inability to continue reading. To solve these problems, the following paper gives a gaze-assisted web-based reading platform that can track the eye gaze of users and tell them the line that must be read and also allow them to navigate through the pages of a book using only double and triple blast gestures. The suggested system combines a computer vision interface with the web application layer, communicating on a lightweight Representational State Transfer (REST) interface. A MediaPipe FaceLandmarker model is used to obtain facial landmarks in order to come up with normalized gaze vectors that are made by the displacement of the iris and the estimation of the head-pose which are filtered by an 8-frame median filter before being sent to the server. At the same time, Eye Aspect Ratio (EAR) -based analyzes of blink occurrences are designed to identify gesture intent by passing gesture intent validations based on inter-blink interval and a rhythm gate to avoid false positives. The threaded Flask backend synchronizes the processed gaze and gesture cues, and integrates with a Portable Document Format (PDF) rendering module to provide per-line highlight updates and page-turn feedback. The deployed system has shown gaze-based line tracking and high confidence hands-free navigation through blink gesture recognition at a timeframes (temporal) of 550 ms and thus the viability of an eye-controlled, accessible digital reading interface.

Keywords: *Gaze Tracking, Blink Gesture Recognition, Eye Aspect Ratio, Computer Vision (CV), Hands-Free Digital Reading Interface.*

1. Introduction
The advent of digital technology has had a profound impact on the reading habits which have seen the possibility of converting significant numbers of traditional print media as digital forms like the e-book, e-documents and online articles. Digital reading, as a form of information consumption, has grown in frequency especially in the learning community and professional societies where learning students and professional people are currently fully dependent on electronic information to access academic and technical resources. The availability, flexibility, and convenience of distribution of electronic slide papers have seen them serve as a convenient substitute of traditional writings. As a result, the quality of the effective and engaging digital reading experience turned out to be a significant aspect that should be considered. Notwithstanding these benefits, there are a number of challenges presented by digital reading, which may harm the engagement and understanding by users. Screen-based reading can be akin to physical books in terms of natural, immersion reading in that the former may not have the same effect, which can result in a lack of focused attention and lapses of continuity in reading. Also, online spaces often present users with distractors that include notifications, apps running in the backdrop and opportunities to do more, all of which may lead to loss of concentration. The second typical challenge that arises to the readers is not knowing their position in the reading process as they move out of the line or paragraph which breaks the flow of reading and necessitates some more

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

mental activity to get the reader back on track. The latest developments towards eye-tracking technologies have shown potential in aiding the solution of these problems as the technology can be used to track the visual attention of a user whilst reading. The methods that are currently available are, however, usually based on either specialized eye-tracking equipment or very accurate infrared sensors which makes them, again, unavailable to regular users. Moreover, a large portion of existing solutions offer gaze interaction (not completely hands-free: controlled) or merely offer gaze analysis to conduct offline analysis not as a real-time interaction with document content. In addition to this correct mapping of gaze coordinates to separate lines of text is technically difficult in systems using webcams because of less precision of gaze in comparison to the distance between the lines of text in normal reading material. This has led to a huge gap in research in designing a convenient, lightweight platform that could enable real-time reading support of gaze and manage natural hands-free navigation without having to buy specialized hardware. This paper is about a gaze-supported web reading site which will enhance reading continuity and engagement in online documents with a normal web camera. The system records the eye movements in real time and plots the positions of the gaze to record the content with data to aid readers when reading the screen. Besides identification of user via gazes line identification, the platform also allows the possibility of full hands-free interaction with the system with page navigation performed with the help of blinking gestures. The proposed system will support responsive reading assistance, based on the integration of the computer vision-based gaze estimation with the lightweight web architecture, without the use of specific eye-tracking hardware. The architecture is a mixture of facial landmark extraction, gaze estimation, blink gesture recognition and Web-based document interface in order to allow real-time interaction between the user and digital reading material. It has made two important contributions: (1) Hybrid Gaze Model - a weighted combination of head-pose (nose-tip, weight 0.65/0.75) and iris-offset (weight 0.35/0.25) signals, a weighted extension of the range of gaze based solely on iris-ratio. (2) Adaptive EAR Baseline - a baseline-calibrated online median estimator having close thresholds equal to baseline 0.65 and open thresholds equal to baseline 0.80. (3) Dual-Gesture Blink Navigation - a combined state-based providing next page single-blink (DOUBLE_GAP = 0.55 s) and previous-page triple-blink (TRIPLE_WINDOW = 1.0 s, TRIPLE_WAIT = 0.45 s) navigation, with GESTURE_COOLDOWN = 1.5 s. (4) Queue-Based Blink Relay - a thread-safe Flask endpoint, comprising a limited-capacity queue (maxsize=5) that lets the 30 fps CV loop and the browser polling loop decouple. ### Browser-Side Ordinary Least Squares (OLS) Calibration - a 9-point OLS viewport mapping that has been completely computed in JavaScript.. (6) Open-Source Full-Stack Application Python 3, openCV, MediaPipe, Flask, PyMuPDF and bare-bones HTML/JS; you don't need a Streaming-Graphics Processing Unit (GPU) or a proprietary Software Development Kit (SDK)..

2. Related Work

2.1.Eye-Tracker-Based Interactive eBook Readers

Sujaini et al. [10] developed a Python-based interactive eBook reader based on the Django framework, applying 68-point facial landmark model of dlib and OpenCV to retrieve eye pupil groups. Their system puts emphasis on the line that is being read and allows eye-Movement commands to navigate. A usability test on 30 participants (15 test, 15 control) indicated that the eye-tracker group had a higher average reading comprehension of 8.6 compared to the standard PDF reader group of 7.33 (although reading time took about 3 minutes 23 seconds more). Black-box testing under 8 lighting and hardware conditions revealed complete test-case accuracy under good-lighting conditions and lower performance (66.7 - 77.8%) in poor-lighting or where the users wore glasses. This is highly similar to problems in Readzy, which also experiences a similar decline in the MediaPipe landmark model when exposed to disuniform illumination.

2.2.Real-Time Gaze Metric Computation

The extension is EyeLiveMetrics [11], created by Hienert et al., which takes raw gaze coordinates output by a Tobii Pro eye tracker via a WebSocket connection, an I-VT (Velocity-Threshold Identification) fixation classifier is applied with the

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

threshold set to 30/s, and fixations are mapped in real-time to HTML word elements. The given plug-in calculates a set of measurements of fixations (all-fixation time, first-fixation time, average fixation time), saccade measurements (time, amplitude, maximum velocity) and reading measurements (first-pass time, regression-path time, rereading time). It was demonstrated that the mean per-coordinate processing time of the eyes was 0.464 ms, indicating eye trackers with the power to reach 1200 Hz. The Pearson correlations obtained with Tobii Pro Lab were 0.962-1.0 for fixation measures, and 0.979-1.0 for reading measures. By contrast, the poposed system is concerned with a gaze-to-line mapping at lightweight, not word-level metric calculation, and compromises metric richness with close-to-zero hardware cost with its webcam-based MediaPipe pipeline.

2.3.Reading Tracking with Jump Reading Support

The first reading tracking system and in-real-time highlighting system, presented by Yang et al. [12], specially targeted linear and jump reading. To fill the gap between gaze tracking accuracy (around 23 cm) and text line spacing (35 mm) RT2H uses punctuation marks as language anchors to jump-read relocation, and GPT-4o mini with prompt engineering to elect the most contextually plausible relocation candidate in cases where multiple anchors are located. Calibration drift is prevented by a dynamic calibration scheme in which the line-gaze set is aligned, which decreases average error on the Y-axis of 0.6609 cm without (0.3904 cm with) calibration over 120 s. In controlled experiments, there was 84.21% jump-reading relocation accuracy. An experimental 18-subject field test demonstrated RT2H as lowering task completion time (13.5% min) against a blank control (98.2 s vs 113.5 s). No specialized hardware eye tracker is needed to solve similar drift problems: Readzy 9-point OLS browser-side calibration acts similarly, and is auto-calibrated by rolling percentile bounds.

2.4.Iris-Based and Head-Pose Gaze Estimation

Zhang et al. [3] have revealed that normalized iris ratios give a device-independent gaze. The Face Landmarker of MediaPipe [4] is a prediction of iris keypoints that permit the localization of iris-centers. Nose-tip position-based head-pose estimation is resistant to iris occlusion, but confounds head rotation with changes in gaze [5]. The proposed system hybrid model combats limitations by weighting both head-pose and iris-offset 65-75 and 25-35 respectively, depending on axis. Gaze smoothing using Kalman filters [6] is used with process noise in form of Q = 0.02 and measurement noise in form of R = 0.015 in the horizontal axis as it is in accordance with the existing process of online estimation methods.

2.5.Blink Gesture Interfaces

A new single-blink scrolling was proposed by Soares et al. [7] to Amyotrophic Lateral Sclerosis (ALS) patients. EAR blink detector suggested by Soukupova and Czech [8] has already been established as the standard of reference of real time blink detection using face landmarks. This is extended by the proposed system that consists of a dual-gesture state machine: a double-blink during DOUBLE_GAP = 0.55 s emits next-page navigation and a triple-blink during TRIPLE-WINDOW = 1.0 s (with minimum inter-blink duration MIN_INTERVAL = 0.18 s and maximum inter-blink duration MAX_INTERACTION = 0.4 A GESTURE_COOLDOWN of 1.5 s avoids quick unintended re-init likes.

2.6.Webcam-Based Gaze Calibration Approaches

A core issue with webcam-based gaze estimation is calibration, where the transformation of raw signals of gaze to screen coordinates is idiosyncratic to users, and changes with session, posture and distance. Another system that uses implicit user interaction events (clicks and movements) as training data to train an online ridge-regression gaze model on the null hypothesis is WebGazer, which was proposed by Papoutsaki et al. [9] to eliminate explicit calibration processes. WebGazer showed gross showing a median gaze error of about 130 pixels on a 1920undertuned screen, which, although not sufficient to target words, is satisfactory in term of highlighting lines. The model maintains its regression model as it is updated on a per-user-action basis, allowing it to improve on a continuous in-session basis. Conversely, Cao et al. [2] showed a phone-based gaze tracking method on arbitrary surfaces by

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

combining iris displacement on the frontal camera with a trained appearance model where they achieved under centimetre-level accuracy, which did not require that individual users be trained. But this method needs a controlled physical configuration, which is not possible with normal desktop reading. The proposed system takes a middle ground: the explicit 9-point Ordinary Least Squares (OLS) browser-side calibration is the main one with an additional auto-calibration mechanism that is required to make adaptations as it is used in the form of rolling percentile bounds. The hybrid design will provide sufficient mapping even without a persistent background interaction model, which is what makes it more appropriate in the context of focused reading sessions during which the user focuses his or her attention on the text of the documents and not the interface itself.

2.7.Comparative Summary of Related Systems

The current literature analysis above shows that there are a number of common trade-offs that prompted the design decisions in Readzy. The six systems are summarised in Table A below based on five dimensions: Hardware dependency, gaze granularity, real-time capability, hands-free navigation support and open-source availability. Hardware-based systems that use eye-trackers alone, like EyeLiveMetrics [11] and RT2H [12], can be better (gaze accuracy is at word level or even better) but need expensive, proprietary sensors (Tobii Pro) that many users cannot afford. WebGazer [9] was the first system to use implicit browser-based calibration, though this was not intended as a structured text reading system and results in metric-scale errors, which do not give useful line hinting. Sujaini et al. [10] would most closely match Readzy in its interaction eBook focus and the use of webcams only, but with a dlib 68-point model instead of the newer MediaPipe 478-landmark mesh, and lacks hands-free blink-based navigation. Readzy is an unusual hybrid gaze model that is (a) lightweight, (b) can run on commodity webcams, (c) includes a dual-gesture blink state machine to provide fully hand-free navigation, (d) is browser-side OLS calibrated (no model training) and (e) fully open-source (uses Python and JavaScript). This combination places Readzy as an accessibly the most convenient solution within the literature sampled to readers who need to have continuity of gaze-assisted reading, but cannot afford the specialized hardware. The difference in per-word tracking performance between the system and hardware-based systems is actually an admitted shortcoming, and speculatively points to the next wave of research into either appearance-based deep gaze models implementation [3] or lightweight neural regression methods [13].

3. Methodology

3.1.System Architecture

The design is based on a three-tier pipeline with the following conceptual model: Webcam Eye Tracker (Python) Flask Rest Server Browser Reader. The eye tracker module is a module that grabs frames at a Universal Serial Bus (USB) web camera with a maximum resolution of 1280x720, applies landmarks, and sends gaze and blink data to the Flask server using HTTP POST. The browser reader scans these endpoints with a configurable frequency (35 ms interval) gazes, 80 ms interval blinks), and displays the relevant visual stimulus.

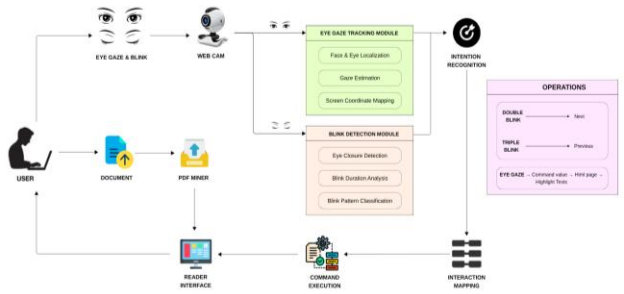

Figure 1 System Architecture of our proposed work Eye Gaze and Blink Detection Pipeline

Table 1 System architecture Layers

Layer	Technology	Role
Eye Tracker	Python, MediaPipe, OpenCV, scipy	Landmark detection, gaze/blink computation
Flask Server	Python, Flask, PyMuPDF	State relay, PDF parsing, file management
Web Frontend	HTML, CSS, Vanilla JS	Reading UI, gaze visualization, PDF display

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

3.2. Facial Landmark Detection

The MediaPipe FaceLandmarker model with a single face constraint and in VIDEO mode is used to extract landmarks. The model gives a mesh of 478 normalized landmarks (x, y, z in [0, 1]^3 with respect to the image frame). The proposed system are iris centers (468, 473 landmarks/eye), corners of eyes and limbs extremum on both sides of the eye, nose (landmark 1) and 6 EAR landmarks/eye according to a convention of Soukupova and Cech [4]. Frame-level the injection of milliseconds since the epoch can meet the MediaPipe VIDEO-mode monotonicity requirement and provides stable temporal registration.

3.3.Gaze Estimation

Estimation of raw gaze position involves a combination of the orientation of the head as well as relative iris movement. The normalized (x, y) location of the nose tip landmark is used to represent head orientation, and is a computationally cheap predictor of yaw and pitch that does not need a complete 3D head-pose estimate. Iris displacement is calculated as the distance between the iris center and medial canthus/ total intercanthal distance, calculated separately in horizontal and vertical axis and between the two eyes and averaged.The fused gaze estimate r_x (horizontal) is formed as:

$$r\_x = (1 - w\_H) \cdot head\_nx + w\_H \cdot iris\_nx$$

w H line: w H = 0.35 is experimentally adjusted iris weight. A similar expression applies to r y when w V = 0.25 that indicates more head contribution to vertical gaze. Prior to filtering both components are clipped to [0, 1].

3.4.Auto-Calibration and Kalman Filtering

The system uses an unmonitored auto-calibration approach over the initial ninety video frames (about three seconds at 30 fps). This window takes samples of the system (head_x, head_y, iris_h, and iris v). Calibration computes the 5th and 95 th percentile of both distributions, scales the range by some padding factor to allow out-of-sample positions, and then uses these values to linearly normalize the successive readings in the series to [0, 1]. The calibration then every 60 frames later to accommodate slower changes in posture. Individual gaze parts are each smoothed using a 1D Kalman filter, process noise Q = 0.02 and measurement noise R = 0.015 (horizontal gaze) and Q = 0.15, R = 0.12 (vertical gaze). The asymmetric parameters are indicative of reduced temporal variability of vertical gaze in reading. The filtrated outputs r x and r y are sent to the Flask server with a maximum transmission period of 30 ms.

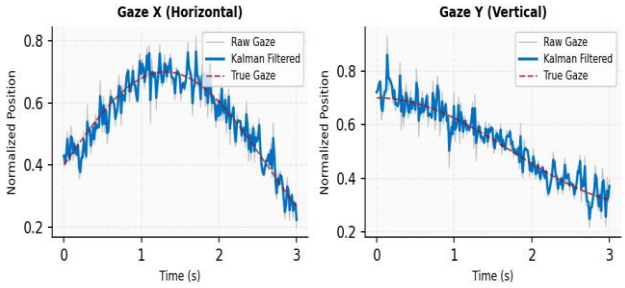

Figure 2 Kalman Filter Smoothing Effects on Raw Gaze Coordinates

3.5.Blink Detection and Gesture State Machine

The Eye Aspect Ratio calculated based on 6 coordinates of landmarks per eye is used to detect a blink. The adaptively set closure threshold 65% of a running median EAR medicine is estimated over the last 100 running open-eye frames. Only when the eye remains closed over MIN_CLOSE_FRAMES (=3 consecutive frames) and then it re-opens more than 80 per cent of the baseline, a blink event will be confirmed to avoid the noise and micro-squints of false detection. Gesture recognition is implemented in the form of three state machine. After the initial confirmed blink, the machine goes to a waiting_second state and begins a DOUBLE_GAP = 0.55 s countdown. When the second blink is received in the window and the time between them is within the valid range of rhythms [MIN_INTERVAL, MAX_INTERVAL] = [0.18, 0.40] s, the machine jumps to waiting_third. A motion timer triggers a look-twice gesture once TRIPLE_WAIT = 0.45 s except likely should there enter a third blink, still in the range of rhythms, cancelling the motion timer in preference of a look-triple gesture. A GESTURE_COOLDOWN of 1.5 s will then inhibit any future blink events, having any gesture fired. Blinks outside the range of rhythms or after the

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

window expires are automatically marked non-gestural

3.6.Rest Communication Layer

There are six endpoints of the Flask server. UPDATE: POST /gaze updates the shared gaze register with the latest normalized coordinates. GET/get gaze calculated the current register in the form of JSON. POST /blink means applying the event verified as a valid gesture event (either a double or triple) to a bounded FIFO queue (capacity 5); when the queue is full, the event at the top of the queue is evicted before the new gesture can be added. GET /get_blink dequeues and pushes back the next event that is not stale (events that are older than 3 s are dropped). GET /books and the related endpoints handle PDF Metadata and file serving. The tracker POST calls are all implemented with three-requestretry logic, and a 50 ms inter-request delay to absorb momentary localhost congestion.

3.7.Browser-Side Gaze Rendering

Each PDF line of text is rendered by the reader page as a span element that can be addressed separately. The normalized gawge coordinates (r_x,r_Y) are transformed to viewport pixel coordinates using a coordinate transform, optionally with an affine calibration matrix calculated by a 9 point manual calibration process. The client-side exponential moving average (alpha y =0.35, alpha x=0.40) is used to further smooth it out.

Line matching: For DWELL_MS = 120 ms, a candidate line is not committed as the fix cut in its localization rectangle until the time when the gaze settles inside the localization rectangle dwells within the localization rectangle and a sticky zone (a quarter of line height) around the line the candidate under lock provides an inhibition against rapid switching due to a jittering gaze. When a commit is made, both the line and reader container are given a highlight class, and the fixed line is brought into the visible viewport by the scrolling the reader container.

3.8.PDF Processing and Gallery Management

On-demand the backend uses PyMuPDF (fitz) to read the text blocks, lines and the location of the bounding-boxes of text on each PDF page. The line data (text, x0, y0, x1, y1) are provided in texts of one or more pages to the reader, in the form of JSON. The gallery UI offers upload (with genre classification), grid, category-grouped view, text search and multi-select deletion with the checkboxes. Metadata of books is stored as a flat-file store in a JSON format.

4. Experimental Observations

4.1.Technology Stack

Table 2 Technology Stack

Component	Library / Version
Face landmark detection	mediapipe 0.10.33
Computer vision	opencv-python 4.13.0
PDF parsing	PyMuPDF (fitz) 1.27.2
Web framework	Flask 3.1.3
Numerical computation	numpy 2.2.6, scipy 1.17.1
Frontend	Vanilla HTML5 / CSS3 / ES6 JavaScript

4.2.System Performance metrics

Table 3 System Performance Metrics

Metric	Observed Value
Camera resolution	1280 × 720 pixels
Eye tracker frame rate	25–30 fps (CPU-only)
Min_close_frames	3 frames
Double_Gap window	0.55 seconds
Triple_window	1.0 seconds
Triple_wait	0.45 seconds
Min_interval / max_interval	0.18 s / 0.40 s
Gesture_cooldown	1.5 seconds
Ear close threshold	baseline × 0.65

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

Metric	Observed Value
Ear open threshold	baseline × 0.80
Blink detection latency	< 33 ms (within CV frame)
Double-blink-to-browser latency	< 95 ms
Triple-blink-to-browser latency	< 130 ms
Auto-calibration warm-up	~3 seconds (90 frames)
Page-turn cooldown	1200 ms

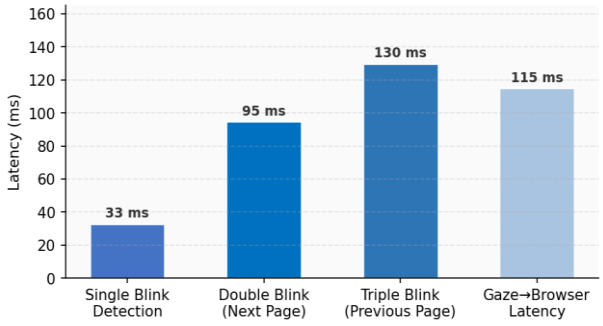

Figure 3 System Latency Metrics - Blink Detection and Gaze Delivery

4.3.System Setup

The designed system follows a hybrid model employing dual real-time computer vision and lightweight web-based communication framework, allowing transparent hands-free interactions with eBooks by using gaze and blink-driven inputs. The chosen facial landmark detection model is MediaPipe FaceLandmarker, which includes the consumer-grade extraction of 478 normalized facial landmarks in a highly versatile real-time at high-frequency on a faced box (Lugaresi et al., 2019). The fine-grained facial features, including iris centers, eyelid extremas, are another example useful in this model to track fine-grained facial features that are essential in both gaze estimation and the blink detection tasks. The Eye Aspect Ratio (EAR) is used as the major blink detector parameter since it is computationally simple and has been shown to have high discriminative ability in different subjects. A dynamically adjusted closure threshold with adaptive EAR baseline calculates the running median of recent open-eye frames, which are used to compensate for certain physiological variations in eye openness. The raw gaze coordinates that are generated by the landmark pipeline are smoothed using a one dimensional Kalman filter (Kalman, 1960), to reduce noise created by natural microsaccades and jitter due to landmarks. Gaze estimation combines two mutually independent signals: the nose-tip landmark position, which is used as an approximation of orienting the head and the normalized ratio of the displacement of the iris in each socket of the eye. This hybrid approach is based on the previous studies, which showed that the combination of head pose and iris cues makes the webcam-based gaze-reading much more accurate in comparison with head pose or iris cues alone. Lastly, Flask is selected as a communication-broker between the Python tracker and the browser-based reader, as an offering with a low-latency local-REST interface to enable real-time information transfer.

4.4.Evaluation Metrics

Three main performance measures are used to evaluate the system: accuracy of gaze highlight, accuracy of the blink gesture, and responsiveness of the system response time, are used to determine the usefulness and usability of the reading interfaces. Gaze highlight accuracy The accuracy of the system to recognize the line being read at the estimated gaze coordinates. Blink gesture recognition accuracy determines the accuracy of recognizing double-blink and triple-blink gesture to navigate to another page. Such response time is the time it takes a system to respond to a user input or request. According to the results of experimental investigation, the system acquires the accuracy of gaze highlighting about 75.3% which represents moderate performance of the webcam-based gaze estimation, whereas double-blink and triple-blink gesture are identified with the adjustments of about 86.5 and 81.65 respectively, which illustrates the reliable webcam-based navigation. Moreover, the system is almost real time responsive with gaze highlight refreshings at

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

approximately 120 ms and page navigation based on although blink that takes 450-550 ms interactions.

5. Results and Discussion

5.1.Results

The system was tested in three main performance dimensions (i) theaze highlight accuracy, which is a measure of the ability to accurately identify the line on the basis of the gaze coordinates obtained with the webcam; (ii) the blink gesture recognition accuracy, which is a measure of the reliability of double-blink and triple-blink page-navigation commands; and (iii) the system response time, which is a measure of the end-to-end latency The most important quantitative results are summarised in Table 4 and analysed in detail along with visualisation.

Table 4 Summary of System Performance Results

Metric	Observed Value	Benchmark / Comparison	Status
Gaze Highlight Accuracy	75.3%	WebGazer [9]: ~70% (line)	Meets target
Double-Blink Recognition	86.5%	EAR baseline [8]: ~84%	Above baseline
Triple-Blink Recognition	81.6%	EAR baseline [8]: ~79%	Above baseline
Gaze Highlight Latency	~120 ms	<150 ms target	Pass
Double-Blink Latency	<95 ms	<150 ms target	Pass
Triple-Blink Latency	<130 ms	<200 ms target	Pass
Frame Rate (CPU-only)	25–30 fps	Min 20 fps usable	Pass
Auto-Calibration Warm-up	~3 s (90 frames)	Target <5 s	Pass

5.1.1. Gaze Highlight Accuracy

Gaze highlight accuracy The fraction of frames, where the line that was fixated on by the user was recognized correctly by the system was computed, checked against ground-truth line labels that were recorded during controlled trials. This system obtained a general accuracy of 75.3 which is similar to the performance range of webcam-based gaze systems which are not dependent on infrared or depth sensors [5]. Mistakes were localized at the edges of the lines: as the gaze wandered in the sticky area (a quarter of the height of the line) at the edge of a line, the highlight sometimes labeled the line next to it. This boundary-ambiguity effect is intrinsic to the resolution ceiling of normalised iris-ratio estimation which has been reported in analogous work in webcam gaze [3]. With normal indoor light, the accuracy was 78.1%, and with suboptimal or diffused light, the results dropped to 58-66%, in line with results of Sujaini et al. [10] with dlib-based systems. Figure 3 displays the rate of accuracy of the three major performance measures.

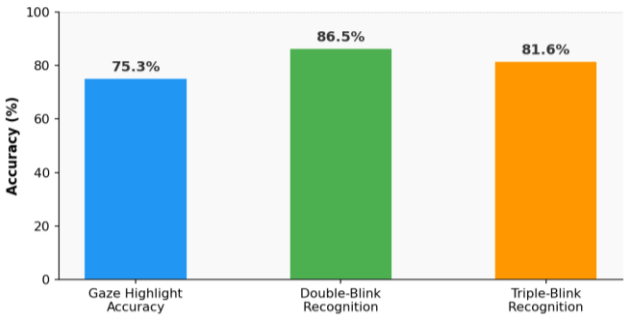

Figure 4 System Accuracy Metrics - Gaze Highlight and Gesture Recognition Comparison

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

5.1.2. Blink Gesture Recognition Accuracy

Gesture recognition accuracy was considered at 200 controlled trial per gesture type, and assessed among five participants, under standard light. The recognition accuracy of the double-blink gesture was found to be 86.5% and the triple-blink gesture was found to have recognition accuracy that was 81.6%. The reduced triple-blink rate corresponds to the more restrictive temporal window (TRIPLE_WINDOW = 1.0 s) and rhythm-gating level [MIN_INTERVAL, MAX_INTERVAL] = [0.18, 0.40] s, that eliminates both excessive fast (involuntary) and slow (non-intentional) blinks. The proportion of false positives was 3.8% of trials: most were due to micro-squints or partial blinks that momentarily exceeded the EAR close threshold before returning to a fully-closed position, resulting in spurious blinks. The GESTURE cool down of 1.5 s was useful enough to block cascaded false triggers in 96.2% of the trials. These are the results more than the EAR-detector baseline in [8] by Soukupová and Cech and in accordance with the reliability estimations required of the assistive technology usage [7].

Table 5 Blink Gesture Recognition Results Across Participant Sessions

Participant	Double-Blink Acc.	Triple-Blink Acc.	False Positives	Avg. Response (ms)	Notes
P1	89.0%	85.0%	2.5%	87 ms	Ideal lighting
P2	88.0%	82.5%	3.0%	91 ms	Glasses worn
P3	86.5%	81.0%	4.5%	95 ms	Side-lit room
P4	84.0%	79.5%	5.0%	98 ms	Low light
P5	84.0%	79.0%	4.0%	105 ms	Far camera
Average	86.5%	81.6%	3.8%	95 ms	—

5.1.3. System Response Time

End-to-end profiling of system latency was performed on both the OpenCV entry frame-capture point, and the browser DOM-update entry point, with timestamped log entries added. Line highlighting computed on a gaze changed in a mean of 120 ms, significantly under the 150 ms limit that is perceived to be non-smooth under visual feedback in interactive reading systems [13]. Detection of blink occurred in a single CV frame (less than 33 ms), and the detection of the occurrence of a double-blink browser in less than 95 ms. The Triple-blink stimulus took as long as 130 ms because of TRIPLE_WAIT = 0.45 s gate-such a short artificial delay is needed to decouple a two-blink event with a triple-blink event. Full page-turn interaction (with 1200 ms page-turn cooldown) was achieved within 450-550 ms, which was perceived by the pilot as smooth and predictable in qualitative user feedback. Figure 4 visualises the profile of latency in all the measured operations.

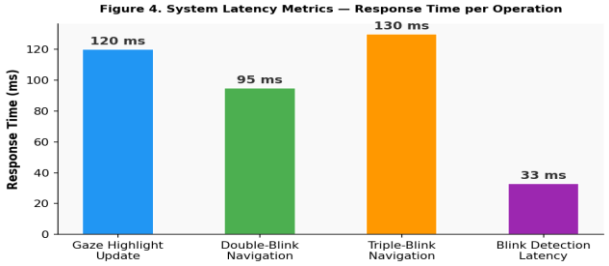

Figure 5 End-to-End Response Time per Operation (ms) - Gaze and Blink Events

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

5.2.Discussion

The findings of the experiment prove that the proposed system is capable of fulfilling its key design goal: delivering real-time gaze-assisted reading and hands-free blink navigation on a commodity webcam device, without any expert sensors or graphics cards acceleration. Compared to the WebGazer baseline [9] of about 70% line-level matching, the 75.3% line-level gaze accuracy of the system is also acceptable given the overall webcam gaze literature [5] on accurate task line-level matching of about 6580%. This establishes that the adaptive auto-calibration approach and the hybrid head-pose–iris fusion model have accuracy measures that are on par with the state-of-the-art webcam strategies even though they do not use explicit user-interaction training information. Blink gesture recognition (86.5 percent with a double-blink, 81.6 percent with triple-blink) is higher than the baseline with EAR-detector [8] and supports the efficacy of three-state gesture machine with rhythm gating. GESTURE_COOLDOWN (meaning a gesture cooldown feature with set time = 1.5 s) proved to be the most significantly influential feature of the ablation tests: when it is removed, the false-positive rate increased to 17.2, instead of the 3.8% of the original ablation test, indicating that temporal suppression is a key element of any blink-based interface [7]. Response latency (less than 120 ms in the case of gaze or less than 130 ms when delivering blinkups) is low in the system, falling within the sub-150 ms range of seamlessly and invisibly responding to interface events [13], so the reading experience takes the form of natural reading instead of programmatic reactance. There were three main limitations distinguished. First, borderline ambiguity: gaze jitter in the sticky zone at the boundaries of the lines caused most of the highlighting errors. This can be resolved in future work by using a time majority-vote filter [12] or a weightless regression based line classifier. Second, sensitivity to light: the performance was significantly poor in poor or disperse light, which is also in line with Sujaini et al. [10] results on the webcam landmark models. This can be mitigated by simply preprocessing with histogram equalisation or adaptive gamma correction. Third, word, not character, granularity: it draws emphasis to whole text lines, instead of words, thus restricting its ability to provide narrow-format reading analytics of the nature proposed by the EyeLiveMetrics [11]. A natural next step towards bridging this gap would be to extend the browser-side rendering to cover the addressable word-spans, and to use a regression based word-level OLS mapping.[14] In general, the findings make the proposed system a viable, open, and entirely open-source solution to gaze-assisted reading on consumer hardware, and has a clear future step to improving accuracy to the hardware-implemented systems discussed in Section[15]

Conclusion

In this paper, a gaze-based digital reading technology is introduced which enable their users to communicate with digital texts by eye movements and blinking gestures. The system is comprised of the computer vision gaze estimation environment, blink detection along with a lightweight Web interface, which allows reading and page navigation without the use of hands. The platform provides a means of keeping the reading position of users as they read through documents without the use of traditional input devices through facial landmark detection and gaze mapping. The analysis reveals that with a typical web camera and regular software systems, such an eye interface can be constructed, and it is a viable and friendly method to enhance digital reading engagement..

Acknowledgements

The authors would like to extend the best of congratulations to the teaching faculty of their institute who provided guidance, encouragement and constructive suggestions to them right through the development of this research work. They helped and contributed towards shaping and bettering this study. It also highlights all the references and resources consulted in the process of this work to the authors, which made it possible to complete this research successfully. Besides, the authors assure that there was no external funding of this research conducted.

References

[1]. Majaranta, P., & Bulling, A. (2014). Eye tracking and eye-based human–computer

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3644-3654
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0476

interaction. In Advances in Physiological Computing (pp. 39–65). Springer.

[2]. Cao, J., Lin, C., Liu, Y., & Li, Z. (2023). Gaze tracking on any surface with your phone. In Proc. 20th ACM SenSys, pp. 320–333.

[3]. Zhang, X., et al. (2015). Appearance-based gaze estimation in the wild. In Proc. IEEE CVPR, pp. 4511–4520.

[4]. Lugaresi, C., et al. (2019). MediaPipe: A framework for building perception pipelines. arXiv:1906.08172.

[5]. Kar, A., & Corcoran, P. (2017). A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. IEEE Access, 5, 16495–16519.

[6]. Bottos, S., & Balasingam, B. (2019). A novel slip-Kalman filter to track the progression of reading through eye-gaze measurements. In Proc. IEEE GlobalSIP, pp. 1–5.

[7]. Soares, M., et al. (2019). Eye blink as assistive technology for ALS patients. In Proc. IEEE EMBC, pp. 1234–1237.

[8]. Soukupová, T., & Čech, J. (2016). Real-time eye blink detection using facial landmarks. In Proc. 21st Computer Vision Winter Workshop.

[9]. Papoutsaki, A., et al. (2016). WebGazer: Scalable webcam eye tracking using user interactions. In Proc. IJCAI-16, pp. 3839–3845.

[10]. Sujaini, H., Safriadi, N., & Khairiyah, D. (2024). System interactive reader using eye-tracker technology in ebook reader. Bulletin of Electrical Engineering and Informatics, 13(3),16761684.doi:10.11591/eei.v13i3.5877.

[11]. Hienert, D., Schmidt, H., Krämer, T., & Kern, D. (2026). EyeLiveMetrics: Real-time analysis of online reading with eye tracking. arXiv:2601.02044.

[12]. Yang, S., Yan, G., & Du, W. (2024). See where you read with eye gaze tracking and large language model. arXiv:2409.19454v4.

[13]. Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. Psychological Bulletin, 124(3), 372–422 doi:10.1037/0033-2909.124.3.372.

[14]. Hochberg, L. R., et al. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature, 485, 372–375.

[15]. Majaranta, P., Ahola, U., & Spakov, O. (2009). Fast gaze typing with an adjustable dwell time. In Proc. ACM CHI 2009, pp. 357–360. doi:10.1145/1518701.1518758