UDC 004.05

V. Oliinyk, S. Korol

AN EFFICIENT REAL-TIME GAZE TRACKING METHOD FOR BROWSER-BASED APPLICATIONS

Abstract: This paper presents a gaze tracking method based on a hybrid gaze direction prediction model, designed for real-time operation in web applications under limited computational resources and without specialized hardware. The proposed approach combines geometric normalization of facial landmarks with a lightweight CNN-Transformer network to estimate gaze direction and project it onto 2D screen coordinates. Designed for scalable and privacy-preserving use in web applications, it addresses the limitations of appearance-only and geometry-only methods. The system uses MediaPipe FaceMesh for 3D landmark detection, followed by normalization, hybrid gaze estimation, and a 9-point calibration procedure using regression-based mapping. A comprehensive experimental setup was developed to evaluate its effectiveness.

Results demonstrate that our approach achieves high angular accuracy and lower jitter during a user active head movement, with real-time inference running entirely in-browser using ONNX Web Runtime. The proposed method is suitable for use in adaptive web interfaces, assistive technologies, educational tools, and behavioral research applications. It offers an accessible pathway for integrating gaze-based interaction into widespread browser platforms without the need for dedicated hardware.

Keywords: gaze tracking, web applications, hybrid gaze direction prediction, browser-based interaction, appearance-based model, screen-space calibration

Introduction

In the era of rapidly evolving digital platforms and increasing interaction with web applications, the need for intuitive and accessible user interfaces is more important than ever [1]. Gaze tracking technologies offer a powerful means of enabling natural interaction by detecting where users are looking on a screen. This capability has applications in interface personalization, usability testing, psychological assessment, and accessibility support. Traditional gaze tracking approaches rely heavily on dedicated infrared-based hardware, offering precise results but at the expense of cost and scalability [2]. These systems are rarely practical for mass adoption, particularly in web-based environments. As an alternative, software-only gaze tracking solutions based on RGB video streams from built-in cameras have emerged as a promising direction. Recent methods have explored the use of either appearance-based models, which learn gaze direction from eye images using deep learning, or feature-based models, which rely on geometric relationships between facial landmarks.

However, each approach has limitations in terms of robustness and accuracy under uncontrolled real-world conditions. This paper proposes a gaze tracking method based on a hybrid gaze direction prediction model, which fuses direction estimation from both deep neural embeddings and geometric eye pose. The method is specifically designed for browser environments, running fully on the client side using JavaScript and lightweight deep learning frameworks such as ONNX Web Runtime. The method uses 3D facial landmarks from MediaPipe FaceMesh to estimate head pose, normalize eye regions, and feed them into a CNN-Transformer model for gaze direction prediction. A 9-point screen-space calibration maps predictions to screen coordinates. Designed for browser-based use, the system supports varying head positions and lighting conditions without relying on specialized hardware. Its modular design enables integration into adaptive interfaces, research tools, and accessibility systems while ensuring user privacy.

Related research and publications

Gaze tracking technologies have evolved significantly over the past two decades, transitioning from high-precision systems relying on infrared illumination and dedicated hardware [3, 4], to more scalable, software-based solutions using standard RGB cameras [5]. Traditional approaches to gaze estimation can be broadly categorized into appearance-based, feature-based, and model-based methods.

Appearance-based models utilize deep learning architectures to infer gaze direction directly from eye or face images. For example, iTracker [6] and GazeNet [7] are CNN-based models trained on large datasets like GazeCapture and MPIIGaze, respectively. These methods achieve high accuracy in controlled environments but are often computationally intensive and unsuitable for in-browser deployment. The ETH-XGaze dataset [8] has also been used to train Transformer-based architectures for improved generalization.

Feature-based methods extract geometric features from facial landmarks – such as iris position, eye corners, or eyelids – to estimate gaze vectors. WebGazer.js [9] is a widely cited browser-native implementation that combines pupil position with screen-space calibration. Similarly, Human.js [10] uses MediaPipe FaceMesh [11] to infer 3D landmark geometry and apply simple vector-based gaze estimation, enabling efficient real-time performance.

Model-based techniques, like those used in OpenGaze [12], reconstruct a simplified 3D face model and solve the Perspective-n-Point (PnP) problem to estimate head pose and project gaze vectors onto the screen. These approaches are more robust to head movements but require camera calibration data, which is not available in most web scenarios.

Several hybrid strategies have recently emerged, combining geometric normalization with deep learning. Notably, Zhang et al. [13] proposed normalization of eye images guided by head pose estimation, while Park et al. [14] demonstrated improved accuracy using head-normalized patches as input to CNNs.

Unlike the majority of existing solutions that were originally developed for offline or native desktop environments, the approach introduced in this study is specifically engineered for web deployment. To our knowledge, it is one of the few methods to integrate such a hybrid direction prediction pipeline tailored for seamless, privacy-respecting use in browser-based applications.

Proposed method

The proposed method is based on a hybrid gaze direction prediction architecture that integrates visual feature learning with geometric normalization to estimate the user's point of gaze in real time within a web browser environment. A schematic representation of the proposed method is shown in Figure 1. The first main stage in the pipeline is face and facial landmark detection [15]. Here, a frame from the webcam stream is processed using the MediaPipe FaceMesh model, which provides 3D coordinates of 468 facial landmarks. From this set, a subset of points around the eyes and iris is extracted for further computation. The next step is head pose normalization, which ensures that variations in head orientation do not affect gaze prediction accuracy. To account for shifts and rotations of the user’s head, the system solves the Perspective-n-Point (PnP) [16] problem using a selected set of 3D facial landmarks from the MediaPipe FaceMesh model. These landmarks include specific points around the eyes, nose, and mouth, which provide sufficient spatial distribution to estimate the user’s 3D head position. The solution yields a rotation matrix and translation vector that describe the user’s head orientation relative to the camera's coordinate system.

This estimated head pose is then used to perform geometric normalization, which aligns the head to a canonical frontal position. A virtual normalized camera is defined with fixed intrinsic parameters (e.g., fixed focal length and optical center), and the original camera view is reprojected into this canonical frame. This transformation significantly reduces inter-subject and intra-subject variability in appearance due to head tilts, rotations, or user-specific anatomy. As a result, the pipeline ensures that identical gaze directions yield similar visual input regardless of how the user is positioned in front of the camera. Following normalization, the relevant image regions containing the eyes are extracted. Eye patches from both the left and right eyes are cropped based on landmark geometry, geometrically aligned using the normalization matrix, resized to a standard resolution (typically 60×36 pixels), and pixel-normalized to reduce illumination variance. These preprocessed image patches serve as the input to the prediction model.

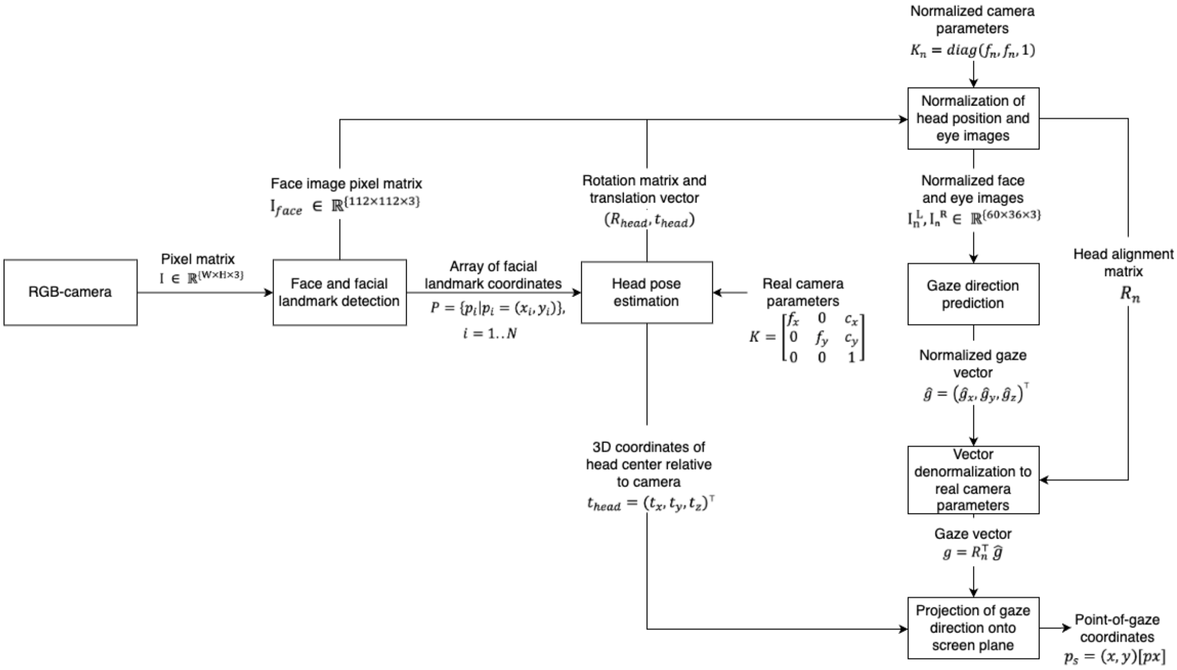
Normalized camera parameters
$K_n = diag(f_n, f_n, 1)$
Normalization of head position and eye images
Face image pixel matrix
$\mathrm{I}_{face} \in \mathbb{R}^{\{112\times112\times3\}}$
Rotation matrix and translation vector
(R_{head}, t_{head})
Normalized face and eye images
$\mathrm{I}_n^L, \mathrm{I}_n^R \in \mathbb{R}^{\{60\times36\times3\}}$
Head alignment matrix
R_n
RGB-camera
Pixel matrix
$\mathrm{I} \in \mathbb{R}^{\{W\times H\times3\}}$
Face and facial landmark detection
Array of facial landmark coordinates
$P = \{p_i | p_i = (x_i, y_i)\}, i = 1..N$
Head pose estimation
Real camera parameters
$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$
Gaze direction prediction
Normalized gaze vector
$\hat{g} = (\hat{g}_x, \hat{g}_y, \hat{g}_z)^\top$
3D coordinates of head center relative to camera
$t_{head} = (t_x, t_y, t_z)^\top$
Vector denormalization to real camera parameters
Gaze vector
$g = R_n^\top \hat{g}$
Projection of gaze direction onto screen plane
Point-of-gaze coordinates
$p_s = (x, y)[px]$

Figure 1. A schematic representation of the proposed method based on a hybrid gaze direction prediction model

At the core of the system lies the gaze direction prediction block, which combines both local texture features and global spatial relationships [17]. The normalized eye images are first processed by a ResNet-18 convolutional neural network (CNN) that extracts spatial features from each patch (Figure 2). These features are then passed to a Transformer encoder, which models contextual dependencies across image regions and captures subtle patterns that correlate with gaze orientation. The combined model outputs a gaze direction vector in 3D, although the final output is typically converted into a 2D angular representation (pitch and yaw) with respect to the normalized camera coordinate system.

To translate these angular outputs into actual screen-space coordinates, a personal calibration step is performed. During a brief 9-point calibration session, the user is instructed to fixate on predefined targets distributed across the screen. For each target, the system records the predicted gaze vector and associates it with the known screen position. These samples are then used to train a lightweight linear regression model that maps predicted gaze directions to pixel coordinates. This calibration compensates for residual inaccuracies introduced by camera distortion, facial variation, or model bias.

Finally, the predicted screen-space gaze point is rendered in real time using a browser-based canvas overlay. This enables dynamic visual feedback and supports interaction use cases such as gaze-based cursor control, attention tracking, or adaptive user interfaces. The entire pipeline is optimized for execution in web environments using ONNX Runtime Web, requiring no GPU acceleration or external server. This allows the system to operate reliably at real-time frame rates on typical consumer devices, while also preserving user privacy by processing all video data locally in the browser.

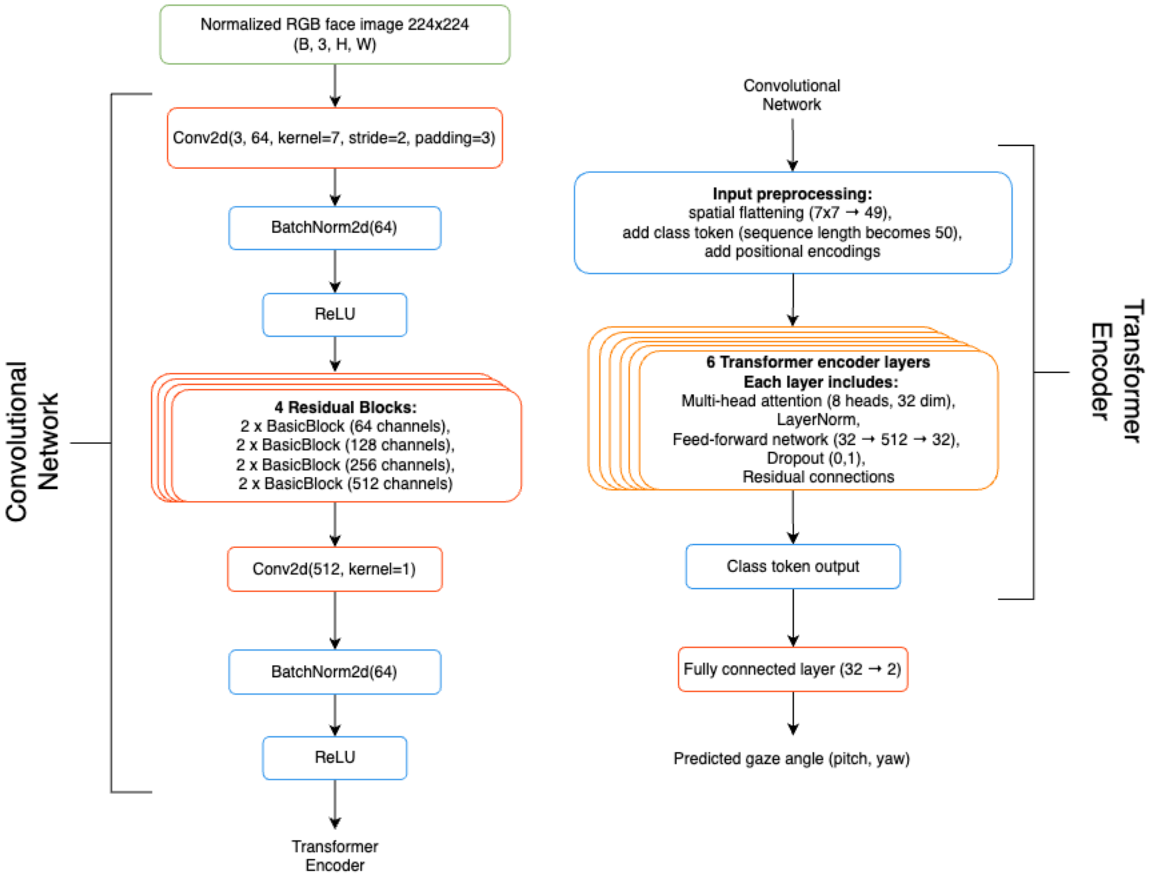
Normalized RGB face image 224x224
(B, 3, H, W)
Conv2d(3, 64, kernel=7, stride=2, padding=3)
BatchNorm2d(64)
ReLU
4 Residual Blocks:
2 x BasicBlock (64 channels),
2 x BasicBlock (128 channels),
2 x BasicBlock (256 channels),
2 x BasicBlock (512 channels)
Conv2d(512, kernel=1)
BatchNorm2d(64)
ReLU
Transformer Encoder
Convolutional Network
Convolutional Network
Input preprocessing:
spatial flattening (7x7 → 49),
add class token (sequence length becomes 50),
add positional encodings
6 Transformer encoder layers
Each layer includes:
Multi-head attention (8 heads, 32 dim),
LayerNorm,
Feed-forward network (32 → 512 → 32),
Dropout (0,1),
Residual connections
Class token output
Fully connected layer (32 → 2)
Predicted gaze angle (pitch, yaw)
Transformer Encoder

Figure 2. Block diagram of the hybrid gaze direction prediction model

Gaze direction prediction model training method

At this stage, a hybrid neural model was trained that combines convolutional layers for extracting local features and a Transformer encoder for modeling global spatial dependencies across the image. The training objective was to construct a regression model capable of predicting gaze direction as a pair of angles pitch and yaw with high accuracy. The model was trained on the XGaze dataset using a custom data loader class designed to accommodate its directory structure. Each subject-specific folder contained RGB image frames, corresponding gaze labels stored as pairs of angles in radians, and intrinsic camera parameters. All annotations were preloaded into memory for faster access during training. Each eye image was preprocessed using standard torchvision transformations, including resizing to 224×224 pixels, pixel normalization, and conversion to tensor format. Head pose normalization was applied before feeding the data into the model.

A series of training experiments were conducted to optimize key hyperparameters, including batch size, learning rate, number of epochs, and learning rate decay strategy.

Table 1 summarizes the core configurations tested, along with the resulting validation angular errors.

Table 1.

Comparison of configurations based on training and validation angular error

№	Batch size	Learning rate	Scheduler	Epochs	Angular error (°)
1	64	1e-3	None	20	7.92
2	128	1e-4	Cosine decay	20	5.87
3	128	5e-5	Cosine decay	30	6.01
4	256	1e-4	StepLR (γ=0.1)	20	6.83
5	128	1e-4	None	20	6.92

The configuration with batch size 128, learning rate 1e-4, and cosine learning rate decay achieved the lowest validation error of 5.87° and was selected for web integration. Further analysis was conducted to assess error stability over time. Figure 3 presents the moving average of angular error across a 20-second test session, with temporal windows of 100 ms. A mild increase in error was observed over time, possibly due to fatigue, lighting drift, or head motion. Despite this, the error remained within acceptable limits (< 5.9°), indicating temporal stability. Also the same figure illustrates the heatmap of the spatial distribution of gaze error as a function of head orientation. As expected, prediction error is lowest (~4.5–6°) in the frontal zone, and increases toward peripheral head poses (±70–80° yaw/pitch), consistent with geometric distortion and visibility loss in extreme poses.

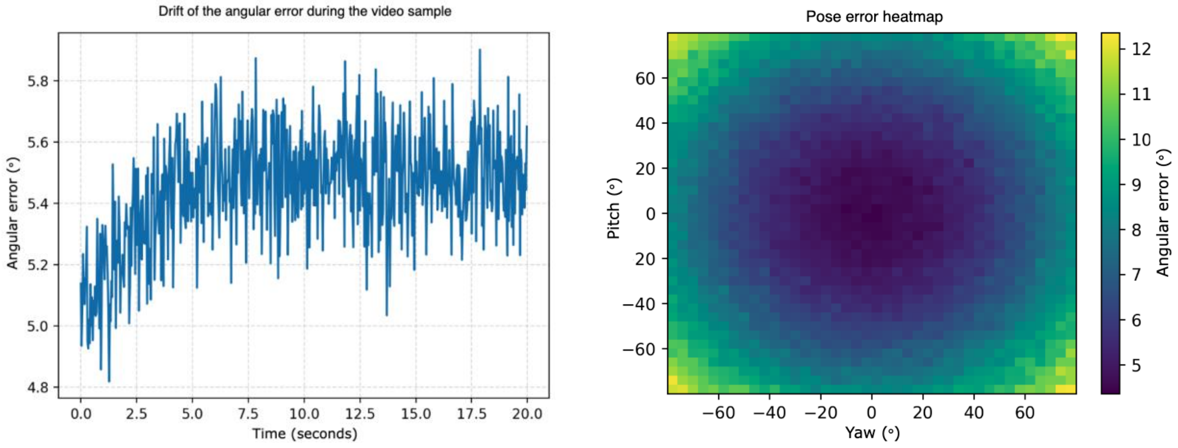

Figure 3. An error drift over the time and pose error heatmap

Experimental Setup

To rigorously evaluate the proposed hybrid gaze tracking method, a structured experimental protocol was developed and implemented in real-world conditions simulating typical web application use. The testing environment was intentionally constrained to client-side browser-based execution without specialized hardware, thereby reflecting the intended deployment scenario. All experiments were performed on a laptop equipped with a 720p RGB webcam and an Intel Core i5 processor (2.3 GHz), 16 GB of RAM, and running Windows 10 with Google Chrome (v136). The camera was positioned directly above the display to replicate common webcam setups. Participants were seated at an approximate distance of 40–50 cm from the screen, with no physical restraints on head movement other than a requirement to keep their face visible and eyes open.

The evaluation protocol involved two key phases: calibration and testing. Calibration was based on a 9-point grid displayed on the screen, requiring the user to fixate on each of the predefined targets. After that, participants were guided through three test sessions, each involving sequential gaze fixation on 12 preselected points uniformly distributed across the screen space. These target coordinates remained fixed throughout all sessions, while the order of appearance was randomized for each trial to reduce prediction bias and user adaptation effects. During each target display (lasting two seconds), predicted gaze points were sampled every 100 ms and averaged to reduce noise. The mean predicted location was then compared to the ground-truth target to compute pixel error. The following test conditions were applied to assess robustness:

- fixed-head condition, where users minimized head motion;
- natural-head condition, allowing free, natural movement;
- challenging lighting condition, simulating mixed or low ambient illumination.

Each method under evaluation including the proposed model and baseline approaches of appearance based and feature based methods – was tested under identical conditions on the same device to ensure fairness and reproducibility. Screen-space error was calculated in pixels. Additional metrics such as jitter (variance over time) and frame rate (FPS) were logged to assess temporal stability and real-time performance.

In total, nine participants of varying ages were involved in the experimental study, resulting in 27 valid experimental sessions. Their adherence to the protocol was monitored throughout the sessions, and in cases where the instructions were not followed correctly, the session was restarted and the corresponding data were excluded from the analysis.

Results and discussion

The results of the experimental evaluation demonstrate the advantages of the proposed hybrid gaze tracking method in comparison with alternative browser-compatible approaches. The study focused on key performance metrics including prediction accuracy, temporal stability, and real-time feasibility. Across all sessions, the proposed hybrid method achieved the best results among the tested techniques. These findings are summarized in Table 2.

Figure 4 presents the boxplots of RMSE values across all methods. The hybrid model consistently shows the lowest median error and smallest interquartile range, reflecting its accuracy and robustness. In contrast, feature-based method demonstrated the highest variance and outlier sensitivity, indicative of instability under natural use conditions.

To assess runtime stability, frame rate (FPS) was monitored over time for each of the three evaluated methods. Figure 5 illustrates the FPS dynamics throughout a 60-second session. The graph reveals a distinct drop in performance at the onset of the testing phase (marked by the dashed vertical line), corresponding to the initialization of calibration routines. Despite this, the proposed hybrid model maintained a consistent processing rate of approximately 17 FPS, while the appearance-based and feature-based methods operated at average rates of 23 FPS and 28 FPS, respectively. Although the hybrid approach exhibits slightly reduced throughput, it remains suitable for real-time operation in web contexts.

Table 2.

Comparison of gaze tracking methods by accuracy, stability, and performance

Method	**Mean Error (pixels)**	**Standard Deviation (pixels)**	**FPS (frames/s)**
Feature-based method (MediaPipe FaceMesh)	97	32	26
Appearance-based method (convolutional network)	79	38	23
Method based on hybrid gaze direction prediction model (proposed method)	68	25	17

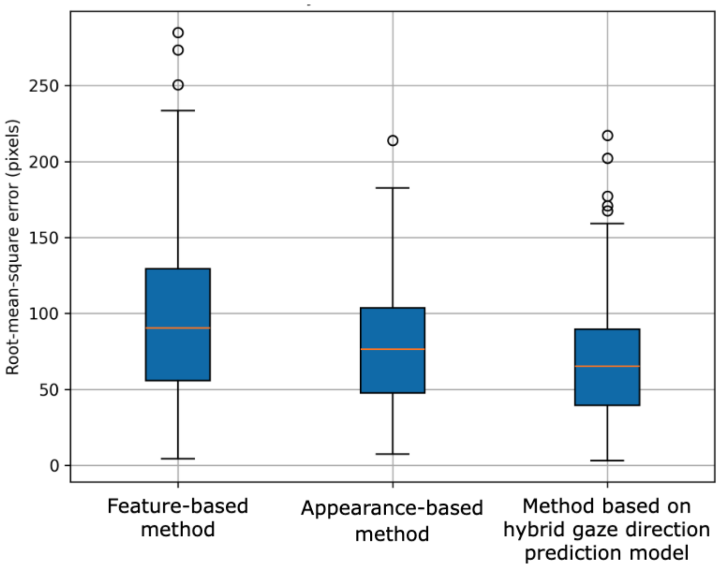

Figure 4. Summary of error for each method

A heatmap of root-mean-square error (RMSE) per target point (Figure 6) confirms the hybrid method's spatial precision. The lowest error was observed at peripheral targets (e.g., 55 px at the upper left), while slightly higher errors were concentrated in the central area (up to 85 px), possibly due to overlapping gaze vectors and calibration biases.

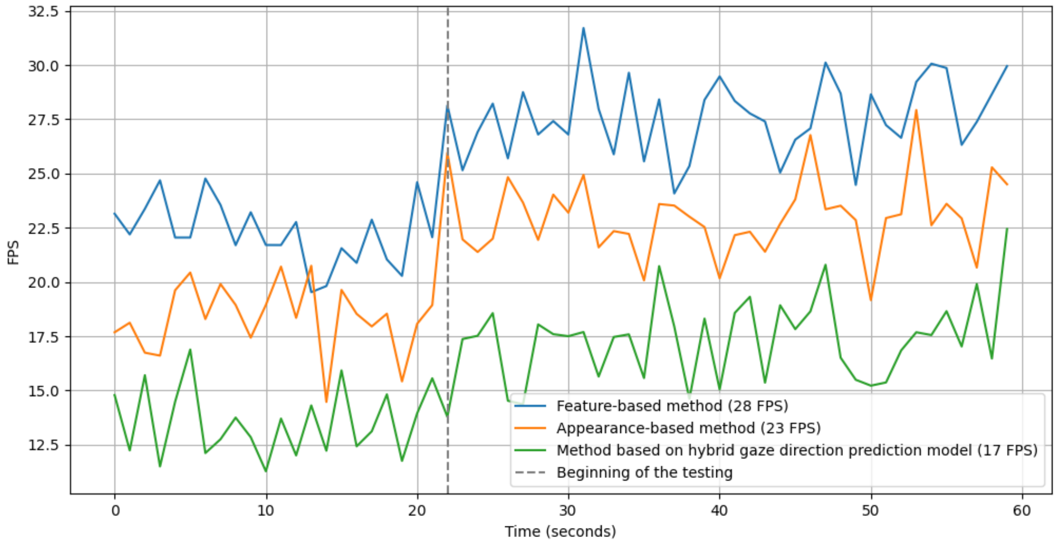

Figure 5. Root mean square error per target test point for selected methods

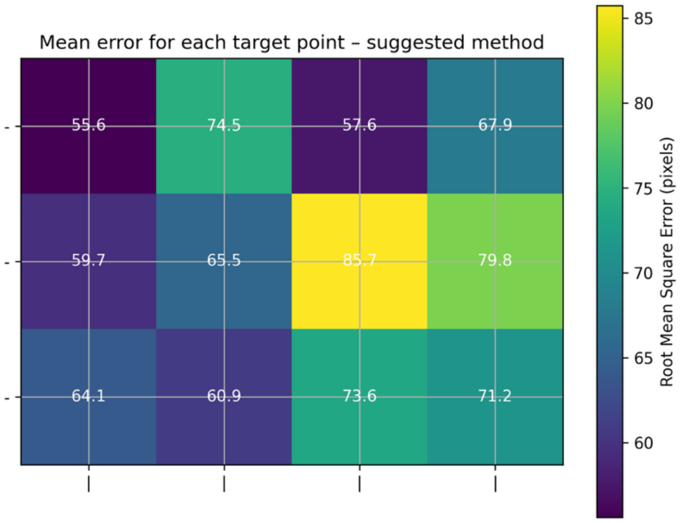

Figure 6. Mean error of suggested method for each target point for all participants data

Overall, the experimental findings confirm that the hybrid method offers the most balanced trade-off between accuracy and stability, albeit at the cost of lower frame rates. In terms of prediction stability, the hybrid model demonstrated the narrowest error variance and lowest median in boxplot analysis, reinforcing its robustness across users and sessions. Despite a moderate reduction in processing speed, the system remained responsive throughout the testing period. The frame rate dynamics plot illustrates stable performance under typical browser conditions, with only minor latency introduced during the calibration phase. Compared to other evaluated methods, which achieved higher throughput but exhibited greater variance and lower precision, the hybrid model offers a compelling balance of spatial accuracy, prediction stability, and runtime feasibility. These findings confirm its suitability for real-time browser deployment, particularly in scenarios where high-precision gaze estimation and privacy-preserving client-side execution are essential.

Conclusion

This paper presents a novel hybrid gaze tracking method optimized for real-time deployment in browser-based environments. By combining geometric normalization with deep neural gaze direction prediction and lightweight screen-space calibration, the proposed system achieves high spatial accuracy, robust temporal stability, and practical runtime performance using only standard RGB cameras and client-side execution. Experimental results across diverse participants confirm that the hybrid approach outperforms both appearance-based and feature-based baselines in terms of accuracy and consistency, despite a modest trade-off in frame rate. The method demonstrates strong generalization across users and environmental conditions, maintaining low RMSE values and stable inference at 17 FPS within modern browsers. These findings validate the feasibility of deploying accurate, privacy-preserving gaze tracking directly in the browser without specialized hardware or external computation. The proposed solution is well-suited for integration, offering a scalable and accessible alternative for real-time gaze interaction in the web ecosystem.

References

1. Oliinyk V. Method for improving accuracy of mobile AR navigators. ISJ Industry 4.0.–2020. Vol. 5, Is. 1. – Pp. 21-22.

2. Chen Z., Shi J., Geng J. A review of gaze estimation methods with head-mounted or remote eye trackers. Sensors. – 2020. – Vol. 20(4). – Pp. 1–23.

3. Hansen D.W., Ji Q. In the eye of the beholder: A survey of models for eyes and gaze. IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2010. – Vol. 32(3). – Pp. 478–500.

4. Duchowski A.T. Eye Tracking Methodology: Theory and Practice. – Springer, 2007. – 333 p.

5. Zhang X., Sugano Y., Fritz M., Bulling A. Appearance-based gaze estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2015. – Pp. 4511–4520.

6. Krafka K., Khosla A., Kellnhofer P., et al. Eye tracking for everyone. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – Pp. 2176–2184.

7. Zhang X., Sugano Y., Fritz M., Bulling A. MPIIGaze: Real-world dataset and deep appearance-based gaze estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 41(1). – Pp. 162–175.

8. Zheng Y., Zhang X., Sugano Y., Bulling A. ETH-XGaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In: European Conference on Computer Vision (ECCV). – 2020. – Pp. 365–381.

9. Papoutsaki A., Sangkloy P., Laskey J., Huang J., Hays J. WebGazer: Scalable webcam eye tracking using user interactions. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI). – 2016.

10. Kumar A., Singh M., Singh A. Human.js: Real-Time Human Pose Estimation Using JavaScript. International Journal of Computer Applications. – 2020. – Vol. 176(30). – Pp. 1–5.

11. Lugaresi C., Tang J., Nash H., et al. MediaPipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172. – 2019.

12. Zhang X., Sugano Y., Bulling A. OpenGaze: Open Source Toolkit for Camera-Based Gaze Estimation and Interaction. arXiv preprint arXiv:1901.10906. – 2019.

13. Zhang X., Sugano Y., Bulling A. Learning gaze representations with a saliency-aware loss. In: International Conference on Computer Vision (ICCV). – 2019. – Pp. 7724–7733.

14. Park S., Spurr A., Hilliges O. Deep pictorial gaze estimation. In: European Conference on Computer Vision (ECCV). – 2018. – Pp. 721–738.

15. Oliinyk V., Ryzhiy A. An efficient face mask detection model for real-time applications. Adaptive Systems of Automatic Control: Interdepartmental Scientific and Technical Collection. – 2022. – No. 1(40). – Pp. 54–64.

16. Qiao R., Xu G., Wang P., Cheng Y., Dong W. An accurate, efficient, and stable perspective-n-point algorithm in 3D space. Applied Sciences. – 2023. – Vol. 13(2). – Pp. 1111.

17. Korol S.P., Oliinyk V.V. Generalized model of the user gaze tracking process in web applications. In: Proc. of the XIII Int. Scientific and Practical Conf. on Information Systems and Technologies “Infocom Advanced Solutions 2025”. Kyiv, Ukraine. – 2025.