



IoT-Enabled Camera-Based Eye Tracking and Data Analysis for Real-Time

F. Aytac Durmaz¹²³, Irem Karagoz¹³,
Taha Cirpici⁴, Saadet Kilic¹, Berkay Ucar⁵,

Gokcer Eskikurt¹⁶, Ezgi Ildirim¹⁶, Irem Kocer¹⁶,
Cenk Efe Bikmaz¹⁶, Askin Mirachan Ergen¹⁶

¹ Istinye University, Istanbul, Turkey

² Bogazici University, Istanbul, Turkey

³ Pievision Technology AS, Istanbul, Turkey

⁴ Türkisch-Deutsche Universität, Istanbul, Turkey

⁵ Istanbul Beykent University, Istanbul, Turkey

⁶ Istinye University Innovative Centre of

Applied Neuroscience (IsuCAN), Istanbul, Turkey

aytacd@gmail.com, iremmerik@gmail.com,
tahacirpici3@gmail.com, kilicsaadet04@gmail.com,
berkayucar60@gmail.com

geskikurt@istinye.edu.tr, ezgi.ildirim@istinye.edu.tr,
kocar.irem513@gmail.com, efebahser@gmail.com,
askinmirachan@gmail.com

Abstract— The high cost and logistical constraints of specialized eye-tracking hardware limit its widespread application in clinical research and diagnostics. To address these limitations, a low-cost, scalable eye-tracking system is engineered to operate with a standard webcam, providing an objective tool to understand attentional processes associated with psychological conditions such as eating disorders, anxiety, and OCD. The core innovation is a dynamic calibration method that continuously maps a user's eye features to on-screen coordinates as they follow a smoothly moving target. This process generates a dense, individualized dataset of several thousand points, a significant improvement over traditional 9- or 16-point static calibrations. This rich data enables a LightGBM machine learning model to perform robust gaze prediction. The system achieves high accuracy, validated through quantitative evaluation, with a low mean absolute error (MAE) that translates to a minimal angular error. Leveraging computer vision algorithms and this novel calibration approach transforms accessible technology into a powerful, objective tool for analyzing the attentional and cognitive processes underlying various psychological conditions.

Keywords—eye tracking, attention deficit, emotional state, IoT, machine learning, webcam, dynamic calibration, mental health.

I. INTRODUCTION

Eye-tracking technology offers a powerful tool for objectively analyzing human attention and cognition, with wide-ranging applications in human-computer interaction (HCI), clinical psychology, usability testing, and educational research [1]. However, the widespread adoption of research-grade systems is frequently impeded by prohibitive costs and the requirement for specialized hardware, creating a significant barrier for many potential applications [2].

Our purpose is to overcome these limitations by developing an eye-tracking system engineered to be simultaneously accurate, sensitive, and highly accessible. This paper presents a system designed for advanced research purposes that is built entirely upon standard hardware (a webcam) and open-source software, ensuring ease of installation and implementation. It is architected for versatility, making it applicable to different analysis and measurement techniques, and includes provisions for cloud integration to support scalable, large-scale data collection and analysis.

The primary contribution of this work is a novel dynamic calibration method that forms the foundation of the system's accuracy. Unlike traditional static calibration, our approach continuously collects thousands of data points, creating a rich, personalized map of the user's gaze [3]. This dense dataset enables a machine learning model to perform robust gaze estimation. This paper details the system's architecture, the underlying mathematical principles of its components, and its empirical validation, demonstrating that an accessible, low-cost platform can achieve the precision required for scientific research.

II. SYSTEM AND METHODOLOGY

Our methodology is structured in four distinct stages, designed to convert raw pixel data from a webcam into meaningful psychological and attentional metrics.

A. System Architecture

The system's pipeline is implemented in Python, leveraging the OpenCV library for image processing tasks. The architecture is modular:

1. *Image Acquisition*: A standard webcam captures the user's facial video stream at a minimum of 100 frames per second (FPS).
2. *Feature Extraction*: The MediaPipe library is employed to detect a dense mesh of 468 facial landmarks in real-time [4]. A specific subset of these landmarks, corresponding to the eye regions, is isolated for further processing.
3. *Noise Reduction*: A Kalman filter is applied to the landmark coordinates to smooth the data, mitigating the jitter inherent in webcam video streams [5].
4. *Model Training & Gaze Prediction*: A pre-trained LightGBM model uses the processed landmark data to predict the (x, y) screen coordinates of the user's gaze [6].
5. *Data Processing and Storage*: The system is designed to be integrated with a cloud backend for large-scale data aggregation and analysis, enabling longitudinal studies and remote monitoring.



Fig.1 Flowchart of the working system.

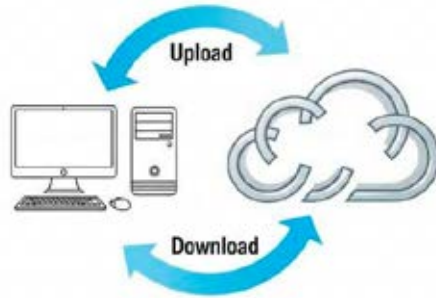


Fig.2 Communication with Cloud-based platform. Once the system completes the real-time data processing, it uploads the resulting visualizations to a cloud backend. This provides users with on-demand access to their stored session data from any location.

B. Dynamic Calibration

The cornerstone of our system's accuracy is its dynamic calibration process. Traditional systems ask the user to fixate on a few static points (e.g., 9 or 16). This is fast but leaves large areas of the screen unmapped, forcing the model to interpolate and often leading to significant errors.

Our approach requires the user to follow a target that moves smoothly and continuously across the screen in a Lissajous (curve-like) pattern for approximately 40 seconds. During this period, we collect thousands of data points. For each frame, we store a tuple containing the ground-truth screen coordinates of the target and the corresponding user's eye features. This rich, continuous dataset provides the machine learning model with a comprehensive map of the user's gaze across every region of the screen, drastically improving mapping accuracy and reliability.



Fig.3 All calibration points captured together.

C. Experimental Apparatus

To ensure a stable data acquisition environment, a custom head support apparatus was designed based on user-centered

ergonomic and biomechanical principles (Fig. 4). Fabricated using a 3D printer, the apparatus features a chin rest ergonomically contoured to the user's anatomy, which effectively keeps the head and chin immobilized throughout the trials. Furthermore, its custom camera mount positions the webcam on an extended arm at a fixed distance, ensuring an unobstructed line of sight to the screen and thereby fostering a more comfortable experimental experience. The webcam streams video data to the host computer via a standard USB connection for real-time processing.



Fig.4 Engineered to mitigate motion artifacts, this 3D-printed apparatus uses an ergonomic design to comfortably immobilize the user's head and maintain a consistent viewpoint for the camera.

D. Model Training

The data collected during calibration is used to train a LightGBM (Light Gradient Boosting Machine) model. LightGBM was chosen for its high efficiency, low memory usage, and excellent performance on tabular data.

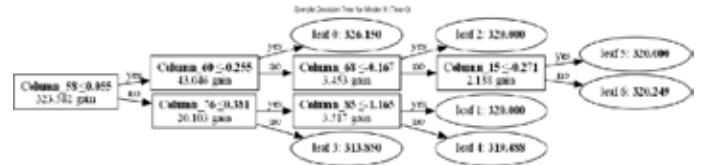


Fig.5 A visualization of a representative decision tree from the LightGBM model trained on our dataset. It illustrates the model's decision-making process, showing the series of feature-based splits at the nodes (branches) that lead to a final prediction value at a terminal leaf.

1. **Feature Engineering:** Before training, we extract key features from the facial landmarks. The most critical is the Eye Aspect Ratio (EAR), which quantifies the openness of the eye and helps disambiguate vertical gaze shifts from blinks or changes in head pose [7]. The EAR is calculated using the distances between vertical and horizontal eye landmarks.

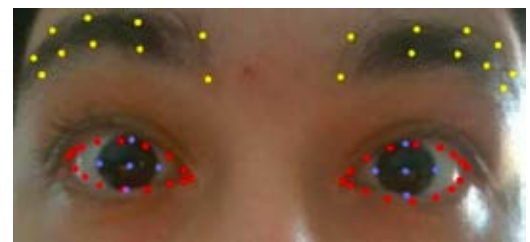


Fig.6 The selected landmarks from the eyelids, eyebrows, iris, and pupil, which are used to train the model and perform real-time tracking of the eye region.

The EAR is defined by the equation:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2 \times \|p_1 - p_4\|}, \text{ where } p_1, \dots, p_6 \text{ are the 2D landmark locations.}$$

2. **Noise Filtering:** To reduce signal noise from minor head movements and sensor imperfections, a Kalman Filter is applied. The filter predicts the next state of the landmark positions based on their current state and recursively corrects this prediction as new measurements arrive. The state update equation is conceptually:

$$\hat{x}_k = K_k z_k + (1 - K_k) \hat{x}_{k|k-1}, \text{ where } \hat{x}_k \text{ is the estimated state, } z_k \text{ is the measurement, and } K_k \text{ is the Kalman gain.}$$

3. **Training Process:** The model is trained on 80% of the calibration data, with the remaining 20% reserved for validation (Hold-Out Cross-Validation). The input features include the filtered coordinates of the eye landmarks and the EAR, while the target labels are the known (x, y) coordinates of the on-screen target.

E. Mapping and Visualization

Once the model is trained, it can predict a user's gaze coordinates in real-time. To interpret this raw data, we employ these three visualization techniques [8]:

- **Heatmaps:** These visuals aggregate gaze data over time, using a color gradient (e.g., red for high, blue for low) to show which areas of a stimulus received the most visual attention.

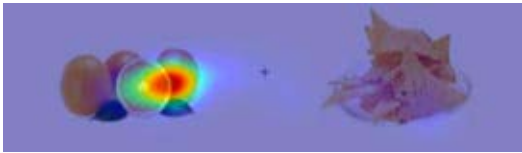


Fig.7 A heatmap showing the gaze distribution of a single participant during the experiment.

- **Scanpaths:** These connect sequential fixation points with lines and represent fixation durations by the size of the circles at each point. This reveals the temporal pattern of visual exploration.



Fig.8 A scanpath illustrating the sequence of fixations made by a single participant during the experiment.

- **Grid Template:** This method overlays the stimulus with a uniform 8x8 grid, where each cell displays the percentage of total fixation duration it received from the user.

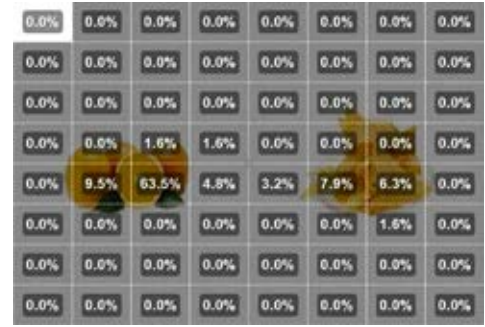


Fig.9 A grid-based template showing the percentage of gaze duration in each cell for a single participant during the experiment.

III. EXPERIMENTAL EVALUATION

To evaluate the performance of our system, we conducted an experiment involving 30 participants. Following a dynamic calibration process, each participant was presented with 90 consecutive, validated visual stimuli related to food choice, specifically designed for use in eating disorder research. Each stimulus consisted of a pair of images representing either high-calorie, low-calorie, or neutral food items. The task aimed to measure and map participants' visual tracking patterns between the two simultaneously presented images.



Fig.10 A sample visual stimulus presented to participants during the experiment.

The error rate of the experiment was quantified in a two-stage process. First, to measure the performance of the machine learning model itself, we employed a hold-out cross-validation method. Our dataset, comprising over 5000 samples, was partitioned into an 80% training set and a 20% test set, with the latter being completely unseen by the model during training. Following the training phase, we calculated the model's mean prediction error in pixels on the test set for both the X and Y axes, using the aforementioned Mean Absolute Error (MAE) formula.

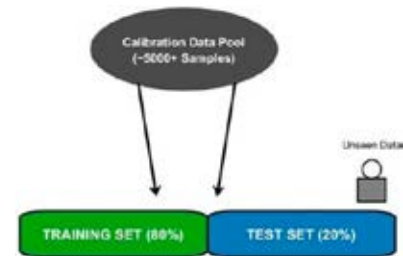


Fig.11 The data splitting procedure, dividing the dataset into training (80%) and test (20%) sets.

In the second stage of our error evaluation, participants were first asked to complete the calibration process. Subsequently, they were instructed to look at any point on the screen and click the mouse to register that exact location as the ground truth. At the moment of each click, the error was calculated as the difference between the X and Y coordinates of the mouse cursor's position and the coordinates concurrently predicted by

the machine learning model. The data aggregated from this repeated procedure are presented in the table below.

TABLE I. FINAL MEAN ERROR RATES

Error Measurements	Average Absolute Errors		
	x	y	Euclid
Model Error	10 to 15 pixels	5 to 10 pixels	14,6 pixels
Application Error	52.25 pixels	53.69 pixels	81.65 pixels

Fig.12 The table of Error Measurements x Average Absolute Errors.

Evaluation Metrics

Performance was quantified using established metrics:

1. **Mean Absolute Error (MAE):** This metric calculates the average absolute difference between the predicted gaze coordinates and the actual target coordinates in the test set. It provides a direct answer to the question: "On average, how many pixels off is the prediction?" It is calculated separately for each axis:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

where y_i is the actual coordinate and $\hat{y}_i$ is the predicted coordinate.

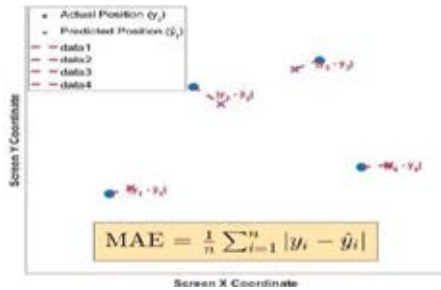


Fig.13 A visualization of the MAE computation process. The diagram shows how individual prediction residuals (the error between actual and predicted positions) are averaged to yield a single metric of model performance.

2. **Gaze Shift Error (Euclidean Distance):** To provide a single error value, we calculate the Euclidean distance between the predicted point (p_x, p_y) and the actual point (a_x, a_y).

$$Error_{pixel} = \sqrt{(p_x - a_x)^2 + (p_y - a_y)^2}$$

3. **Angular Error:** Pixel error is dependent on-screen resolution and viewing distance. A more robust metric is the angular error, which measures the visual angle between the predicted and actual gaze points. It is calculated as:

$$Error_{angular} = \arctan\left(\frac{Error_{pixel} * P_{size}}{D}\right),$$

where P_{size} is the physical size of a pixel and D is the distance from the user's eye to the screen. An error of $<1^\circ$ is generally considered high accuracy for webcam-based systems [2].

Using this formula, the system's average pixel error corresponds to an angular error of 1.73° .

IV. CONCLUSION

This paper detailed the design and validation of a high-accuracy, low-cost eye-tracking system that replaces traditional static calibration with a novel dynamic technique. This approach creates a dense, personalized data map, allowing a LightGBM model to achieve significantly lower prediction error. To ensure affordability, we leveraged inexpensive 3D printing to fabricate a custom ergonomic head-support apparatus, which drastically reduced hardware costs while ensuring data quality. Our results demonstrate that this system achieves an accuracy that is highly competitive for webcam-based solutions, successfully transforming accessible hardware into a powerful scientific instrument for the objective analysis of human behavior. This work opens the door for its use in scalable research and real-world applications, particularly in budget-constrained environments. By addressing key barriers of cost and complexity, our approach can make empirical eye-tracking research more accessible. It empowers a broader community of scientists and developers to explore the complexities of human attention and cognition in diverse fields ranging from clinical psychology to usability testing.



Fig.14 The experimental setup shown during a live data collection session with a participant.

REFERENCES

- [1] K. Harezlak and P. Kasprowski, "Application of eye tracking in medicine: A survey, research issues and challenges," *Computerized Medical Imaging and Graphics*, vol. 65, pp. 176-190, Apr. 2018.
- [2] K. Krafa *et al.*, "Eye tracking for everyone," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2176-2184.
- [3] J. Liu, J. Chi, and Z. Yang, "A review on personal calibration issues for video-oculographic-based gaze tracking," *Frontiers in Psychology*, vol. 15, Art. no. 1309047, Feb. 2024.
- [4] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile GPUs," *arXiv preprint arXiv:1907.06724*, 2019.
- [5] D. Sauter, B. J. Martin, N. Di Renzo, and C. Vomscheid, "Analysis of eye tracking movements using innovations generated by a Kalman filter," *Med. & Biol. Eng. & Comput.*, vol. 29, no. 1, pp. 63-69, Jan. 1991.
- [6] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [7] T. Soukupova and J. Cech, "Eye blink detection using facial landmarks," in *Proc. 21st Comput. Vis. Winter Workshop (CVWW)*, Rimske Toplice, Slovenia, Feb. 2016.
- [8] G. Drusch, J. C. Bastien, and S. Paris, "Analysing eye-tracking data: From scanpaths and heatmaps to the dynamic visualisation of areas of interest," in *Advances in science, technology, higher education and society in the conceptual age (STHESCA)*, 2014, p. 25.