www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(2), 2026, 243-255. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2026.v17i2.1304

Binocular CNN for Robust Webcam Gaze Tracking: Calibration and Temporal Filtering under Static and Dynamic Protocols

Joel Urquiza-Martínez, Gabriel González-Serna, Andrea Magadan-Salazar, Nimrod González-Franco, Jonathan Villanueva-Tavira

e-mail: {m24ce066, gabriel.gs, andrea.ms, nimrod.gf, jonatha.vt}@cenidet.tecnm.mx.

Abstract. We compare single-eye (SET) and double-eye (DET) convolutional models for gaze tracking using webcams. A lightweight CNN processes paired left/right eye crops to predict on-screen gaze and is evaluated under two protocols: (i) a static 9-point calibration and (ii) a dynamic "blue-ball" trajectory across screen corners and edges. To enhance generalization, data augmentation (pose, lighting, blur/noise, partial occlusion) and a broader participant pool are employed. To further improve robustness and interpretability, the system integrates temporal smoothing via a Kalman filter and a DBSCAN-based clustered calibration stage, thereby reducing jitter, suppressing outliers, and stabilizing gaze trajectories under real-world conditions. Performance is measured by pixel error, angular error, and trajectory stability. Under identical training conditions, DET consistently reduces error and yields smoother tracking than SET, particularly under asymmetric lighting and partial occlusions. Using only a standard laptop webcam and a lightweight CNN, the proposed approach achieves real-time performance without infrared sensors or proprietary licenses, providing an accessible and reproducible solution for gaze estimation on consumer hardware.

Keywords: Eye-tracking, CNN, User Experience, Gaze estimation, Calibration, Webcam, HCI.

Article Info
Received Dec 26, 2025
Accepted Jan 19, 2026

1. Introduction

Eye tracking is a key capability for human–computer interaction, accessibility, behavioral research, and immersive media. However, many high-precision systems still rely on corneal-pupil reflection (PCCR) with infrared (IR) optics, which requires controlled environments, high-resolution sensors, and proprietary software, increasing acquisition and operational costs and limiting their use outside laboratories or well-funded settings (Vermeeren, et al., 2010). In contrast, methods based on convolutional neural networks (CNN) can leverage conventional webcams to collect eye-tracking data, substantially lowering entry barriers and facilitating integration into everyday environments (Daniel, 2021).

Despite these advances, webcam-based systems remain sensitive to real-world factors such as partial occlusions from eyelids or glasses, asymmetric lighting, small head movements, and user variability (Pix4d 2025). A common simplification in previous work is to train and evaluate the model using only one eye (monocular) (Vidhya et al., 2025), which reduces input dimensionality but discards complementary information when both eyes are considered (binocular). This monocular restriction can amplify errors under conditions most relevant to practical use (e.g., uncontrolled lighting).

This article addresses the bottleneck problem caused by the cost and licensing of specialized eye-tracking sensors, as well as the limitations of the monocular model. It was analyzed whether fusing binocular information in a CNN improves robustness and accuracy compared to the monocular version, while maintaining the same optimization and data handling. This protocol for moving visual targets has been used in recent evaluations with webcams (Vidhya et al., 2025) to verify whether the prediction remains within a radius around the stimulus across all screen regions.

To improve generalization under conditions of capture with conventional hardware, we expanded training through augmentation techniques (Dilmegani, C. 2025; pose, lighting, blur/noise, partial occlusion) and increased participant diversity. We report performance in pixels on the screen plane, angular error (in degrees), and trajectory stability in the moving-target test. In summary, we position binocular appearance-based gaze estimation as a practical, low-licensing alternative to IR/PCCR systems, maintaining webcam accessibility and mitigating failures that persist in monocular CNN approaches. Additionally, because they rely on a single visual channel, these architectures are more vulnerable to asymmetric lighting, partial occlusions, and reflections on glasses or eyelids.

In contrast, the proposed approach introduces a binocular architecture that merges the left and right eye crops into a single input tensor, preserving the original CNN topology. This design allows the model to learn complementary relationships between the eyes without significantly increasing computational complexity, thereby improving robustness to visual noise, spatial consistency, and gaze-point accuracy, even with consumer webcams.

2. Methodology

An experimental design was implemented comparing a monocular model with a binocular one for gaze estimation using a webcam on conventional computer setups. The methodology is defined by: (i) a data collection algorithm that synchronizes on-screen stimuli with eye captures and target coordinates; (ii) the construction of the dataset with subjects, partitions, and augmentations; (iii) training a lightweight CNN under identical configurations for both conditions; (iv) the description of two evaluation experiments (static calibration with nine points and dynamic trajectory of a blue sphere); and (v) the practical application of these experiments to measure pixel/degree error and trajectory stability.

Figure 1 shows the overall functional flow of the proposed system. The process begins with video capture from a standard webcam, from which input frames are extracted (Figure 1, block A). Next, the face and eye detection module uses MediaPipe Face Mesh to locate facial landmarks and segment the regions of both eyes (Figure 1, block B). These regions undergo geometric normalization, which corrects for tilt variations (yaw/pitch) and converts the image to grayscale (RGB2GRAY) to standardize input (Figure 1, block C). Finally, the CNN model processes the normalized eye images and predicts gaze coordinates (x, y), recording each output along with its timestamp (Figures 1, blocks D and E).

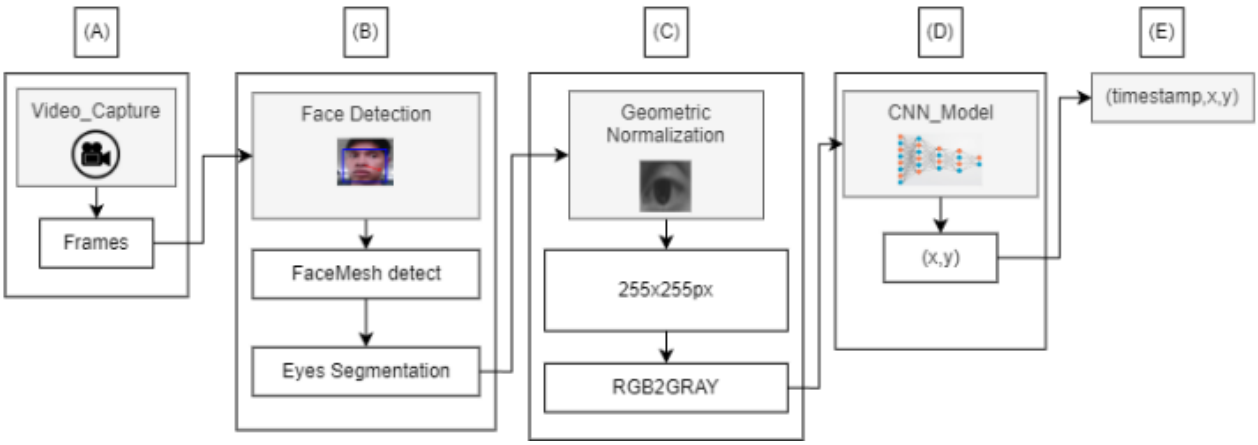


Figure 1 Functional flow of both models.

2.1. Data Collection Algorithm

A sequential, controlled, and reproducible protocol was designed to generate input-label pairs for training and evaluating gaze estimation models with conventional webcams. The screen is divided into a 4×4 grid (Figure 2), and a fixation point appears at the center of each cell for five seconds, instructing the participant to keep their gaze on that target. Sessions are conducted under controlled lighting conditions and at an approximate distance of 60 cm from the monitor to homogenize capture geometry.

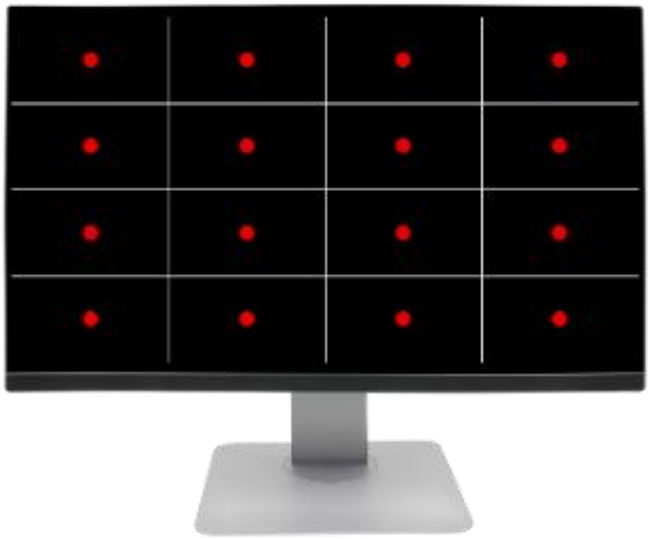


Figure 2 Calibration of 16 points.

Materials and software. An integrated or USB webcam was used along with a standard monitor.
Stimulus sequence. The animation moves across the 16 grid centers, displaying a pulsating red dot at each position, dynamically drawn along with reference lines. This presentation provides a clear spatial guide of the screen plane and facilitates precise synchronization with captured frames. The system calculates the center of each cell (center_x, center_y) and uses it as the target label for the corresponding sample.

A full grid traversal lasts approximately 80 seconds (16 cells × 5 seconds), creating a systematic and balanced set of eye views and screen labels under stable conditions. The choice of a 16-region grid and the use of animated stimulus interfaces align with recent practices in webcam-based gaze estimation, where the grid ensures uniform spatial coverage of the display and the animation allows for standardized and reproducible evaluation.

2.2. Dataset

The dataset was organized by participant to prevent unwanted mixing between subjects and to facilitate clean partitions. For each participant, left- and right-eye captures were stored separately (Figure 3), and within each eye, samples were organized according to the 16 quadrants of a 4×4 grid shown on the screen during acquisition.

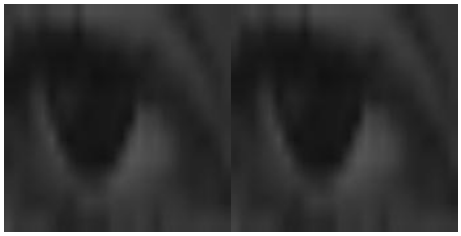


Figure 3 Segmentation of the image into both eyes of 256x256 px.

After applying controlled data augmentations (slight rotations, moderate zoom, lighting variations, blurring, and noise), a robust corpus was formed that balances the distribution across cells and reduces the risk of overfitting to individuals or specific conditions (Murel, J. et al., 2025) (Figure 4). As a result of this process, a final dataset consisting of approximately 480,000 images was obtained.

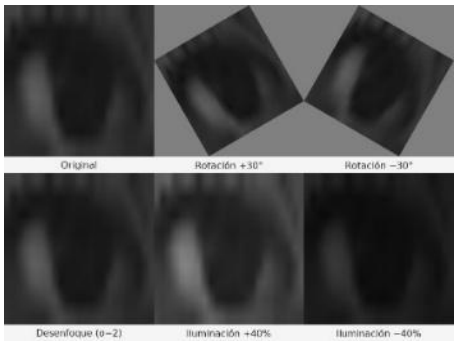


Figure 4 Transformations are applied for data augmentation.

The extraction of eye regions was the core of the preprocessing. In each frame, facial landmarks were estimated using MediaPipe Face Mesh, and from the subset corresponding to each eye, a precise polygonal mask was drawn. Using this reference, a region of interest (ROI) was defined with a safety margin to prevent cropping of the eyelids or eyelashes. Each ROI was extracted from the original frame, resized to 256×256 pixels, and converted to grayscale to normalize appearance across subjects and lighting conditions.

During data loading, values were normalized to the range [0, 1] by dividing by 255. When the binocular variant was required, images from both eyes were stacked into two channels within a single tensor, preserving the base CNN architecture and enriching the available visual signal. Dataset partitions were made by participant (train/val/test) to prevent information leakage between sets and to evaluate inter-individual generalization capability.

2.3. Training

Training was implemented in TensorFlow/Keras under two equivalent conditions: a monocular model (SET) and a binocular model (DET). Both variants share the same base architecture; the only difference is in the input form: a single eye image (1 channel) or the combination of both eyes stacked as two channels (2 channels). In each case, the network receives grayscale images of 256×256 pixels and outputs two values representing the gaze coordinates on the screen. The decision to stack the eye images in the binocular case preserves the model topology and ensures a fair comparison across conditions, so that any observed improvement can be attributed solely to the additional information from the binocular channel. For model training and evaluation, the augmented dataset was split into 70% for training, 15% for validation, and 15% for testing, corresponding to approximately 336,000 images for training, 72,000 for validation, and 72,000 for testing. This partition remained fixed throughout all experiments to ensure experimental consistency and reproducibility of the reported results.

2.3.1. Network Architecture

The CNN model (Figure 5) consists of three convolutional blocks (Conv–ReLU–MaxPooling) followed by four dense layers and a linear output layer. Each convolutional block uses 3×3 filters with 32, 64, and 128 kernels, respectively, followed by 2×2 max pooling operations that reduce the spatial dimensionality and promote local invariance. Subsequently, the Flatten layer transforms the feature maps into a one-dimensional vector that feeds the dense layers of 512, 256, 128, and 64 neurons, all with ReLU activation. The final linear output layer contains two neurons that generate the predicted gaze coordinates (x, y). During training, the Adam optimizer was used with a learning rate of 0.001, a batch size of 32 samples, and a maximum of 50 epochs. The chosen loss function was mean squared error (MSE), suitable for continuous regression tasks.

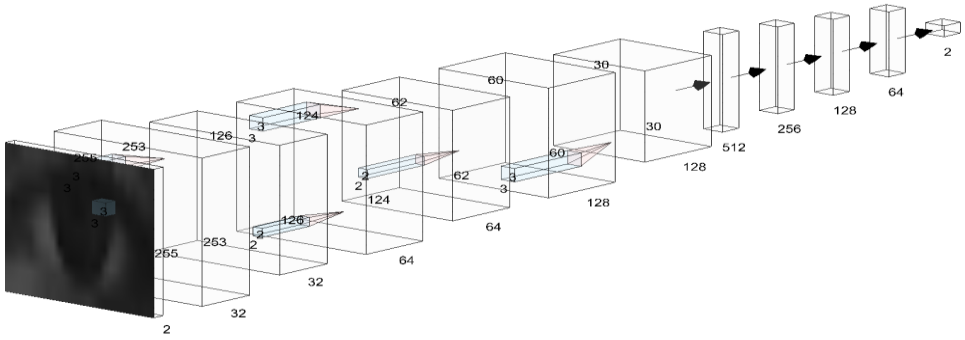


Figure 5 Architecture of the proposed CNN.

Before each iteration, the eye images are normalized by dividing each pixel value by 255 to keep them in the [0, 1] range. Additionally, data augmentation (slight rotations, lighting variations, blurring, and partial occlusion) was applied to improve the model's generalization to real-world conditions. The hyperparameters were kept consistent to ensure the comparison between the two models was conducted under the same conditions (Table 1).

Table 1 Hyperparameters of the training for both models.

Hyperparameters	SET	DET
Image_height	256px	256px
Image_width	256px	256px
Num_channels	1 channel	2 channels
Epoch	50 epoch	50 epoch
Batch_size	32	32
Optimizer	Adam ((η=0.001)	Adam ((η=0.001)
Loss function	MSE	MSE

2.4. Implementation of data cleaning modules

During eye tracking, various faults and inconsistencies were identified in the base model's processing flow. In particular, the gaze point exhibited abrupt movements, unexpected jumps, and a trajectory that was difficult to interpret. Even when the user kept their gaze fixed on a point, the model failed to remain stable: its behavior appeared random and exhibited variations that did not correspond to the actual eye movements. To mitigate these issues, a Kalman filter (Figure 6, block E) was implemented, which helped smooth the gaze point trajectory and reduce the influence of outliers. This filter adjusts two key parameters: the process variance, which controls how much the gaze can change from one frame to the next, and the measurement variance, which determines the level of confidence placed in the model's prediction. By tuning these values, a more fluid, coherent visualization was achieved that is consistent with the natural movement of the human eye and maintains stability even during prolonged fixations. In addition to the Kalman filter, a fundamental element of professional eye-tracking systems was incorporated: prior calibration. This phase is crucial because it allows the model to adapt to conditions different from those during training, compensating for variations in lighting, distance, head orientation, and individual user characteristics. To do this, a 16-point calibration was designed, along with a robust clustering algorithm, DBSCAN (Figure 7, block E).

This method takes two essential parameters: eps, which defines the maximum distance between points to consider them part of the same group, and min_samples, which indicates the minimum number of samples needed to form a valid cluster. Based on these parameters, DBSCAN groups the coordinates obtained at each calibration point and generates a representative centroid, automatically discarding outliers or samples affected by noise. This technique significantly improved the system's stability, reducing errors under conditions such as low lighting, reflections, slight involuntary movements, or variations in the user's position. Overall, integrating the Kalman filter with DBSCAN in the calibration process greatly strengthened the system's robustness, yielding a more accurate, smoother, and more resilient gaze point under real-world, challenging conditions.

An additional distinctive aspect of the proposed approach is the explicit incorporation of a data-cleaning and stabilization module following the initial gaze estimation. While most related work focuses on model design or learning architectures, this study introduces a methodological approach that combines clustering techniques with Kalman filtering to refine and stabilize temporal predictions. This strategy improves the coherence of the estimated trajectory and reduces the impact of frame-by-frame noise, representing a significant methodological difference compared to conventional eye-tracking approaches.

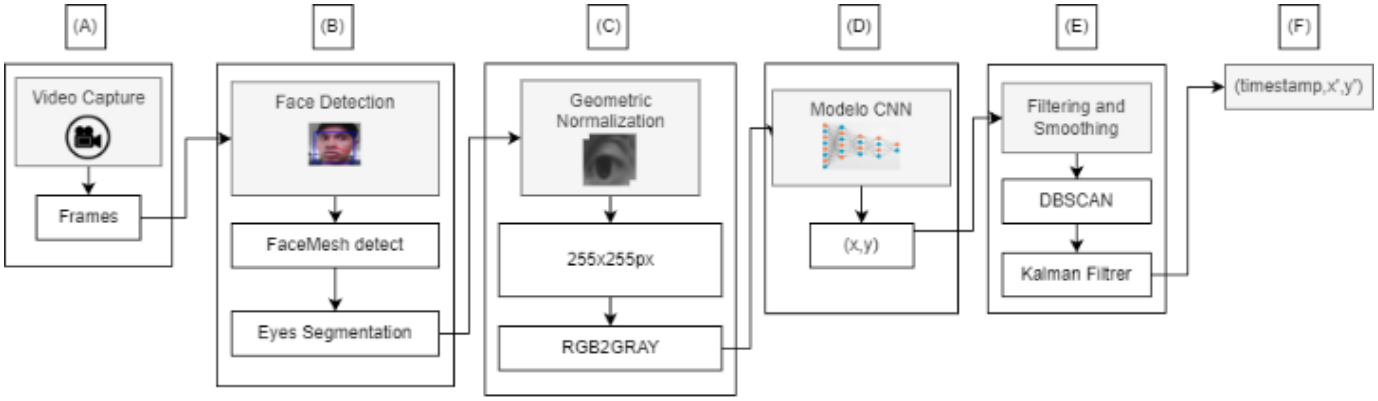


Figure 6 Processing flow with a new block

2.5. Description of the experiments for the evaluation of the models

To quantify the performance of the monocular and binocular variants under identical capture and preprocessing conditions (TensorFlow/Keras), we defined two complementary protocols (Novák, J. et al., 2024). The first, static, aims to uniformly sample the screen plane to estimate point accuracy; the second, dynamic, simulates a continuous tracking scenario to evaluate temporal stability. In both cases, the stimulus generator is synchronized with the frame acquisition loop so that each eye image is associated with the exact screen target coordinate. The main metrics are the mean Euclidean distance (MED) in pixels and the angular error in degrees; in the dynamic protocol, we also include measures of trajectory stability (e.g., RMSE along the path and the percentage of samples within a tolerance radius).

2.5.1. Experiment 1: Calibration at 9 points (static)

The test involved presenting nine fixed points evenly distributed on a 3×3 grid on the screen (see figure 6). Each user fixates their gaze on the active point for three seconds; at the end of the interval, the system automatically advances to the next stimulus. This procedure generates a dense, well-distributed matrix of pairs (eyes, coordinates) suitable for evaluating spatial accuracy across different regions of the visual plane. At 60 fps, approximately 300 frames are captured per point, for a total of about 2,700 per session. Using this data, the MED and the angular error per point are calculated, along with the overall average. Additionally, the dispersion (standard deviation) and error heat maps are reported to identify areas where the model tends to degrade (for example, corners or edges).

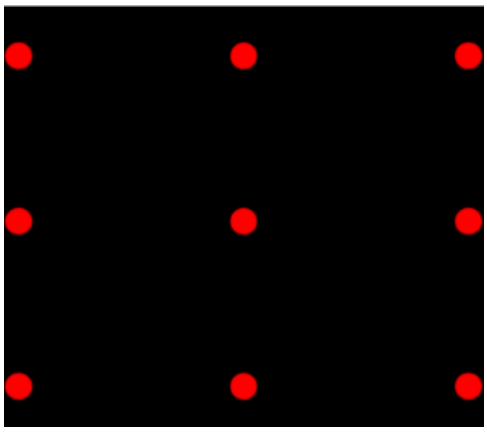


Figure 7. 9-point calibration experiment.

2.5.2. Experiment 2: Blue Ball

The second experiment consisted of a dynamic sequence in which a blue sphere moved at a constant speed, traversing the entire perimeter of the screen clockwise. The movement begins in the top-left corner and sequentially advances to the other corners (Figure 7). This test was designed to simulate a situation closer to the natural behavior of the user's gaze, in which it moves smoothly rather than fixating on the fixed points of the static 9-point calibration. The main goal was to evaluate the stability of eye tracking and the models' ability to maintain accurate estimates during continuous, prolonged eye movements.

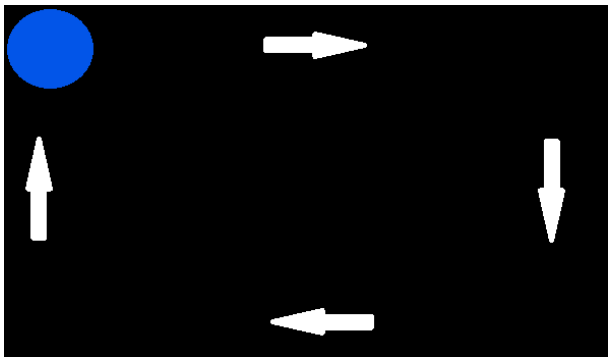


Figure 8 Blue Ball Experiment.

2.6. Application of the experiments

The two experiments were conducted under the same capture conditions used to build the dataset, ensuring strict comparability between training and evaluation. The sample consisted of 10 participants (men and women selected at random), and all sessions took place in a room with favorable lighting, with the screen positioned 60 cm from the participant's eyes (Figure 8). The system used a consumer webcam and the same preprocessing stack already described (face detection, eye ROI extraction, grayscale conversion, and normalization), while maintaining the sampling frequencies used during data collection (60 fps as the target setting; 30 fps as the minimum acceptable for modest equipment). These conditions replicate those of the data collection stage, in which screen positions were stimulated using a grid and controlled timing at each point, ensuring a uniform flow of pairs (eyes, coordinates) per participant.

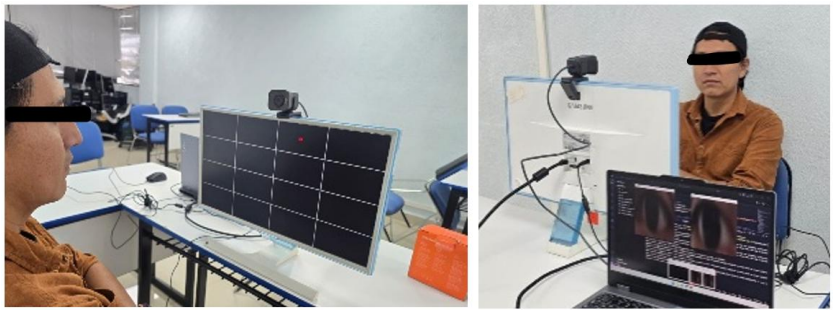


Figure 9 Participant 12 during an experiment.

Both experiments used the same eye ROI pipeline (256×256, grayscale, normalized), synchronized frame by frame with the stimulus coordinate, and the same quality criteria as those used during dataset creation. This ensures that the results reflect the effect of the input type (one eye vs. both eyes) on spatial accuracy and movement robustness exclusively, without introducing biases from differences in capture or environment.

3. Results And/Or Discussion

According to the structure defined in the base document, the results were obtained from 10 participants under the same capture conditions as those used for the dataset (favorable lighting and a 60 cm distance). In general, the binocular model (DET) consistently outperforms the monocular model (SET). Higher classification accuracy (Accuracy, Precision, Recall, F1) and lower regression errors (MAE, Median, RMSE) are observed. In practice: more precise gaze point estimations and more stable trajectories, with particular robustness against asymmetric lighting and partial occlusions.

3.1. Experiment 1 – 9-point calibration

The indicators show that DET achieves a more uniform error distribution across the grid, with clear reductions in MAE, MED, and RMSE compared with SET (see Table 2). The increases in Accuracy/Precision/Recall/F1 indicate that the system more accurately locates gazes within the expected regions (AOIs) and produces fewer false positives/negatives. The improvement is especially noticeable in corners and edges, areas where monocular focus tends to degrade due to changes in pose or lighting. In summary, with fixed stimuli, DET offers higher point accuracy and less dispersion.

Table 2 Results of experiment 1 with both models.

9-point Calibration		
Metrics	SET	DET
Accuracy	76%	82%
Precision	78%	85%
Recall	74%	80%
F1-Score	76%	82%
MAE	121px	91px
MED	161px	120px
RMSE	194px	145px

The nine-point calibration shows that the binocular model (DET) outperforms the monocular model (SET) in all evaluated metrics (Figure 9). The DET model achieves 82% accuracy, 85% precision, 80% recall, and 82% F1-score, compared with 76%, 78%, 74%, and 76% for SET.

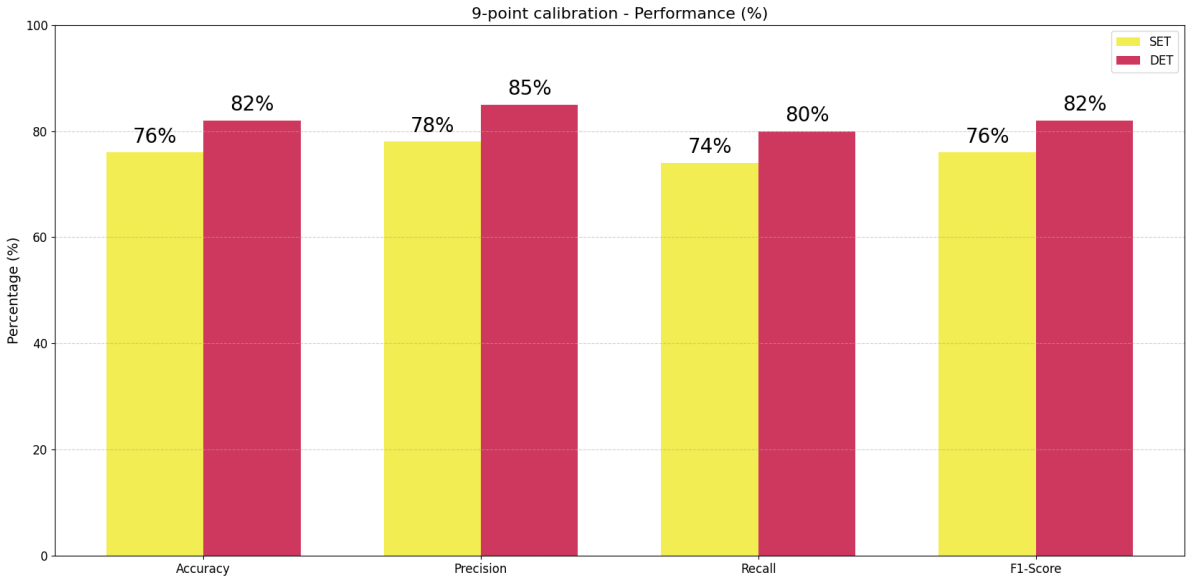


Figure 9. Graphical performance of the classification metrics (Calibration at 9 points).

In the 9-point calibration, the binocular model (DET) (Figure 10) clearly reduces errors compared to the monocular (SET).

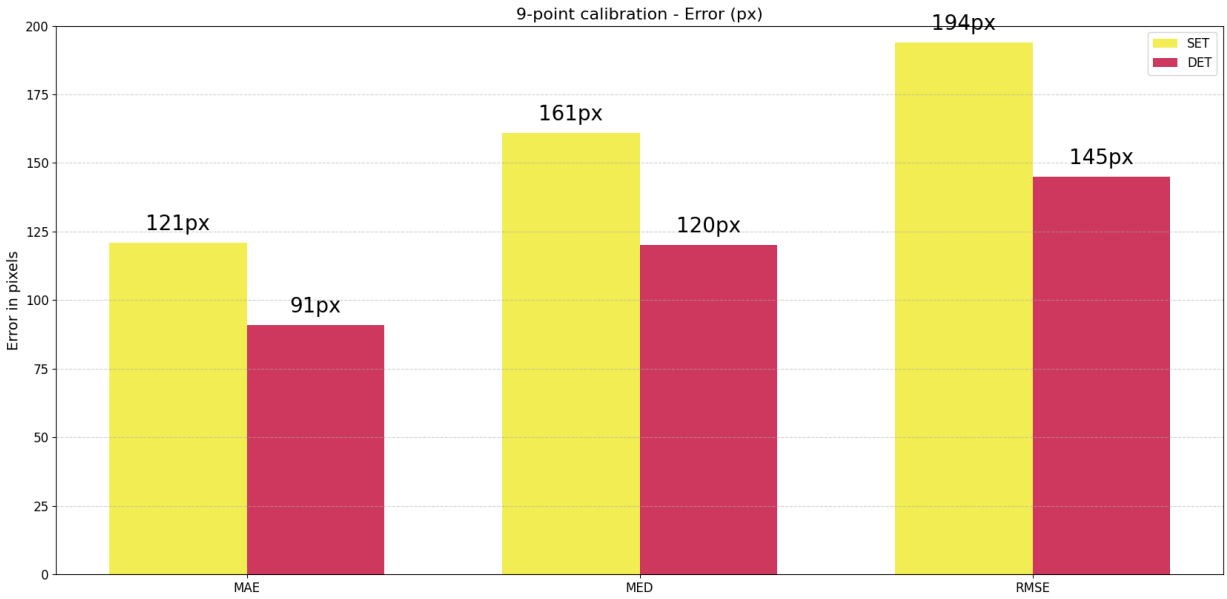


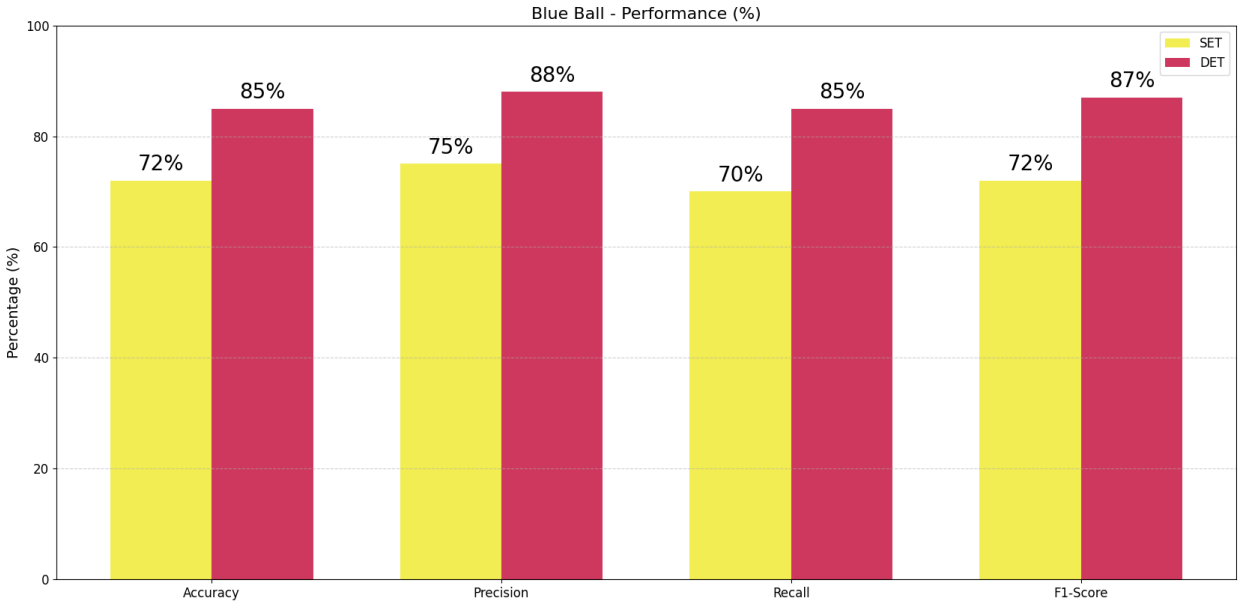
Figure 10. Graphical performance of the regression metrics (Calibration at 9 points).

3.2. Experiment 2 – Blue Ball

In this experiment, the continuous sequence, DET, exhibits smoother trajectories and less drift, as reflected by a lower RMSE along the path and a higher proportion of samples within the tolerance threshold around the moving stimulus. Improvements in Accuracy/Precision/Recall/F1 (see Table 3) confirm a better 'lock' on the target during movement, with fewer error peaks and faster recovery from micro-occlusions or light variations. With moving stimuli, the binocular advantage is accentuated, demonstrating greater temporal stability and operational reliability for continuous use.

Blue Ball		
Metrics	SET	DET
Accuracy	72%	85%
Precision	75%	88%
Recall	70%	85%
F1-Score	72%	87%
MAE	135px	72px
MED	176px	94px
RMSE	258px	138px

In both tests, DET consistently outperforms SET: it increases accuracy and reduces pixel errors in both static and dynamic scenarios, reinforcing the hypothesis that combining both eyes provides complementary signals that yield greater system accuracy and robustness (Figure 11).
In the BlueBall dynamic experiment, the binocular model (DET) significantly reduces errors compared to the monocular (SET).



Figue. 11. Graphical performance of the classification metrics (Blue ball).

MAE: from 135 px to 72 px; MED: from 176 px to 94 px; and RMSE: from 258 px to 138 px; cuts of approximately 46–47%. This results in smoother trajectories, better target locking, and less drift during continuous movement (Figure 12).

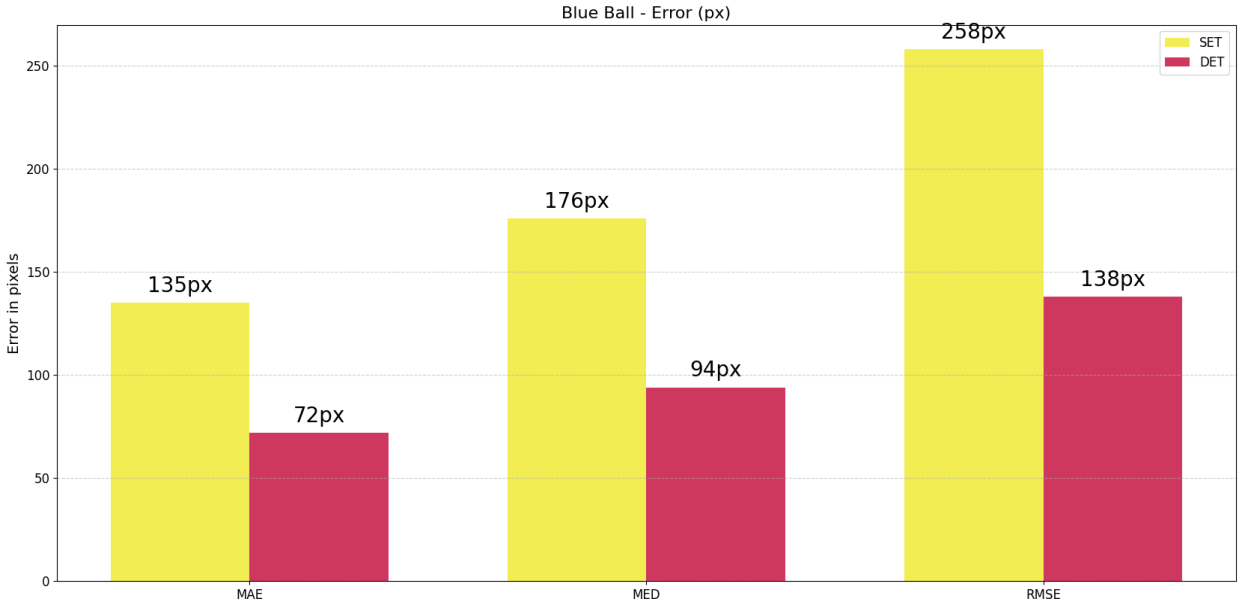


Figure. 12. Graphical performance of the regression metrics (Blue Ball).

3.3. Limitations and considerations regarding generalization

Although the experimental evaluation was conducted with a relatively small cohort (N = 10) and under moderately controlled conditions, the results provide a valid proof of concept within the considered experimental scope. In this context, the proposed model demonstrates robust performance under general usage conditions by incorporating multiple filters that precisely delineate the ocular area, reducing the influence of irrelevant regions or noise in the facial image.

This approach favors that the inference process specifically focuses on the eye region, mitigating the impact of external variations such as changes in lighting or the presence of corrective lenses, which did not significantly alter the system's performance in the conducted experiments. However, limitations arise in particular scenarios, such as for users with certain visual impairments or atypical oculomotor conditions, such as strabismus, where ocular alignment can affect the stability of reference points and, consequently, gaze estimation.

Additionally, the current experimental scale limits the direct extrapolation of results to broader use contexts. In real-world applications, inter-user diversity—including variations in facial morphology, blinking patterns, lens use, and gaze behavior—may introduce domain shifts relative to the acquisition conditions evaluated. Therefore, as future work, it is proposed to expand participant diversity and capture scenarios that incorporate multiple lighting configurations, viewing distances, and head poses, and to conduct stratified analyses to more accurately quantify the system's generalization capacity.

3.3. Comparisons with Public Datasets

Many public eye-tracking datasets are designed under different assumptions, such as specific image resolutions, fixed camera setups, different calibration schemes, or capture protocols that do not explicitly separate monocular and binocular inputs per eye. Additionally, several of these datasets do not provide high-frequency temporal sequences or detailed control over the acquisition at the calibration point, which limits their direct compatibility with the proposed approach.

In this regard, the main goal of this study is not to establish an absolute performance comparison against existing benchmarks, but rather to analyze in a controlled manner the relative impact of using monocular and binocular information within the same experimental pipeline. However, future extensions of this work could explore adaptation strategies or cross-evaluation with public datasets, provided that the consistency between acquisition protocols and the model's hypotheses is maintained.

4. Conclusions

This work demonstrates that it is possible to reduce costs and democratize gaze estimation using a lightweight convolutional neural network (CNN) and a webcam, without requiring an infrared (IR/PCCR) sensor or proprietary licenses. A complete and reproducible methodology was designed, including: a synchronized collection algorithm with visual stimuli, a participant-stratified dataset with precise extraction of eye regions of interest (ROI) of 256×256 pixels in grayscale, controlled data augmentations, and training in TensorFlow/Keras under the same hyperparameters for two topologically equivalent conditions: SET (monocular) and DET (binocular). Beyond the quantitative improvement, the practical relevance of this work lies in its ability to operate with everyday hardware—a simple webcam—without relying on IR optics or proprietary software. This democratizes access to eye tracking, enabling its use in education, accessibility, remote interaction, and low-budget environments.

The approach combines experimental rigor and low cost, offering a viable alternative for lightweight research and commercial applications that previously required specialized equipment. Furthermore, incorporating the Kalman filter and DBSCAN helped reduce outliers, improve calibration stability, and obtain more coherent trajectories, thereby reinforcing the system's reliability in real-world scenarios. These techniques, along with the ability to adjust their parameters, give the system the flexibility needed to adapt to different users and usage conditions. The results are conclusive: the binocular approach (DET) consistently outperforms the monocular (SET) in all metrics and protocols. In the static grid, the binocular model provides greater spatial accuracy and less dispersion, especially at the edges and corners; in the dynamic grid, it maintains a more stable tracking, with less drift and accumulated error. This technical advantage stems from both eyes providing complementary and redundant signals under adverse conditions (asymmetric lighting, micro-occlusions, or reflections): when one eye loses quality, the other compensates, and the network leverages geometric relationships (e.g., vergence) that are not available in the monocular case.

This positions the proposal as an accessible and robust alternative for human–computer interaction, accessibility, and real-time educational or industrial environments. For future work, expanding participant diversity and contexts, exploring minimal calibration strategies, and deploying the model on embedded platforms are planned, always maintaining the guiding principle of this research: maximum accuracy and stability with the lowest possible cost and complexity of use.

References

Aguayo. (2025). *Eye-tracking en la investigación UX*. https://aguayo.co/es/blog-aguayo-experiencia-usuario/eye-tracking-investigacion-ux/

Ansari, M. A., Kasprowski, P., & Obaidat, M. S. (2021). Gaze tracking using an unmodified web camera and a convolutional neural network. *Applied Sciences, 11*(19), 9068. https://doi.org/10.3390/app11199068

Boesch, G. (2023). *Deep residual networks (ResNet, ResNet-50): A complete guide*. Viso.ai. https://viso.ai/deep-learning/resnet-residual-neural-network/

Bonnin, R. (2017). *Machine learning for developers*. Packt Publishing.

C3 AI. (2020). *Root mean square error (RMSE)*. https://c3.ai/glossary/data-science/root-mean-square-error-rmse/

Daniel. (2021). *Convolutional neural network: Definición y funcionamiento*. DataScientest. https://datascientest.com/es/convolutional-neural-network-es

Dilmegani, C. (2025). *12+ data augmentation techniques for data-efficient ML*. AIMultiple. https://research.aimultiple.com/data-augmentation-techniques/

Engati. (2023). *Euclidean distance*. https://www.engati.com/glossary/euclidean-distance

Evdokimov, D. (2024). How scientists use webcams to track human gaze. *Frontiers for Young Minds, 12*. https://doi.org/10.3389/frym.2024.1259404

Falch, L. (2024). Webcam-based gaze estimation for computer screen interaction. *Frontiers in Robotics and AI, 11*, 1369566. https://doi.org/10.3389/frobt.2024.1369566

Jain, A. (2024). *A comprehensive guide to performance metrics in machine learning*. Medium. https://medium.com/@abhishekjainindore24/a-comprehensive-guide-to-performance-metrics-in-machine-learning-4ae5bd8208ce

Kristensen, S. (2022). *Eye tracking*. iMotions. https://imotions.com/eye-tracking/

López, I. (2023). *Nueva tecnología eye tracking*. Instituto Oftalmológico Recoletas. https://iorecoletas.com/eye-tracking/

Lubinus Badillo, J., et al. (2021). Redes neuronales convolucionales: Un modelo de deep learning en imágenes diagnósticas. Revisión de tema. *Revista Colombiana de Radiología, 32*(3), 5591–5599. https://doi.org/10.53903/01212095.161

Murel, J., & Kavlakoglu, E. (2025). *¿Qué es el aumento de datos?* IBM. https://www.ibm.com/mx-es/think/topics/data-augmentation

Novák, J., et al. (2024). Eye tracking, usability, and user experience: A systematic review. *International Journal of Human-Computer Interaction, 40*(17), 4484–4500. https://doi.org/10.1080/10447318.2023.2221600

Sauro, J. (2022). *Essential eye-tracking visualizations and metrics*. MeasuringU. https://measuringu.com/eye-tracking/

Vermeeren, A. P. O. S., Law, E. L.-C., Roto, V., Obrist, M., Hoonhout, J., & Väänänen-Vainio-Mattila, K. (2010). User experience evaluation methods: Current state and development needs. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction* (NordiCHI 2010). Association for Computing Machinery.

Vidhya, K., et al. (2025). Real-time gaze estimation using webcam-based CNN models for human–computer interactions. *Computers, 14*(2), 57. https://doi.org/10.3390/computers14020057