A Fast and Accessible Neural Network Based Eye-Tracking System for Real-Time Psychometric and HCI Applications

Emanuele Iacobelli[1], Davide Pelella[1], Valerio Ponzi[1,2], Samuele Russo[3] and Christian Napoli[1,2,4]

[1]Department of Computer, Automatic and Management Engineering, Sapienza University of Rome, 00185 Rome, Italy
[2]Institute for Systems Analysis and Computer Science, Italian National Research Council, 00185 Roma, Italy
[3]Department of Psychology, Sapienza University of Rome, 00185 Rome, Italy
[4]Department of Computational Intelligence, Czestochowa University of Technology, 42-201 Czestochowa, Poland

Abstract

Eye-tracking technology has long been a valuable tool across various domains, and recent advancements in neural networks have significantly expanded its versatility and potential. However, real-world applications continue to face challenges such as accommodating users' natural movements, variations in lighting, occlusions of the eyes, and the limited availability of large, open-source datasets for training models. To address these issues, we developed a comprehensive pipeline that produces a lightweight and efficient model, requiring only an RGB camera as external hardware, making it easily deployable on standard PCs. Key input features include facial images, eye regions, head pose angles, the Eye Aspect Ratio (EAR), and a face grid that determines the face's location within the camera's frame. The model was trained using a custom dataset, in which participants were instructed to fixate on both randomly positioned points and the standard 9-point grid commonly employed in eye-tracking calibration. The resulting system was integrated into a real-time application, offering fast and accessible gaze tracking, making it well-suited for studies requiring rapid gaze assessments across broad regions of the screen, such as psychometric research and Human-Computer Interaction (HCI) tasks. Its design is particularly advantageous for gaze laterality studies, which explore hemispheric dominance and attentional bias in cognitive and emotional processing, key concepts relevant to ADHD and dyslexia. Moreover, the system's capabilities naturally extend to emotional and decision-making tasks, where broad-area gaze tracking can support the analysis of preference formation and attentional patterns without the need for specialized hardware.

Keywords

Eye Tracking, Machine Learning, Real-Time Application, Appearance-Based Eye Tracking System, Gaze Laterality Studies

1. Introduction

The human senses gather approximately 11 million bits of information per second, with about 80% being visual and the remainder distributed among the other senses. Due to the dominance of visual perception, AI-based technology [1, 2, 3, 4, 5] has become a valuable research tool in fields such as psychology [6, 7, 8, 9], marketing [10, 11], healthcare [12, 13, 14, 15], safety [16, 17], Human-Computer Interaction (HCI) [18, 19, 20, 21], and Virtual Reality (VR) and robotics [22, 23, 24]. This technology is particularly crucial in psychometric applications, facilitating studies on cognitive functions like focus, emotion recognition, and decision-making, as well as in gaze laterality research, where phenomena such as hemispheric dominance and attentional bias are investigated. Historically, professional systems relied on expensive hardware, such as scleral search coils [25], electrooculography [26], EEG [27, 28], and infrared cameras [29], limiting accessibility. However, advancements in computer vision and machine learning, particularly Convolutional Neural Networks (CNNs), have made eye-tracking technology more accessible, providing fast and reliable gaze tracking without the need for specialized hardware. Delving more into the details, several studies have highlighted the broad utility of these systems, especially in understanding gaze laterality and its implications for neurological conditions. For example, in [30], an eye-tracking system was used to analyze gaze fixation and variability in children with ADHD, successfully identifying differences in visual attention that distinguish ADHD patients from healthy controls. Similarly, [31] investigated reading performance in children with ADHD, providing key insights into how the condition affects oculomotor control and reading ability, highlighting its potential for educational and clinical applications. in [32] a similar approach is used for to diagnose autism spectrum disorder. In addition, these systems have proven effective in detecting dyslexia by capturing distinctive eye movement patterns during reading tasks. This approach, powered by CNNs, enables early identification of dyslexia, allowing for timely interventions [33].

To summarize, eye-tracking in gaze laterality research provides a unique window into cognitive processes, al-

SYSYEM 2024: 10th Scholar's Yearly Symposium of Technology, Engineering and Mathematics, Rome, December 2-5, 2024
iacobelli@diag.uniroma1.it (E. Iacobelli); ponzi@diag.uniroma1.it (V. Ponzi); samuele.russo@duniroma1.it (S. Russo); cnapoli@diag.uniroma1.it (C. Napoli)
0009-0003-1379-9106 (E. Iacobelli); 0009-0000-2910-0273 (V. Ponzi); 0000-0002-1846-9996 (S. Russo); 0000-0002-3336-5853 (C. Napoli)

lowing for a deeper understanding of how attentional resources are allocated across the visual field. For that reason, the motivation behind developing our lightweight real-time application is to enable more researchers to study gaze movement patterns without the need to invest in expensive professional eye-tracking systems. By reducing the cost and complexity, we aim to make this technology more available for a wider range of studies focused on cognitive and neurological research. As eye-tracking becomes more accessible, its application in both research and clinical environments will continue to grow, offering new avenues for understanding and addressing these conditions.

2. Related Works

2.1. Eye-Tracking Approaches

In literature is possible to distinguish among 2 possible approaches, free from heavy specific instrumentation: Model-Based, and Appearance Based [34].

2.1.1. Model-Based Approach

The model-based approach utilizes a 3D geometric model to determine the direction of the eye's gaze. This is done by calculating a vector that connects the 3D positions of the eyeball's center and the pupil's center. These positions are derived from 2D eye landmarks and the 2D position of the iris in the image, which are then projected onto the 3D model. Initially, research in this area focused on developing accurate geometric models, but more recent advancements have shifted towards improving the precision of eye landmark detection using machine learning methods [35, 36, 37, 38, 39, 40].

For example, [41] describes an eye-tracking system that uses the Kinect v2 sensor. This device, equipped with RGB and depth cameras, identifies facial landmarks and computes the 3D gaze vector by combining face orientation with eye direction. Another system, presented in [42], employs the Supervised Descent Method (SDM) to detect 2D facial landmarks, while depth information from the Kinect is used to estimate the user's 3D head pose. The eye regions are further processed using the Starburst algorithm to estimate the pupil center for accurate gaze tracking.

A more recent approach [43] uses a combination of Unet and Squeezenet networks to significantly improve the accuracy and memory efficiency of eye-gaze tracking, making it feasible even on smartphones. Although model-based techniques offer the advantage of being training-free and adaptable to various conditions, they can still face challenges with the precision of landmark detection and the accurate positioning of the iris.

2.1.2. Appearance-Based Approach

Appearance-based methods aim to learn a direct mapping between the input image and the eye-gaze direction without relying on camera calibration or geometric models [44]. These methods are highly flexible, but they can be sensitive to head movements. Currently, the most effective approaches leverage convolutional neural networks (CNNs) and their variants to create mapping functions. While CNNs often achieve high accuracy on benchmark datasets, they can struggle to generalize across different datasets unless trained on large-scale annotated datasets, which are time-consuming and complex to create.

Recent works have made significant efforts to overcome these challenges by creating diverse and comprehensive datasets that improve the training and generalization of CNN models. For example, the MPIIGaze dataset [45] is a widely-used resource that contains over 200,000 images of 15 participants captured in real-world environments. This dataset helps improve gaze prediction in unconstrained settings, with variations in lighting, head pose, and other real-world factors.

Similarly, ETH-XGaze [46] provides a large dataset with high-quality annotations, including images from 110 subjects captured under a wide range of head poses and lighting conditions. This dataset addresses the limitations of smaller datasets and enables CNN models to learn robust gaze estimations in diverse environments.

Additionally, the FAZE dataset [47] is designed specifically to tackle domain generalization problems. FAZE includes a large number of participants and images across different devices and environments, aiming to enhance the generalization of appearance-based gaze estimation models by incorporating domain adaptation techniques.

For instance, [48] introduced GazeCapture, a dataset of videos recorded using smartphone front cameras under varying lighting conditions and head movements. They used this dataset to train a CNN to predict the screen coordinates a user is looking at on a smartphone or tablet. The input to the CNN includes segmented images of the eyes and face, as well as a mask showing the face's location in the image. To enhance real-time performance (10–15 FPS on modern mobile devices), the authors applied a technique called dark knowledge to reduce model complexity.

An alternative approach, proposed by [49], works in a desktop environment and uses an RGB camera to track eye movement. The system first segments the eye region, detects the iris center and the inner eye corner and then calculates an eye vector representing the eye's movement. A second-order polynomial mapping function, combined with head pose information, is used to map this eye vector to screen coordinates while compensating for head movements.

More recent work [50] shifts the focus from traditional

eye-gaze tracking to time-varying signals such as the vertical displacement between the iris and the inner eye corner, which is less affected by head movements. Instead of a direct mapping function, this method uses a CNN to track multiple eye feature points, including the iris center and eyelid positions. These points are then used to generate eye movement signals, which are fed into a specialized CNN for user behavior recognition.

2.2. Challenges and Approach

Despite notable advancements, real-world applications of eye-tracking technologies continue to face significant challenges. These challenges arise from environmental factors such as varying lighting conditions, reflections in the images (e.g., glare), objects on the face (e.g., eyeglasses), differences in contrast between the iris and pupil due to varying iris colors, and individual variations in eye anatomy. Additionally, the required computational resources, combined with the limited range of vertical eye movements, further complicate these implementations. Furthermore, the end-to-end approach relies on access to large-scale, publicly available datasets for training, which presents an additional hurdle. As a result, despite their potential, these methods have not yet been widely adopted, often being overshadowed by specialized eye-tracking equipment designed for specific purposes.

To address these challenges, a comprehensive pipeline has been developed, encompassing dataset collection, model architecture design, and real-time testing. The goal is to utilize Convolutional Neural Networks to create an end-to-end gaze prediction system that uses only images captured from a standard laptop webcam, aiming to achieve real-time performance.

3. Implementation

This section explores the implementation of the entire pipeline, from data collection to the architecture and real-time tracking, expanding the key components.

3.1. Dataset Collection

To develop a robust eye-gaze tracking system using just a portable computer's webcam, the dataset is crucial. The actual available ones present many limitations, such as the poor amount of data, poor quality data, or less liberty in the disposition and interaction of the user with the screen and the distance with the camera. Others, with higher volume data, are based on mobile devices, not allowing an easy transition from vertical screens of smartphones to horizontal PC screens, similarly, the proximity and the relative angle of interaction to the device itself are drastically different. To overcome these limitations, a brand-new collection of the dataset was necessary, which was more suitable for the task of interest. To address these challenges a system was designed to record the user's gaze on a PC's screen optimizing the data to the task of interest.

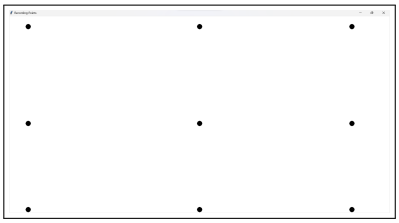

Figure 1: Recording points are used to collect the dataset for eye-gaze tracking. The recording pipeline sequentially displays each point on the laptop screen for a specified duration, while the webcam records the user's response. Each point corresponds to a key coordinate, capturing the user's gaze at nine meaningful cardinal points.

3.1.1. Recording

Data collection used 15-inch laptops in various environmental and lighting conditions. To mitigate the potential biases introduced by the use of a single webcam for all the data, multiple webcams from different computers were utilized. This strategy ensured the collection of a diverse set of images, simulating possible real-world applications and enhancing the robustness and generalization capability of the model limiting the bias introduction.

The custom dataset was gathered using specially developed software designed to display nine strategically chosen key points on the screen. These points included one at each corner of the screen, one at the center, and one at each of the four cardinal directions on the screen, Nord, sud, est, and west, as illustrated in Figure 1. Participants were instructed to fixate on each point sequentially, as they were shown, for a predetermined amount of time. This method allowed the collection of data samples for each gaze point while permitting participants to naturally adjust their head orientation and position like in a typical user interaction. Besides these 9 points, a variable number of random points were also shown on the screen, one after the other.

Additionally, the data collection process included sessions where participants were asked to wear glasses, to enrich the dataset with varied and challenging conditions.

3.2. Data extraction and Annotation

3.2.1. Face, mask grid and eyes

Each video is then processed extracting candidate frames. Each frame is inspected and the cropped face image is extracted if available. Face detection is executed using MediaPipe Face Detection, a lightweight model based on the BlazeFace architecture, which provides state-of-the-art techniques optimized for real-time applications. This model also performs well under challenging conditions such as partial occlusions, diverse facial orientations, and varying lighting conditions. The MediaPipe detector outputs the coordinates (x, y, w, h) of the bounding box around the detected face, which will be used to generate the face grid. This grid will provide a spatial map of face positioning within the video frame, helping the model to understand where the face is positioned relative to the entire frame. For each detected face, the bounding box coordinates (x, y, w, h) are scaled down to fit a grid of size 25×25. The bounding box is then mapped to this grid, marking cells where the face is 1 and all other cells as 0. This binary grid serves as one of the inputs to the model, facilitating the learning of spatial relationships in the gaze estimation tasks. The pipeline proceeds to the detection of the eyes, which employs either Haar cascades or the lib library, depending on which method yields the most accurate results on the specific conditions, as determined through a human-in-the-loop evaluation. While the Haar Cascades already provide a bounding box to crop the region of interest of the eyes, the dlib uses the landmark features of the eyes, considers padding around, and then crops. The eyes are not automatically included in the dataset, instead, each pair is inspected to ensure they are successfully recognized and sufficiently open. This check is crucial for confirming the quality of the data and that at least the horizontal position of the pupil can be discerned, excluding instances where the eyes are fully closed.

The face grid together with the face and eye images are grouped with the gaze point as in Figure 3 and then expanded with the additional input features.

3.2.2. Eye Aspect Ratio

If both eyes are correctly detected, the pipeline proceeds to associate the corresponding Eyes Aspect Ratio. The EAR is a geometric measure used to quantify the openness of the eyes. It is computed for each eye using six specific facial landmarks. For the left eye, the EAR is calculated as follows:

$$\mathrm{EAR}_{right} = \frac{\|P_{38} - P_{42}\| + \|P_{39} - P_{41}\|}{2\|P_{37} - P_{40}\|}$$

$$\mathrm{EAR}_{left} = \frac{\|P_{44} - P_{48}\| + \|P_{45} - P_{47}\|}{2\|P_{43} - P_{46}\|}$$

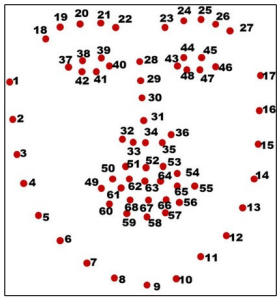

Figure 2: Facial landmarks provided by Dlib's 68 model, which detect the face and then the coordinate (x,y) of the 68 total features, providing information about the aperture of mouth, eye, and the orientation of the head. The points from 37 to 41 and from 43 to 48 will be leveraged for the computation of the Eye Aspect Ratio. Points 37 and 46 are leveraged for the roll pose, points 28, and 9 for the tilt, and the 34, 37, and 46 for the yaw.

Where $P_{37}, P_{38}, \ldots, P_{48}$ are the landmarks around the left and right eyes, respectively, according to the Figure 2. This metric facilitates the identification of eyelid position and blinks and leverages this information to improve the robustness of the model.

3.2.3. Roll-Pitch-Yaw

The head orientation is derived from facial landmarks detected in each frame. Roll is determined by the tilt of the line connecting the outer corners of the eyes (landmarks 37 and 46) relative to the horizontal axis, indicating left or right head tilt. The pitch measures the vertical tilt of the head and is calculated from the vertical position of the top of the nose bridge (landmark 28) relative to the chin (landmark 9), showing whether the head is tilted upward or downward. Yaw, indicating left or right head rotation around the vertical axis, is calculated from the position of the nose tip (landmark 34) relative to the midpoint between the eyes (average of landmarks 37 and 46). These three angles provide a comprehensive 3D orientation of the head, enhancing the accuracy of gaze estimation without necessitating a 3D head model or extensive computations, making the system adaptable for real-time applications where the user's head position varies.

In the end, this information is paired with the corresponding gaze point on the screen, selected from nine possible options.

3.3. Training Preprocessing

Some preprocessing steps were performed before feeding the data into the model for training to ensure the reliability and robustness of the system.

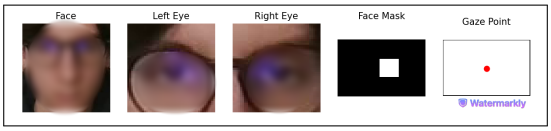

Figure 3: Sample of the dataset: The first element represents the cropped face from the frame. The bounding box coordinates of the head are used to create the Face Mask grid by dividing the entire frame into a grid and labeling the points occupied by the head. The second and third elements represent the eyes cropped from the face. All this information is linked to the gaze position on the screen, measured in pixels. The images are blurred for privacy reasons.

3.3.1. Image Resizing and Cropping

All images, face and eye regions, were resized to a uniform dimension of 64 × 64 pixels to maintain consistency across the dataset and to be fed into the model.

3.3.2. Histogram Equalization

Histogram equalization was employed to improve feature extraction. This technique adjusts pixel values in an image to enhance overall contrast. By redistributing the intensity levels, it equalizes the histogram of the output image. This process makes the model more robust in identifying relevant features under varied lighting conditions.

3.3.3. Data Augmentation

Several data augmentation techniques were applied to enhance the robustness of the model. Specifically, a random crop was used to simulate limited visibility of the face or eyes, and Gaussian Blur was employed to mimic poor image quality or focus. Variability in brightness and saturation was introduced, along with random rotations and random erasing of portions and filling it with random values. These techniques help reduce overfitting and improve the model's ability to generalize from the training data to unseen data in real-world applications.

These preprocessing steps, collectively, ensure that the data fed into the model is of high quality, consistent in size and format, and varied enough to promote robust learning and prediction accuracy.

3.4. Model

In this section will be presented the model, the architecture, and the training. The object was the realization of an efficient model able to provide good performances and run in real-time on a real-world application. The core of the implementation involved developing and training a convolutional neural network (CNN) to predict the gaze point based on the processed input features.

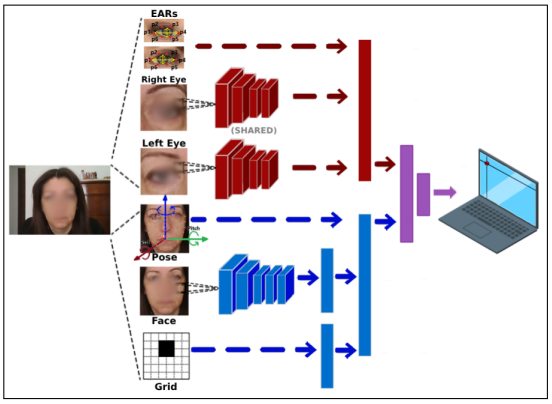

Figure 4: Model Architecture pipeline: The model is organized in 2 parallel pipelines that work on the eye and face. The first, in red, takes as input the cropped eye images and the Eye Aspect Ratio computed with the facial landmarks. The CNNs that take as input the eyes share the parameters. The second, in blue, takes as input the cropped face, the Mask grid, and the head pose. Then the outputs are concatenated to compute the gazepoint. The images are blurred for privacy reasons.

3.4.1. Model Architecture

The model architecture draws inspiration from the iTracker model [48], incorporating modifications to enhance performance. These modifications include additional input features such as head pose angles (yaw, pitch, and roll), the Eye Aspect Ratio (EAR), and the reorganization and reduction of the layers, to provide a lighter model with faster convergence. The complete pipeline is shown in Figure 4.

The Eye Aspect Ratio incorporation started from the consideration that, in normal conditions, users will tend to open their eyes wider when looking at higher points on a screen and as narrow as they are looking downward on the screen. The integration of the EAR information aims to specifically enhance the sensitivity of the model to vertical gaze shifts, improving the performance of the model on the vertical axe prediction and better handling cases in which the pupil is hardly observable by the simple raw image provided by the webcam.

The integration of head orientation data, along with the face grid, aims to provide the model with comprehensive information about the head's spatial positioning, without the necessity for computationally demanding external 3D models of the head or the eyes. Leverages the advantages of model-based methods while avoiding their drawbacks.

The model's architecture is organized in two distinct semantic pathways for the eyes and face, each consisting of several convolutional layers followed by pooling layers, these layers are designed to capture fine-grained details

necessary for accurate gaze estimation. The eye pathway processes separately the eye images with convolutional layers with shared parameters between the right and left eye, then the information is integrated with the EAR of both eyes with a fully connected. The face pathway processes the entire face region through a similar series of convolutional layers, then combines this information with the face grid, and roll pitch yaw angles.

3.5. Loss Function

The choice of an appropriate loss function is extremely important for the effectiveness of model training. During development, two primary loss functions were evaluated: Mean Squared Error (MSE) and Huber Loss.

Huber Loss was used to mitigate the outlier sensitivity issue with MSE, and the large scale of pixel predictions. It combines the best properties of MSE and Mean Absolute Error (MAE), behaving like MSE for small errors and like MAE for large errors, reducing the influence of outliers on the model's training. The Huber Loss is defined as:

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta(|y - \hat{y}| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

Where δ is a threshold parameter that dictates the transition point between the squared loss and the absolute loss. This property makes Huber Loss particularly promising for this application, as it balances the need for robustness with the sensitivity to small errors, critical for the precise prediction of gaze points. As shown in 5, Huber Loss provided a significant improvement in model convergence and performance compared to MSE, leading to more stable training and reduced gradient accumulation

3.5.1. Regularization

To further increase the robustness of the architecture, were leveraged some regularization techniques. Together with the already cited data augmentation, working on the data, on the model side leveraged the dropout, with a hyperparameter tuning which led to a successful value of 0.2. The training loop then incorporated a learning rate scheduler together with an early stopping.

3.6. Real-Time Tracking

The implementation of the real-time tracking functionality represents an essential step for practical applications. The following section describes the system's setup, the operational flow, and the technologies employed.

3.6.1. System Setup

For the real-time application, the system uses standard laptop webcams, 1280 x 720p, to capture video frames

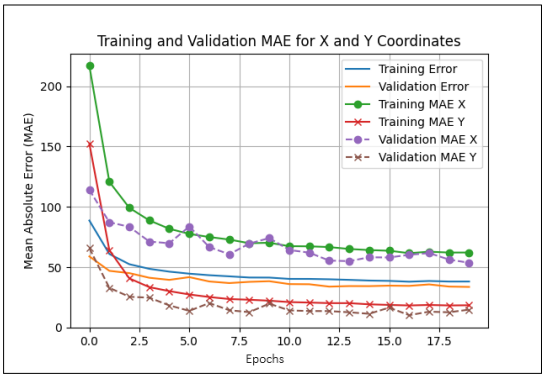

Figure 5: Model Training Plot: The mean absolute error (MAE) for pixel coordinates (x,y) is illustrated, with training data in blue and validation data in orange. The green and red lines represent the MAE for the x and y coordinates in the training data, while purple and brown depict these in the validation data. Notably, the MAE for the x coordinates is consistently higher than for the y coordinates, likely due to the larger pixel scale on the laptop screen.

of the user looking at the laptop screen 15-inch, maintaining consistency with the dataset. These frames are captured at a standard video frame rate of 30 frames per second, usually provided by commonly available webcams, which balances between providing smooth video and the computational load on the system. Each captured frame undergoes a series of preprocessing steps like in the training phase to maintain consistent data and to enhance the model's performance.

3.6.2. Calibration

The application allows to perform an optional calibration step to improve the results on the actual user of the eye-tracking. To perform the calibration, the system proceeds to show 9 points on the screen, in the 9 main representative points, for each collects the prediction provided by the model and compares it with the actual ground truth. Then leverage the difference between the two values to improve further predictions of the actual user.

3.6.3. Feature Extraction

The system offers flexibility in selecting the method for extracting eye patches from images, with options including the dlib68 and the eye cascade approaches. Following this it calculates the Eye Aspect Ratio and head angles. Then the model uses this information to predict the gaze point in screen coordinates in real-time. This step is computationally intensive and is optimized to run efficiently on standard consumer hardware without significant delays. The predicted gaze point is immediately displayed on the user's screen, providing real-time feedback.

4. Results

The proposed eye-tracking system demonstrates significant and promising improvements in gaze estimation compared to existing methods, excelling not only in prediction accuracy but also in efficiency—an essential factor for real-time applications. These gains are largely attributed to model optimizations that resulted in a more lightweight design. Specifically, the system performed smoothly on a laptop GPU, achieving a frame rate of 50 frames per second under optimal visibility and environmental conditions. On a laptop CPU, the model also maintained commendable performance, consistently delivering 30 frames per second without any drop in accuracy. This places the system on par with state-of-the-art models but with fewer parameters, making it more efficient.

To validate the model's effectiveness, a real-time application was developed. This application captures the live video feed from the webcam, processes it through the model to predict the user's gaze point, and then displays the predicted point on the screen, providing immediate feedback. To further assess performance, the model was compared with several state-of-the-art architectures on a common task, where the screen was divided into cells to track accuracy. The system showed promising results in terms of both inference time for real-time deployment and prediction accuracy, performing competitively against the benchmark models.

4.1. Comparison tasks

To evaluate the eye-tracking recordings and benchmark model performance, the real-time eye-tracking system was leveraged to perform a Fixation-Zone task [51]. To maintain consistency, all the experiments were performed on a laptop with an incorporated camera and a 15-inch screen. The approach performs a zone-wise classification accuracy, aggregated over the participants, where the users are instructed to fixate on specific regions of the screen, which turn green for a certain amount of time, free to move, as long as their gaze is constrained within the boundaries. The experiments instructed to perform a total of 3 tasks, where each aimed to enforce and study the model performances to specific behavior and compare this information with other SOTA architectures like the MPIIGaze, ETHXGaze, and the FAZE. In the first one, the screen was divided into 4 grid cells, determining the overall behavior of the model, and observing the general performances of eye-tracking all over the screen. The second task has 2 grid cells that split vertically the screen on two sides, this allows to better focus on the architecture capability to recognize the horizontal movement of the gaze. In the last task, which divided the screen into two grid cells horizontally, one above the other, the main focus was paid to the vertical movement recognition of the eyes, a critical aspect in the eye-tracking field.

The comparison of the model with the MPIIGaze, ETHXGaze, and FAZE pointed out a series of considerations about the performances of the models. The comparison focused on the zone classification accuracy in a grid setup of the screen (Figure 6),

4.1.1. Four Cell Grid task

Our model showed an overall accuracy of 88.5% with precision of 0.887 and recall of 0.885, excelling particularly in the top left grid cell while showing weaker performance in the bottom right grid cell. This shows a promising overall behavior, with room for improvement due to the unbalanced result in the four cells. Interestingly, the proposed architecture, compared to the other models, showed a slightly better understanding of the top left and top right cases, rather than the lower one.

4.1.2. Vertical Dual-Grid Task

The second task aimed to inspect the model's capability of recognizing horizontal movement, and the model demonstrated a good 93% overall accuracy, with 0.935 precision and 0.912 recall, quite struggling with the right section. This result comes from the previous considerations on the 4-grid task, in which, the bottom-right case was shown to be responsible for a drop-down in the prediction performances, making suffering this lack also to this other task when needed to correctly identify the gaze-point into the right part of the screen.

4.1.3. Horizontal Dual-Grid Task

The third task focused on evaluating the model's ability to identify vertical eye movement accurately. Here, the model achieved an accuracy of 91%, precision of 0.925, and recall of 0.899, in correctly recognizing the vertical grid cell observed. While it is apparent that other models experience a significant decline in performance transitioning from horizontal to vertical eye movement tasks, the proposed model exhibited only a slight drop in performance. It still significantly outperformed the other models, especially in the top cell case. Unfortunately, the bottom cell exhibited a slightly lower precision of the model when the gaze point approached the screen's center, leading to some misclassifications that slightly exceeded the bottom grid cell boundary and resulted in errors.

Unfortunately, many misclassification cases were also linked to unfavorable user visibility or environmental conditions. These factors made predictions more challenging for the model, highlighting areas for improvement and the potential to surpass existing architectures.

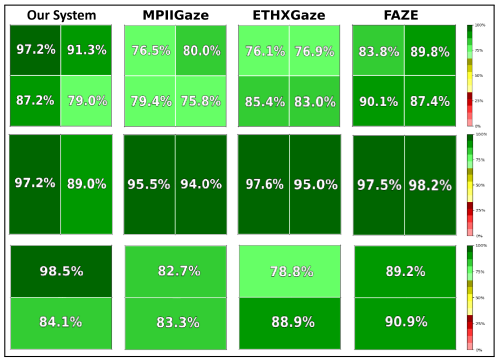

Figure 6: Comparison of gaze predictions of our network with state-of-the-art models MPIIGaze, ETHXGaze, and FAZE on zone classification accuracy. The first row represents the 4-cell task, where our system outperforms the other models in the Upper cells. The second row depicts the horizontal task, in which our model achieves good results, though it struggles with the right side, partly due to lower accuracy from the bottom right cell in the previous task. The third row illustrates the vertical task, showing that the developed model experiences a smaller performance drop compared to other models, although it still struggles with the bottom cell due to some imprecision near the center of the screen.

5. Conclusions

This work presented an end-to-end eye-tracking solution designed to be lightweight, utilizing only a standard webcam, while maintaining high accuracy and low resource requirements. The results indicate that the proposed system can be effectively applied in various real-world scenarios, achieving robust performance in both vertical and horizontal gaze detection. This versatility makes it a practical tool for studies in areas such as psychometrics and Human-Computer Interaction (HCI), especially those focused on gaze laterality and cognitive assessments for broad regions of the screen. Interestingly, the model demonstrated significant robustness in detecting vertical gaze movements, likely due to its high sensitivity to eye aperture ratio, making it particularly adept at distinguishing between upper and lower gaze positions. This capability was confirmed during task evaluations, where the system showed better precision in upper-screen positions compared to lower ones. Some imprecision was noted in central areas of the screen, particularly in distinguishing between center-up and center-low positions, likely due to the natural tendency for the eyes to be more open in upper gaze positions. Despite these challenges, the model maintained efficiency even on smaller laptop screens and at greater distances, contrasting with typical close-range setups required by mobile devices. Future work could focus on enhancing the system's robustness under diverse lighting conditions and user poses by enriching the dataset with more varied samples and a wider range of user demographics. Increasing the number of fixation points during data collection could also provide a more comprehensive understanding for the model, improving precision across all screen areas. Additionally, modifying the model to focus solely on the eye regions, rather than the entire face, could improve its performance in situations where face visibility is limited or when only one eye is visible. This refinement would not only make the model more efficient but also help it handle challenging conditions such as medical constraints or occlusions more effectively. In summary, the proposed system represents a significant advancement in making eye-tracking technology more accessible and practical for a wide range of everyday applications, reducing the need for expensive specialized hardware and offering a versatile tool for research and clinical environments.

References

[1] M. M. Mariani, R. Perez-Vega, J. Wirtz, Ai in marketing, consumer research and psychology: A systematic literature review and research agenda, Psychology & Marketing 39 (2022) 755–776.

[2] P. Banerjee, B. Sindhu, S. Sindhu, et al., Exploring the intersections of ai (artificial intelligence) in psychology and astrology: a conceptual inquiry for human well-being, J Psychol Clin Psychiatry 15 (2024) 75–77.

[3] C. Napoli, F. Bonanno, G. Capizzi, An hybrid neuro-wavelet approach for long-term prediction of solar wind, in: Proceedings of the International Astronomical Union, volume 6, 2010, p. 153 – 155. doi:10.1017/S174392131100679X.

[4] C. Napoli, G. Pappalardo, E. Tramontana, An agent-driven semantical identifier using radial basis neural networks and reinforcement learning, in: CEUR Workshop Proceedings, volume 1260, 2014. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84919742629&partnerID=40&md5=c3ee8a3fa1716b39215326edfc67d955.

[5] G. Capizzi, G. Lo Sciuto, C. Napoli, E. Tramontana, Advanced and adaptive dispatch for smart grids by means of predictive models, IEEE Transactions on Smart Grid 9 (2018) 6684 – 6691. doi:10.1109/TSG.2017.2718241.

[6] C. Clifton, F. Ferreira, J. M. Henderson, A. W. Inhoff, S. P. Liversedge, E. D. Reichle, E. R. Schotter, Eye movements in reading and information processing: Keith rayner's 40year legacy, Journal of Memory and Language 86 (2016) 1–19. URL: https://www.sciencedirect.com/science/article/pii/S0749596X15000960. doi:https://doi.org/10.1016/j.jml.2015.07.004.

[7] S. Russo, C. Napoli, A comprehensive solution for psychological treatment and therapeutic path planning based on knowledge base and expertise sharing, in: CEUR Workshop Proceedings, volume 2472, 2019, p. 41 – 47.

[8] G. Lo Sciuto, S. Russo, C. Napoli, A cloud-based flexible solution for psychometric tests validation, administration and evaluation, in: CEUR Workshop Proceedings, volume 2468, 2019, p. 16 – 21.

[9] S. Falciglia, F. Betello, S. Russo, C. Napoli, Learning visual stimulus-evoked eeg manifold for neural image classification, Neurocomputing 588 (2024). doi:10.1016/j.neucom.2024.127654.

[10] C. Napoli, G. Pappalardo, E. Tramontana, A hybrid neuro-wavelet predictor for qos control and stability, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8249 LNAI, 2013, p. 527 – 538. doi:10.1007/978-3-319-03524-6_45.

[11] G. De Magistris, S. Russo, P. Roma, J. T. Starczewski, C. Napoli, An explainable fake news detector based on named entity recognition and stance classification applied to covid-19, Information (Switzerland) 13 (2022). doi:10.3390/info13030137.

[12] P. S. Holzman, L. R. Proctor, D. L. Levy, N. J. Yasillo, H. Y. Meltzer, S. W. Hurt, Eye-tracking dysfunctions in schizophrenic patients and their relatives, Archives of general psychiatry 31 (1974) 143–151.

[13] S. I. Illari, S. Russo, R. Avanzato, C. Napoli, A cloud-oriented architecture for the remote assessment and follow-up of hospitalized patients, in: CEUR Workshop Proceedings, volume 2694, 2020, p. 29 – 35.

[14] C. Napoli, C. Napoli, V. Ponzi, A. Puglisi, S. Russo, I. E. Tibermacine, Exploiting robots as healthcare resources for epidemics management and support caregivers, in: CEUR Workshop Proceedings, volume 3686, 2024, p. 1 – 10.

[15] S. Russo, S. I. Illari, R. Avanzato, C. Napoli, Reducing the psychological burden of isolated oncological patients by means of decision trees, in: CEUR Workshop Proceedings, volume 2768, 2020, p. 46 – 53.

[16] H. Singh, J. S. Bhatia, J. Kaur, Eye tracking based driver fatigue monitoring and warning system, in: India International Conference on Power Electronics 2010 (IICPE2010), 2011, pp. 1–6. doi:10.1109/IICPE.2011.5728062.

[17] A. Alfarano, G. De Magistris, L. Mongelli, S. Russo, J. Starczewski, C. Napoli, A novel convmixer transformer based architecture for violent behavior detection, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 14126 LNAI, 2023, p. 3 – 16. doi:10.1007/978-3-031-42508-0_1.

[18] R. Jacob, K. Karn, Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises, volume 2, 2003, pp. 573–605. doi:10.1016/B978-044451020-4/50031-1.

[19] N. Brandizzi, S. Russo, G. Galati, C. Napoli, Addressing vehicle sharing through behavioral analysis: A solution to user clustering using recency-frequency-monetary and vehicle relocation based on neighborhood splits, Information (Switzerland) 13 (2022). doi:10.3390/info13110511.

[20] R. Brociek, G. D. Magistris, F. Cardia, F. Coppa, S. Russo, Contagion prevention of covid-19 by means of touch detection for retail stores, in: CEUR Workshop Proceedings, volume 3092, 2021, p. 89 – 94.

[21] N. Brandizzi, S. Russo, R. Brociek, A. Wajda, First studies to apply the theory of mind theory to green and smart mobility by using gaussian area clustering, in: CEUR Workshop Proceedings, volume 3118, 2021, p. 71 – 76.

[22] V. Ponzi, S. Russo, V. Bianco, C. Napoli, A. Wajda, Psychoeducative social robots for an healthier lifestyle using artificial intelligence: a case-study, in: CEUR Workshop Proceedings, volume 3118, 2021, p. 26 – 33.

[23] N. N. Dat, V. Ponzi, S. Russo, F. Vincelli, Supporting impaired people with a following robotic assistant by means of end-to-end visual target navigation and reinforcement learning approaches, in: CEUR Workshop Proceedings, volume 3118, 2021, p. 51 – 63.

[24] G. Capizzi, C. Napoli, S. Russo, M. Woźniak, Lessening stress and anxiety-related behaviors by means of ai-driven drones for aromatherapy, in: CEUR Workshop Proceedings, volume 2594, 2020, p. 7 – 12.

[25] E. Whitmire, L. Trutoiu, R. Cavin, D. Perek, B. Scally, J. Phillips, S. Patel, Eyecontact: scleral coil eye tracking for virtual reality, 2016, pp. 184–191. doi:10.1145/2971763.2971771.

[26] Y. Tian, J. Cao, Fatigue driving detection based on electrooculography: a review, EURASIP Journal on Image and Video Processing 2021 (2021) 33.

[27] S. Russo, I. E. Tibermacine, A. Tibermacine, D. Chebana, A. Nahili, J. Starczewscki, C. Napoli, Analyzing eeg patterns in young adults exposed to different acrophobia levels: a vr study, Frontiers in Human Neuroscience 18 (2024). doi:10.3389/fnhum.2024.1348154.

[28] N. Boutarfaia, S. Russo, A. Tibermacine, I. E. Tibermacine, Deep learning for eeg-based motor imagery classification: Towards enhanced human-machine

interaction and assistive robotics, in: CEUR Workshop Proceedings, volume 3695, 2023, p. 68 – 74.

[29] M. W. Johns, A. Tucker, R. Chapman, K. Crowley, N. Michael, Monitoring eye and eyelid movements by infrared reflectance oculography to measure drowsiness in drivers, Somnologie 11 (2007) 234–242.

[30] D. Y. Lee, Y. Shin, R. W. Park, S.-M. Cho, S. Han, C. Yoon, J. Choo, J. M. Shim, K. Kim, S.-W. Jeon, et al., Use of eye tracking to improve the identification of attention-deficit/hyperactivity disorder in children, Scientific Reports 13 (2023) 14469.

[31] S. Caldani, E. Acquaviva, A. Moscoso, H. Peyre, R. Delorme, M. P. Bucci, Reading performance in children with adhd: An eye-tracking study, Annals of Dyslexia 72 (2022) 552–565.

[32] V. Ponzi, S. Russo, A. Wajda, C. Napoli, A comparative study of machine learning approaches for autism detection in children from imaging data, in: CEUR Workshop Proceedings, volume 3398, 2022, p. 9 – 15.

[33] B. Nerušil, J. Polec, J. Škunda, J. Kačur, Eye tracking based dyslexia detection using a holistic approach, Scientific Reports 11 (2021) 15687.

[34] Q. Ji, D. Hansen, In the eye of the beholder: A survey of models for eyes and gaze, IEEE Transactions on Pattern Analysis & Machine Intelligence 32 (2010) 478–500. doi:10.1109/TPAMI.2009.30.

[35] F. Fiani, S. Russo, C. Napoli, An advanced solution based on machine learning for remote emdr therapy, Technologies 11 (2023). doi:10.3390/technologies11060172.

[36] E. Iacobelli, S. Russo, C. Napoli, A machine learning based real-time application for engagement detection, in: CEUR Workshop Proceedings, volume 3695, 2023, p. 75 – 84.

[37] F. Fiani, S. Russo, C. Napoli, A fully automatic visual attention estimation support system for a safer driving experience, in: CEUR Workshop Proceedings, volume 3695, 2023, p. 40 – 50.

[38] E. Iacobelli, V. Ponzi, S. Russo, C. Napoli, Eye-tracking system with low-end hardware: Development and evaluation, Information (Switzerland) 14 (2023). doi:10.3390/info14120644.

[39] S. Pepe, S. Tedeschi, N. Brandizzi, S. Russo, L. Iocchi, C. Napoli, Human attention assessment using a machine learning approach with gan-based data augmentation technique trained using a custom dataset, OBM Neurobiology 6 (2022). doi:10.21926/obm.neurobiol.2204139.

[40] F. Fiani, V. Ponzi, S. Russo, Keeping eyes on the road: Understanding driver attention and its role in safe driving, in: CEUR Workshop Proceedings, volume 3695, 2023, p. 85 – 95.

[41] B. C. Kim, D. Ko, U. Jang, H. Han, E. C. Lee, 3d gaze tracking by combining eye-and facial-gaze vectors, The Journal of Supercomputing 73 (2017) 3038–3052.

[42] X. Xiong, Z. Liu, Q. Cai, Z. Zhang, Eye gaze tracking using an rgbd camera: a comparison with a rgb solution, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, 2014, pp. 1113–1121.

[43] Z. Wang, J. Chai, S. Xia, Realtime and accurate 3d eye gaze capture with dcnn-based iris and pupil segmentation, IEEE transactions on visualization and computer graphics 27 (2019) 190–203.

[44] I. E. Tibermacine, A. Tibermacine, W. Guettala, C. Napoli, S. Russo, Enhancing sentiment analysis on seed-iv dataset with vision transformers: A comparative study, in: ACM International Conference Proceeding Series, 2023, p. 238 – 246. doi:10.1145/3638985.3639024.

[45] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Mpiigaze: Real-world dataset and deep appearance-based gaze estimation, IEEE transactions on pattern analysis and machine intelligence 41 (2017) 162–175.

[46] X. Zhang, S. Park, T. Beeler, D. Bradley, S. Tang, O. Hilliges, Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16, Springer, 2020, pp. 365–381.

[47] S. Park, S. D. Mello, P. Molchanov, U. Iqbal, O. Hilliges, J. Kautz, Few-shot adaptive gaze estimation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9368–9377.

[48] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, A. Torralba, Eye tracking for everyone, 2016. arXiv:1606.05814.

[49] C. Meng, X. Zhao, Webcam-based eye movement analysis using cnn, IEEE Access 5 (2017) 19581–19587.

[50] Y.-m. Cheung, Q. Peng, Eye gaze tracking with a web camera in a desktop environment, IEEE Transactions on Human-Machine Systems 45 (2015) 419–430.

[51] S. Saxena, L. K. Fink, E. B. Lange, Deep learning models for webcam eye tracking in online experiments, Behavior Research Methods 56 (2024) 3487–3503. URL: https://doi.org/10.3758/s13428-023-02190-6. doi:10.3758/s13428-023-02190-6.