

Article

Improving Eye-Tracking Data Quality: A Framework for Reproducible Evaluation of Detection Algorithms

Christopher Gundler ^{1,*} , Matthias Temmen ² , Alessandro Gulberti ³ , Monika Pötter-Nerger ³ 
and Frank Ückert ¹

¹ Institute for Applied Medical Informatics, University Medical Center Hamburg-Eppendorf, 20246 Hamburg, Germany; f.ueckert@uke.de

² EyeTrax GmbH & Co. KG, 49076 Osnabrück, Germany; matthias.temmen@eyetrax.de

³ Department of Neurology, University Medical Center Hamburg-Eppendorf, 20246 Hamburg, Germany; a.gulberti@uke.de (A.G.); m.poetter-nerger@uke.de (M.P.-N.)

* Correspondence: c.gundler@uke.de

Abstract: High-quality eye-tracking data are crucial in behavioral sciences and medicine. Even with a solid understanding of the literature, selecting the most suitable algorithm for a specific research project poses a challenge. Empowering applied researchers to choose the best-fitting detector for their research needs is the primary contribution of this paper. We developed a framework to systematically assess and compare the effectiveness of 13 state-of-the-art algorithms through a unified application interface. Hence, we more than double the number of algorithms that are currently usable within a single software package and allow researchers to identify the best-suited algorithm for a given scientific setup. Our framework validation on retrospective data underscores its suitability for algorithm selection. Through a detailed and reproducible step-by-step workflow, we hope to contribute towards significantly improved data quality in scientific experiments.

Keywords: eye-tracking; pupil detection algorithm; methodological framework; detection quality



Citation: Gundler, C.; Temmen, M.; Gulberti, A.; Pötter-Nerger, M.; Ückert, F. Improving Eye-Tracking Data Quality: A Framework for Reproducible Evaluation of Detection Algorithms. *Sensors* **2024**, *24*, 2688. <https://doi.org/10.3390/s24092688>

Academic Editors: Pamela Zontone, Antonio Affanni and Tad Brunye

Received: 26 January 2024

Revised: 18 April 2024

Accepted: 20 April 2024

Published: 24 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Eye-tracking technology has become pervasive in recording, assessing, and evaluating eye movements, gazes, and pupil reactions across diverse research fields [1]. The methodological approach gained increasing interest as a non-invasive, easy-applicable measure of brainstem activity in behavioral sciences and provides an indirect readout marker for brain state activity, alertness, and arousal levels [2]. In human and animal models, changes in attention, mental effort, and emotional distress are tracked by spontaneous fluctuations in pupil diameter [3–5]. Accordingly, pupillometry was used as an experimental measure to continuously monitor variations in wakefulness during periods of walking [6], a go/no-go signal detection task [6], and an auditory oddball detection task, where the pupil diameter exhibited a U-shaped relationship with the reaction time performing the task with shortest latencies at intermediate pupil diameters [7]. Thus, pupil diameter is a useful indicator of brain state, revealing neural mechanisms by which the brain is controlled on a moment-to-moment basis to predict optimal behavioral performance.

As the demand for objective measurements continues to grow, research and industry are driven to refine the required eye-tracking technology. Reliable measurements become even more important given their growing importance in medicine as an indirect biomarker to track disease states or to differentiate between disease entities. Pupil responses are used to discover subclinical affections of the optic nerve [8] and monitor fatigue in multiple sclerosis [9]. Quantitative pupil observations are widely used in intensive care units for monitoring patients with severe intracranial pathology [10] and are discussed to be useful in Alzheimer's disease detection, diagnosis, and monitoring, alone or in combination with additional biomarkers [11]. In Parkinson's disease, early pupillometry changes are

observed even in preclinical stages [12], which are further modified depending on the disease stage throughout the disease [13]. Advancements in these fields, resulting in reduced costs, decreased complexities, and improved accuracy, are expected to further enhance the prevalence and significance of eye-tracking technology in the future [14].

Since the late 19th century, stationary eye trackers have been extensively researched for scientific purposes [15]. More recently, head-mounted video oculography has emerged as a valuable alternative, enabling experimental setups that were previously impossible [16]. Advancements in optical technology have led to the incorporation of cameras into glasses-like devices or virtual reality headsets, enabling continuous eye recording using non-visible near-infrared light. Unlike the proprietary devices used in the past, this technology offers transparency in the recording process and allows for re-evaluation with advances in analysis methods: affordable off-the-shelf technology is readily available or could be tailored to specific needs [17,18]. Consequently, this eye-tracking technology aligns directly with the four principal components of open science [19].

Despite the advantages of such flexible and archivable tailor-made solutions, a significant responsibility lies in fine-tuning the factors that influence the recording process. Alongside the physical camera setup, the pupil detection algorithms represent a crucial component of the eye-tracking pipeline. Despite the apparent simplicity of detecting the corresponding blob of pixels in a near-infrared image, this task remains challenging and unsolved. Factors such as contact lenses, dust, makeup, and other external influences further complicate the measurement process [20]. Given the plethora of available algorithms, each with distinct advantages and disadvantages, researchers must carefully choose a suitable method and associated hyperparameters based on their specific research objectives [21]. This publication aims to provide a framework for making such choices that are grounded in reliable evidence and ease of application.

Recording and analyzing eye-tracking data present challenges that are far from trivial [22]. However, these well-known complexities have resulted in a significant body of published knowledge and advice in literature [23,24]. These insights encompass a wide range of topics, including recording setups and devices [14,15], strategies for optimizing data quality [25], and proper analysis methods [26], providing valuable guidance for the appropriate use of this powerful methodology. As a result, researchers who aim to utilize eye-tracking technology can benefit from a rich ecosystem of scientific work that serves as a foundation for their investigations.

1.1. Navigating the Landscape of Eye-Tracking Algorithms

To assist applied researchers in selecting an appropriate pupil detection algorithm for their measurement pipeline, evaluation surveys like that of Fuhl et al. [27] have been conducted to compare and rate these algorithms. The diversity of potential recording designs has led to the existence of different datasets [28]. These datasets have been used for training or evaluating pupil detection algorithms, resulting in varying types of labels associated with the samples depending on three major types of algorithms:

1. The most fundamental type of annotation involves detecting the pupil center within the image [27]. With the two-dimensional coordinates of the detected pupil center and a commonly estimated three-dimensional eye model, subsequent procedures can calculate the corresponding eyeball rotation [29].
2. In scenarios where pupil size is of interest, such as pupillography, algorithms typically yield the best-fitting ellipse enclosing the pupil [17]. The output provides information on the size of the major and minor axes along with their rotation and two-dimensional position.
3. A more versatile representation utilizes a segmentation map covering the entire sample [28]. This segmentation mask is a binary mask where only the pupil is indicated. Some of these algorithms may also provide data on other eye components, such as the iris and sclera. Theoretically, this encoding allows for the use of partially hidden pupils due to eyelids or blinks.

Importantly, these three different annotation types create a hierarchy of information. The segmentation mask contains the most comprehensive information, but reducing it to an ellipse or even the pupil center is feasible. Researchers rely on large annotated datasets, assuming that algorithms performing well on them will generalize effectively to unseen setups and subjects. This generalization is crucial for practical applications where robustness and reliability are paramount considerations.

1.2. Choosing an Algorithm for Own Research

The concept of generalizability plays a crucial role in the assumption of algorithm performance [30]. However, assuming that sufficient performance on one dataset will translate to sufficient performance on a custom setup is not guaranteed. Several authors, such as Cazzato et al. [16], have emphasized the importance of custom adaptations to achieve satisfactory performance. These complexities arise from various aspects:

- Recorded samples vary significantly based on the camera position, resolution, and distance [31]. As a result, samples from different recording setups are not directly comparable. The non-linear transformation of the pupil when viewed from larger eye angles can present additional challenges to the algorithms [32].
- Algorithms often require the setting of hyperparameters that are dependent on the specific samples. Many of these hyperparameters have semantic meanings and are tailored to the camera's position. While reusing published values may suffice if the setups are similar enough, obtaining more suitable detections will likely depend on fine-tuning these parameters.
- The population of subjects may differ considerably due to the context of the measurement and external factors. In medical contexts, specific correlated phenotypes may seriously hinder detection rates. There is a scarcity of published work, such as Kulkarni et al. [33], that systematically evaluates induced bias in pupil detection. Furthermore, challenges exist even within the general population, as documented by Fuhl et al. [27]. For instance, detecting pupils in participants wearing contact lenses requires detectors to perform well under this specific condition without introducing bias.
- Metrics used for performance evaluation can vary significantly between studies. Often, metrics are chosen to optimally assess a specific dataset or use case. For instance, the evaluation paper by Fuhl et al. [27] used a threshold of five pixels to classify the detection of the pupil center inside a sample as correct. While this choice is sound for the tested datasets, samples with significantly different resolutions due to another camera setup necessitate adopting alternative concepts.

Given these complexities, evaluating pupil detection algorithms requires careful consideration of their context. Making general claims of superior performance compared to all competitors is challenging. Consequently, custom considerations and evaluations in the application of pupil detection algorithms remain necessary to ensure their appropriate and reliable use in specific setups.

In addition to the hypothetical overall performance, various key concepts influence the decision in favor of one pupil detection algorithm over another. Similar to other machine learning use cases, softer concepts become relevant in the context of eye tracking [16]. For example, the transparency and interpretability of an algorithm might pose an important dimension, especially in sensitive medical areas where understanding the reasons behind a specific output becomes crucial for proper interpretation and validation. Another significant consideration is the licensing of the algorithm. Some algorithms may have licensing restrictions that limit their usage to non-commercial purposes. In academic settings, such licenses might be incompatible with classical open-source licenses like the GPL (General Public License), which requires re-licensing under the same terms. This can hinder the proper publication and dissemination required for transparent science. Taking all these factors into account further complicates the process of selecting an appropriate pupil detector beyond its assumed detection performance.

Even with a solid understanding of the literature, selecting the most suitable algorithm for a specific research project poses a challenge. To the best of our knowledge, no published piece of software provides more than six algorithms with the same API [17]. Therefore, empowering applied researchers to choose the best-fitting detector for their research needs is the primary contribution of this paper. Providing such a framework that facilitates the evaluation and comparison of pupil detection algorithms could allow researchers to make informed decisions and enhance the quality and reliability of eye-tracking data analysis.

2. Methods

The objective of this publication is the development and usage of a framework to empower researchers across diverse scientific disciplines to independently assess pupil detection algorithms in their experiments. The software must be inclusive, user-friendly, and not require in-depth knowledge of the algorithm's technical details or implementation. Under the premise that the researcher should be able to optimize the measurements of their raw experimental data for the sake of more sustainable science.

2.1. Defining Criteria for the Framework

To uphold fundamental principles, we have defined essential criteria that any proposed framework must meet to effectively support academic practice. Automating the complexities associated with the evaluation process is paramount, minimizing the need for manual intervention. However, a precise definition of underlying constraints is necessary to ensure a sustainable and reusable utility.

- **Flexibility:** The proposed framework must exhibit maximum flexibility in its hardware and software environment for seamless execution. It should operate offline without reliance on remote servers, enabling widespread use in all countries.
- **Accessibility:** Additionally, the framework should not be tied to specific commercial systems that may pose accessibility issues due to license regulations or fees. Applied researchers should have the freedom to use the framework directly on their existing experimental systems, avoiding data duplication, preserving privacy, and simplifying knowledge management. As such, the framework should be compatible with a wide range of hardware, including UNIX-based operating systems commonly licensed as open-Source software, as well as the popular yet proprietary Microsoft Windows.
- **Ease of setup:** Once the system is available, setting up the framework should be straightforward and not require advanced technical knowledge. This may appear trivial but is complicated by the diversity of pupil detection algorithms. Existing implementations often depend on various programming languages and require multiple libraries and build tools, making the installation process challenging and time-consuming. To overcome this issue, the framework should not demand manual setup, but facilitating faster, and achieve assessment results easier.
- **Scalability:** The proposed framework must be scalable to handle the large volumes of samples in modern datasets, the diversity of algorithms, and the numerous tunable hyperparameters. Fortunately, the independence of algorithms and samples allows for easy parallelization of detections, enabling the efficient utilization of computational resources. The framework should be capable of benefiting from a single machine, multiple virtual machines, or even a cluster of physical devices, ensuring efficient exploration of the vast search space.
- **Modularity and standardization:** The framework should be designed with a modular approach and adhere to established standards and best practices. Embracing existing standards simplifies support and ensures sustainable development. Moreover, adhering to these standards allows for the re-use of individual components within the system, facilitating the integration of selected pupil detection algorithms into the final experiment seamlessly.
- **Adaptability for researchers and developers:** The framework should not only cater to researchers employing pupil detection algorithms but also be accessible to devel-

opers creating new detectors. By simplifying the evaluation process, developers may enhance their algorithms.

Given the defined constraints, a setup based on microservices and distributed computing appears highly appealing.

2.2. Inclusion Criteria of the Pupil Detection Algorithms

To ensure a comprehensive and meaningful comparison of pupil detection algorithms, we conducted a literature analysis to gather the currently available implementations. This investigation was built upon an extensive review by Zandi et al. [17], which served as the foundational reference for identifying relevant papers and follow-up publications. Additionally, the combination of considered algorithms based on their work and our additional research is shown in Table 1.

Please note that while this collection may not be fully exhaustive, it encompasses the majority of algorithms currently employed by researchers in the field. Each algorithm was individually evaluated for inclusion based on specific criteria:

- **Availability of implementations:** To ensure reproducibility, the published algorithms had to be accompanied by associated implementations. Although textual descriptions may exist, replicating an algorithm without access to its original implementation can introduce unintended variations, leading to inconsistent results. Therefore, only algorithms with readily available and accurate implementations as intended by the original authors were included.
- **Independence of dependencies and programming languages:** While no strict enforcement of specific dependencies or programming languages was imposed, a preference was given to algorithms that could be executed on UNIX-based systems. This choice was driven by the desire to avoid proprietary components and promote open-source software in science. As a result, algorithms solely available as compiled Microsoft Windows libraries without accompanying source codes were excluded. Similarly, algorithms implemented in scripting languages requiring a paid license, such as MATLAB, were not included.

Table 1. Overview of publications featuring pupil detection algorithms examined in our study. A checkmark enclosed in brackets denotes cases where the code is unavailable, but a compiled library is provided. Explanations for the exclusion of certain algorithms from the subsequent analysis are explicitly detailed in the last column.

Publication	Code Available?	Programming Language	Included?
Akinlar et al. [34]	✓	Python	✓
Xiang et al. [35]	-		Excluded: Code not available
Bonteanu et al. [36]	-		Excluded: Code not available
Cai et al. [37]	-		Excluded: Code not available
Fuhl et al. [28]	-		Excluded: Code not available
Kothari et al. [38]	✓	Python	✓
Larumbe-Bergera et al. [39]	-		Excluded: Code not available
Shi et al. [40]	-		Excluded: Code not available
Wan et al. [41]	✓	Python	✓
Wang et al. [42]	✓	Python	✓
Bonteanu et al. [43]	-		Excluded: Code not available
Fuhl et al. [44]	-		Excluded: Code not available
Han et al. [45]	✓	Python	Excluded: No weights for the neural network
Manuri et al. [46]	-		Excluded: Code not available
Bonteanu et al. [47]	-		Excluded: Code not available
Bonteanu et al. [48]	-		Excluded: Code not available
Bonteanu et al. [49]	-		Excluded: Code not available
Bozomitu et al. [50]	-		Excluded: Code not available

Table 1. Cont.

Publication	Code Available?	Programming Language	Included?
Eivazi et al. [51]	✓	Python	✓
Han et al. [52]	-		Excluded: Code not available
Krause et al. [53]	✓	C++	✓
Miron et al. [54]	-		Excluded: Code not available
Yiu et al. [55]	✓	Python	Excluded: Unable to specify the container
Fuhl et al. [56]	(✓)	C++	Excluded: Binary library not available for Linux
Fuhl et al. [57]	(✓)	C++	✓
George et al. [58]	-		Excluded: Code not available
Li et al. [59]	-		Excluded: Code not available
Martinikorena et al. [60]	✓	MATLAB	Excluded: Requires proprietary interpreter
Santini et al. [61]	✓	C++	Excluded: Temporal extension of another algorithm
Santini et al. [62]	✓	C++	✓
Vera-Olmos et al. [63]	✓	Python	Excluded: Unable to specify the container
Fuhl et al. [64]	-		Excluded: Code not available
Topal et al. [65]	-		Excluded: Code not available
Vera-Olmos et al. [66]	-		Excluded: Code not available
Fuhl et al. [67]	-		Excluded: Code not available
Fuhl et al. [27]	✓	C++	✓
Fuhl et al. [68]	✓	C++	✓
Javadi et al. [69]	✓	.NET	Excluded: Not available for Linux
Świrski et al. [70]	✓	C++	✓
Kassner et al. [71]	✓	Python	✓
Li et al. [72]	✓	C++	✓

✓: Yes; (✓): Code is unavailable, but a compiled library is provided; -: No.

The implementations that satisfied the criteria above were deemed suitable for inclusion within the framework. As part of our selection process, we did not require real-time execution of the algorithms, recognizing that offline analyses may be appropriate and necessary for specific experimental setups. To the best of our knowledge, this framework constitutes the most extensive collection available for evaluation purposes.

2.3. Architecture and Design of the Framework

The proposed, newly developed framework of this study employs a microservice architecture that was chosen for its inherent advantages in robustness, scalability, and compatibility (Figure 1). In this architecture, the minimal building blocks are represented as fully autonomous containers, providing self-contained and lightweight visualizations. These containers utilize the existing kernel of the operating systems, rendering complete simulations unnecessary, as seen in virtual machines. Each container encompasses all the essential dependencies required for the implementation of a specific pupil detection algorithm, encompassing the necessary software components, libraries, data dependencies, and the source code itself. This comprehensive encapsulation of the algorithm's environment eliminates the need for manual installation and configuration, streamlining the deployment process for scientists.

The microservice architecture ensures the autonomy of each container, making them self-sufficient and isolated. As a result, the licensing of the container content corresponds precisely to the included code, enhancing intellectual property protection. This autonomy also contributes to the flexibility and suitability of the architecture in accommodating the diverse range of pupil detection algorithms.

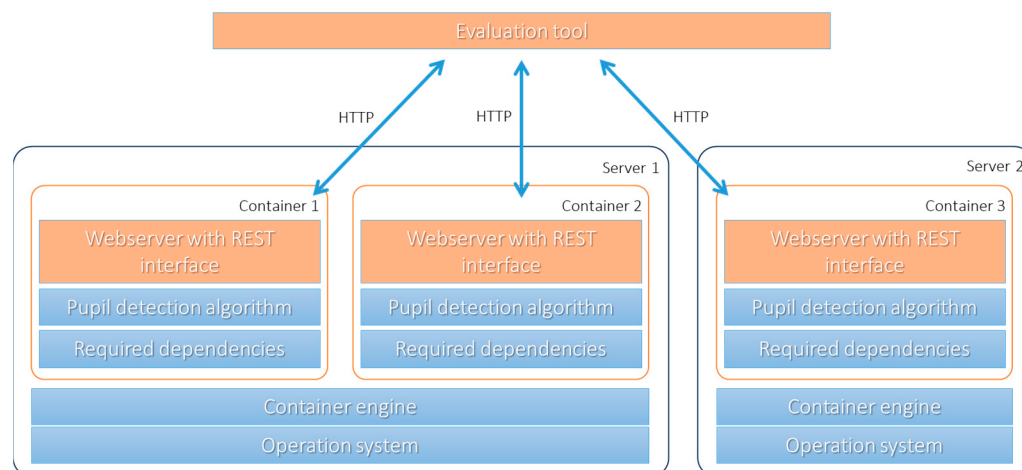


Figure 1. Visualization of a possible research setup involving two servers running three containers with pupil detection algorithms. All orange parts are part of the framework defined in this work. The evaluation tool sends the eye frames as images to the containers given their URL, where a web server providing a REST interface validates the data, hands it over to the pupil detection algorithm, and returns the value to the evaluation tool. As the number of containers per server is only limited in terms of the available computational resources, the system can scale flexibly.

Inspired by industry trends, the design of this architecture acknowledges the collective deployment advantages of autonomous services. Container runtimes, supporting various conditions and requirements, facilitate the execution of individual containers. The concept of hardware is largely abstracted in this setup, allowing the seamless utilization of physical hardware, virtual machines, or cloud-based resources. The versatility of this approach enables researchers to harness all available resources, facilitating scalability and enhancing the framework's efficiency. As the experiment's computational demands increase, the architecture scales nearly linearly with the available resources, promoting efficient utilization and resource allocation.

Services delivered in the form of containers rely on well-documented interfaces to facilitate interaction. The inner workings of the container implementation remain opaque and are treated as a black box. Therefore, communication and data transmission necessitate a robust network stack. We utilized the Hypertext Transfer Protocol (HTTP) due to its wide adoption and the availability of corresponding server and client libraries for most programming languages. Additionally, all the interfaces were designed to follow the Representational State Transfer (REST) style, which enables human-readable communication with the containers, enhancing compatibility and accessibility. To ensure standardization and interoperability, we employed the OpenAPI standard to allow for the automatic discovery, querying, and testing of the available resources. With these well-established building blocks, we designed a unified application interface for all pupil detectors, enabling their flexible use in diverse contexts.

Given the containers with well-defined REST application interfaces, most programming languages support sending samples and receiving estimated pupil data. Consequently, researchers can utilize a container runtime and develop evaluation procedures individually. However, managing the lifecycle of containers and handling input and output operations may become repetitive and cumbersome. To address this, the proposed framework offers an optional management tool designed for performance optimization. Controlled through a command line interface and a user-friendly configuration file, this software facilitates scalable and memory-efficient bulk processing of multiple samples. The obtained pupil estimates, encoded as JSON, then serve as the foundation for specific evaluations conducted by the researcher. This streamlined process simplifies the evaluation of multiple pupil detection algorithms, allowing researchers to focus on the core scientific aspects of their experiments.

2.4. Validation Data and Procedure

To demonstrate the potential of our proposed framework, we utilized a novel dataset. As a prototypical example of a specific research setup that might often occur in scientific reality, we utilized the collection of eye movement data within a virtual reality (VR) environment. The immersive nature of VR often introduces constraints in terms of the camera positioning and angle. For the exemplary utilization of our proposed framework, we utilized a novel dataset originally recorded by the company eyeTrax GmbH & Co. KG (Osnabrück, Germany). Before data collection, the participants provided informed consent, ensuring compliance with ethical guidelines for human subjects research. The dataset comprised manually extracted frames from the eye videos of 44 pseudonymized participants. The recording itself was based on cameras by the company Pupil Labs GmbH (Berlin, Germany) recording both eyes with a temporal resolution of 200 frames per second and a spatial resolution of 200 by 200 pixels. The secondary usage of this novel dataset ensured that it had not been previously used for training of machine-learning-based algorithms. Accordingly, it allowed us to analyze the data as a representative sample of a hypothetical population that could be observed within a specific research setup or commercial application.

In general, our framework is flexible enough to account for an unsupervised setup where, once a new pupil detection algorithm with unique advantages is developed, its authors may choose to test its consistency with existing algorithms. However, most authors will likely try to find the best approach for their setup. In this case, a labeled ground truth dataset is required. We performed manual annotation of the center of the pupil on four images per subject to establish a diverse gold standard. To ensure a representative sample, the 176 frames were randomly selected and stratified for both the left and right eye. In instances where the pupil was only partially visible, such as shortly before a blink, we approximated the center of the pupil and indicated the value as an approximation to understand the behavior of the algorithms under challenging conditions.

3. Results

The notion that a single pupil detection algorithm could universally cater to all scenarios is highly implausible. In this chapter, we describe an exemplary analysis conducted using our proposed framework to identify an optimal pupil detection algorithm for our given scientific setup. For the following steps, an interactive Jupyter Notebook is available to replicate the findings and apply them to custom data without advanced knowledge in a programming language.

3.1. Defining our Evaluation Criteria

Given the definition of our tasks as a supervised task and our labeled ground truth, we need to define our metric of interest. A possible option would be a binary label indicating whether an observation was similar to the specific ground truth.

For our hypothetical setup, we choose to rather measure the error continuously as the Euclidean distance from our reference value. For easier interpretation of the findings, Figure 2 provides an overview of the significance of such continuous error measurements in our specific eye-tracking setup. That visualization helps us determine which values might be considered sufficient and which values might be considered good. Given the specific resolution and our human intuition, only an estimate below 10 pixels appears appropriate.

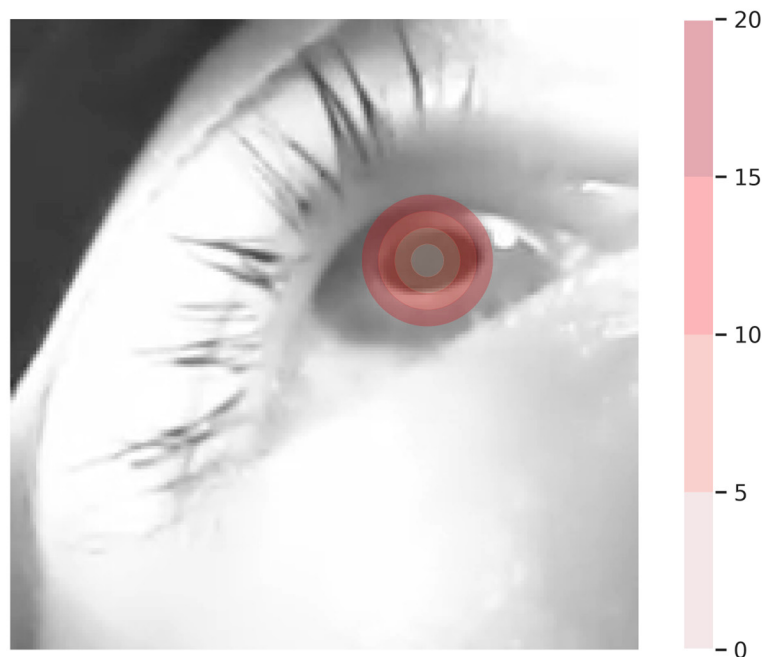


Figure 2. Visualization depicting the influence of diverse error magnitudes, quantified by the Euclidean distance between the annotated pupil center and potential predictions, on raw samples sized at 200 by 200 pixels. The center of the circles represents our ground truth annotation. Notably, errors surpassing 10 pixels appear substantial and may significantly compromise the accuracy of the detection process.

3.2. Generating the Predictions of all Pupil Detection Algorithms

Given the annotated ground truth, we can generate the predictions of the individual algorithms. For this purpose, we provide two different approaches: A Python script could serve as a foundation for developing custom pipelines. The script parses command line arguments and prints the results in a text file with comma-separated values. For those users with limited programming experience, we alternatively provide software parsing a config file where the necessary configuration could be defined with a simple text editor. While an additional graphical user interface may be even easier, we decided against its implementation due to the difficulties of implementing it in a platform-independent way. We applied all pupil detection algorithms with their default hyperparameters to the entire dataset.

3.3. Evaluation of the Pupil Detection Algorithms

The pupil detection algorithms can be evaluated. Utilizing the annotated ground truth dataset and the predictions of the pupil detection algorithms, we calculated the Euclidean distance between both positions in pixel space to obtain the errors. These values are depicted in Figure 3 and serve as a quantitative measure of the accuracy and performance of each algorithm in capturing the true pupil center positions.

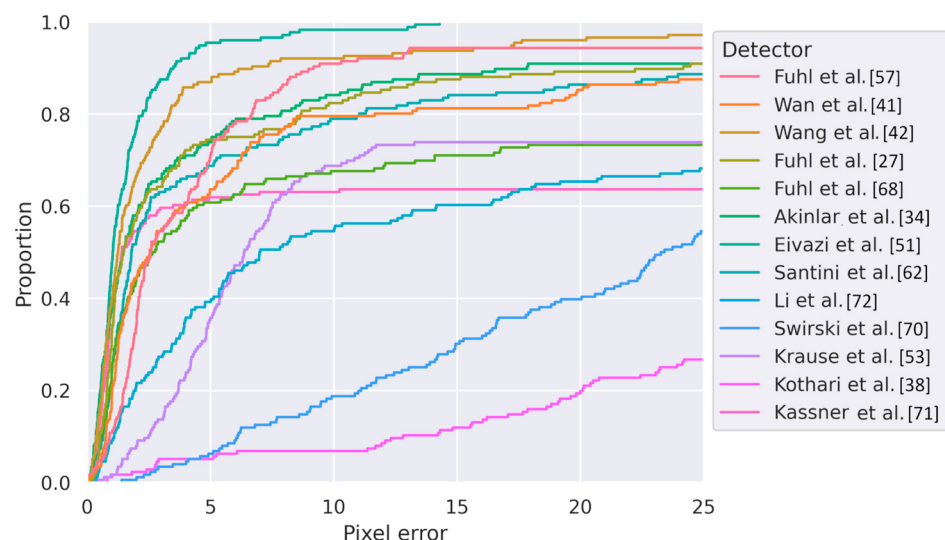


Figure 3. Empirical cumulative distribution functions illustrate the (Euclidean) pixel error across various pupil detectors, employing the default parameters as specified by the original authors on our dataset. The ordinate directly represents quantiles, with steeper curves indicating algorithms better suited to the specific dataset. Notably, certain algorithms predict over 80 percent of samples with errors below five pixels, while others encounter greater challenges. It is crucial to note that the algorithms were not fine-tuned, and the results should be interpreted as the lower boundary of achievable detection quality [27,34,38,41,42,51,53,57,62,68,70–72].

Upon initial examination, visualizing the errors using empirical cumulative distribution functions (ECDF), f_i offers significant advantages for comparing the performance of different pupil detection algorithms. Firstly, by restricting the x-axis to the error values already visualized in Figure 2, we can estimate the number of samples for which individual detectors produce entirely incorrect results. The presence of outliers and large values does not hinder visualization as in a histogram [73].

Secondly, interesting quantiles like the median could be directly read from the y-axis. These robust estimators provide valuable insights into the quality of each algorithm's performance. Finally, the individual functions of the ECDF offer a visual means of estimating the overall performance of each detector, similar to interpreting a receiver operating characteristic (ROC) curve. The steepest and most leftward rise to 100% corresponds to the most appropriate detector given the characteristics of the recorded samples, subjects, annotators, and evaluation metric. Accordingly, the detector by Eivazi et al. [51] appears to be a promising candidate for application on similar data.

3.4. Testing for Statistically Significant Performance Differences in Pupil Detectors

The empirical cumulative distribution functions serve as a basis for a first comparison of the pupil detection algorithms. However, it is essential to recognize that they are approximations of the true underlying distributions $F_i(x)$. To reason under the resulting uncertainty, estimates of confidence intervals using methods like the Dvoretzky–Kiefer–Wolfowitz inequality or statistical tests become necessary. As an exemplary research question, a researcher might aim to determine if utilizing a separate pupil detection algorithm is worth the effort compared to sticking with the existing algorithm by Pupil Labs, which is already in use. In this context, the researcher could define the null hypothesis (H_0) as " $F_i(x) \geq F_j(x)$ for all x ," and the alternative hypothesis (H_1) as " $F_i(x) < F_j(x)$ for all x ."

For comparing the empirical cumulative distribution functions, the Kolmogorov–Smirnov test provides a robust and non-parametric foundation for generating additional evidence. Setting f_i to represent the best-performing algorithm and F_j to represent the algorithm by Pupil Labs, the test rejects the null hypothesis with a p -value less than 0.0001. Consequently, we can safely assume that the alternative pupil detection algorithm might be

more suitable for recording setups resembling the annotated ground truth. This finding further supports the utility of error visualization and statistical tests in making informed decisions regarding the choice of pupil detection algorithms for specific eye-tracking scenarios.

4. Discussion

The increasing popularity of eye-tracking technology for addressing sophisticated research questions has led to greater access to raw recordings, promoting reproducible research in response to the demand for open science. Consequently, there is a growing need for efficient tools to optimize the employed methods.

In this context, our developed framework addresses this requirement by providing building blocks that allow practitioners to easily assess the suitability of various algorithms for different research paradigms. The time-consuming manual setup of the requirements and environments for each approach is replaced by a solution providing clear interfaces. Its flexibility enables seamless integration with a wide range of published pupil-detection algorithms and eye-tracking setups. The standardized interface further facilitates the implementation of custom evaluation suites, the integration into existing software, and the inclusion of novel detectors into the framework.

However, certain limitations must be considered for practical application. Firstly, the currently available implementations may not be optimized for speed, prompting the exploration of optimized implementations, offloading work to remote servers, or employing parallel processing to enhance performance. It is important to emphasize that the framework is primarily intended for offline analysis after data recording, not real-time applications during experiments. Secondly, the containers used as building blocks for hardware independence are not virtual machines. Accordingly, they do not account for different processor architectures. If the pupil detectors or their dependencies rely on those subtle differences for increased performance, we cannot compensate. Running some of the detectors is not possible, for example, on those devices facilitating ARM chipsets. Thirdly, the retrospective nature of the analysis relies on access to raw data, making the recording of video streams necessary to ensure re-analysis and consistency in future studies. Ensuring access to raw data rather than solely derived values like pupil center and diameter remains crucial due to the wide range of devices and hardware configurations available. Finally, although our framework significantly reduces the workload for researchers in evaluating algorithms for their specific workflow, some manual labor is still required. Currently, the selection of algorithm parameters cannot be automated, necessitating researchers to examine these parameters manually for optimization purposes after an initial selection. Automation efforts could enhance the framework's usability and efficiency.

For future work, we additionally suggest expanding the framework to include other essential pupil parameters beyond the pupil center, such as pupil ellipse measurements, which are crucial for pupillometry studies. Integrating these parameters would lead to a more comprehensive and accurate characterization of pupil behavior. Additionally, we propose employing the framework to investigate methodological questions beyond individual setups. For instance, it could facilitate a comparative analysis of various state-of-the-art algorithms for underrepresented groups, thus mitigating potential biases in eye-tracking studies.

5. Conclusions

The notion that a single pupil detection algorithm could universally handle all eye-tracking research scenarios is highly implausible. We present a comprehensive, flexible, scalable, easy-to-set-up, modular, and adaptable framework to identify the best-suited algorithm for a given scientific setup. While certain limitations exist, our software presents a versatile and effective tool for researchers seeking to enhance the accuracy and reliability of their eye-tracking analyses. In providing a unified interface for 13 pupil detection algorithms, we more than double the number of algorithms currently usable within a single

software package. Given its representation of the currently available algorithms, we hope to further stimulate the methodological development in eye-tracking research.

Author Contributions: Conceptualization, methodology, and software, C.G.; validation, M.T.; writing—original draft preparation, C.G.; writing—review and editing, C.G., M.T., A.G., M.P.-N. and F.Ü. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge financial support from the Open Access Publication Fund of UKE—Universitätsklinikum Hamburg-Eppendorf and DFG—German Research Foundation for open access publication.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all the participants involved in the study.

Data Availability Statement: The framework is available on GitHub <https://github.com/Christopher2/ommatidia>, accessed on 20 April 2024. The datasets of the individual predictions by the algorithms generated and analyzed during the current study required for replication are available in the ZFDM repository with the DOI <https://doi.org/10.25592/uhhfdm.13720>. The raw frames are not publicly available due to privacy constraints but are available from the corresponding author upon reasonable request.

Conflicts of Interest: M.T. is a shareholder of eyeTrax GmbH & Co. KG, which develops medical software for eye-tracking. The other authors declare no conflicts of interest.

References

1. Punde, P.A.; Jadhav, M.E.; Manza, R.R. A Study of Eye Tracking Technology and Its Applications. In Proceedings of the 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM), Aurangabad, India, 5–6 October 2017; pp. 86–90.
2. Larsen, R.S.; Waters, J. Neuromodulatory Correlates of Pupil Dilation. *Front. Neural Circuits* **2018**, *12*, 21. [\[CrossRef\]](#) [\[PubMed\]](#)
3. McGinley, M.J.; Vinck, M.; Reimer, J.; Batista-Brito, R.; Zagha, E.; Cadwell, C.R.; Tolias, A.S.; Cardin, J.A.; McCormick, D.A. Waking State: Rapid Variations Modulate Neural and Behavioral Responses. *Neuron* **2015**, *87*, 1143–1161. [\[CrossRef\]](#)
4. Iriki, A.; Tanaka, M.; Iwamura, Y. Attention-Induced Neuronal Activity in the Monkey Somatosensory Cortex Revealed by Pupillometrics. *Neurosci. Res.* **1996**, *25*, 173–181. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Alnæs, D.; Sneve, M.H.; Espeseth, T.; Endestad, T.; van de Pavert, S.H.P.; Laeng, B. Pupil Size Signals Mental Effort Deployed during Multiple Object Tracking and Predicts Brain Activity in the Dorsal Attention Network and the Locus Coeruleus. *J. Vis.* **2014**, *14*, 1. [\[CrossRef\]](#)
6. McGinley, M.J.; David, S.V.; McCormick, D.A. Cortical Membrane Potential Signature of Optimal States for Sensory Signal Detection. *Neuron* **2015**, *87*, 179–192. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Murphy, P.R.; Robertson, I.H.; Balsters, J.H.; O’connell, R.G. Pupillometry and P3 Index the Locus Coeruleus–Noradrenergic Arousal Function in Humans. *Psychophysiology* **2011**, *48*, 1532–1543. [\[CrossRef\]](#)
8. Muhafiz, E.; Bozkurt, E.; Erdoğan, C.E.; Nizamoğulları, Ş.; Demir, M.S. Static and Dynamic Pupillary Characteristics in Multiple Sclerosis. *Eur. J. Ophthalmol.* **2022**, *32*, 2173–2180. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Guillemain, C.; Hammad, G.; Read, J.; Requier, F.; Charonitis, M.; Delrue, G.; Vandeleene, N.; Lommers, E.; Maquet, P.; Collette, F. Pupil Response Speed as a Marker of Cognitive Fatigue in Early Multiple Sclerosis. *Mult. Scler. Relat. Disord.* **2022**, *65*, 104001. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Prescott, B.R.; Saglam, H.; Duskin, J.A.; Miller, M.I.; Thakur, A.S.; Gholap, E.A.; Hutch, M.R.; Smirnakis, S.M.; Zafar, S.F.; Dupuis, J.; et al. Anisocoria and Poor Pupil Reactivity by Quantitative Pupillometry in Patients with Intracranial Pathology. *Crit. Care Med.* **2022**, *50*, e143–e153. [\[CrossRef\]](#)
11. Chougule, P.S.; Najjar, R.P.; Finkelstein, M.T.; Kandiah, N.; Milea, D. Light-Induced Pupillary Responses in Alzheimer’s Disease. *Front. Neurol.* **2019**, *10*, 360. [\[CrossRef\]](#)
12. La Morgia, C.; Romagnoli, M.; Pizza, F.; Biscarini, F.; Filardi, M.; Donadio, V.; Carbonelli, M.; Amore, G.; Park, J.C.; Tinazzi, M.; et al. Chromatic Pupillometry in Isolated Rapid Eye Movement Sleep Behavior Disorder. *Mov. Disord.* **2022**, *37*, 205–210. [\[CrossRef\]](#) [\[PubMed\]](#)
13. You, S.; Hong, J.-H.; Yoo, J. Analysis of Pupillometer Results According to Disease Stage in Patients with Parkinson’s Disease. *Sci. Rep.* **2021**, *11*, 17880. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Stein, N.; Niehorster, D.C.; Watson, T.; Steinicke, F.; Rifai, K.; Wahl, S.; Lappe, M. A Comparison of Eye Tracking Latencies Among Several Commercial Head-Mounted Displays. *i-Perception* **2021**, *12*, 2041669520983338. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Cognolato, M.; Atzori, M.; Müller, H. Head-Mounted Eye Gaze Tracking Devices: An Overview of Modern Devices and Recent Advances. *J. Rehabil. Assist. Technol. Eng.* **2018**, *5*, 2055668318773991. [\[CrossRef\]](#) [\[PubMed\]](#)

16. Cazzato, D.; Leo, M.; Distante, C.; Voos, H. When I Look into Your Eyes: A Survey on Computer Vision Contributions for Human Gaze Estimation and Tracking. *Sensors* **2020**, *20*, 3739. [[CrossRef](#)] [[PubMed](#)]
17. Zandi, B.; Lode, M.; Herzog, A.; Sakas, G.; Khanh, T.Q. PupilEXT: Flexible Open-Source Platform for High-Resolution Pupillometry in Vision Research. *Front. Neurosci.* **2021**, *15*, 676220. [[CrossRef](#)] [[PubMed](#)]
18. Stengel, M.; Grogorick, S.; Eisemann, M.; Eisemann, E.; Magnor, M.A. An Affordable Solution for Binocular Eye Tracking and Calibration in Head-Mounted Displays. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 15–24.
19. Vicente-Saez, R.; Martinez-Fuentes, C. Open Science Now: A Systematic Literature Review for an Integrated Definition. *J. Bus. Res.* **2018**, *88*, 428–436. [[CrossRef](#)]
20. Tonsen, M.; Zhang, X.; Sugano, Y.; Bulling, A. Labelled Pupils in the Wild: A Dataset for Studying Pupil Detection in Unconstrained Environments. In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, Charleston, SC, USA, 14–17 March 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 139–142.
21. Chen, Y.; Ning, Y.; Kao, S.L.; Støer, N.C.; Müller-Riemenschneider, F.; Venkataraman, K.; Khoo, E.Y.H.; Tai, E.-S.; Tan, C.S. Using Marginal Standardisation to Estimate Relative Risk without Dichotomising Continuous Outcomes. *BMC Med. Res. Methodol.* **2019**, *19*, 165. [[CrossRef](#)] [[PubMed](#)]
22. Orquin, J.L.; Holmqvist, K. Threats to the Validity of Eye-Movement Research in Psychology. *Behav. Res.* **2018**, *50*, 1645–1656. [[CrossRef](#)]
23. Steinhauer, S.R.; Bradley, M.M.; Siegle, G.J.; Roecklein, K.A.; Dix, A. Publication Guidelines and Recommendations for Pupillary Measurement in Psychophysiological Studies. *Psychophysiology* **2022**, *59*, e14035. [[CrossRef](#)]
24. Godfroid, A.; Hui, B. Five Common Pitfalls in Eye-Tracking Research. *Second. Lang. Res.* **2020**, *36*, 277–305. [[CrossRef](#)]
25. Kret, M.E.; Sjak-Shie, E.E. Preprocessing Pupil Size Data: Guidelines and Code. *Behav. Res.* **2019**, *51*, 1336–1342. [[CrossRef](#)] [[PubMed](#)]
26. van Rij, J.; Hendriks, P.; van Rijn, H.; Baayen, R.H.; Wood, S.N. Analyzing the Time Course of Pupillometric Data. *Trends Hear.* **2019**, *23*, 233121651983248. [[CrossRef](#)] [[PubMed](#)]
27. Fuhl, W.; Santini, T.C.; Kübler, T.; Kasneci, E. EllSe: Ellipse Selection for Robust Pupil Detection in Real-World Environments. In Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, Charleston, SC, USA, 14–17 March 2016; ACM: New York, NY, USA, 2016; pp. 123–130.
28. Fuhl, W.; Schneider, J.; Kasneci, E. 1000 Pupil Segmentations in a Second Using Haar Like Features and Statistical Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Virtual, 11–17 October 2021; pp. 3466–3476.
29. Santini, T. Towards Ubiquitous Wearable Eye Tracking. Ph.D. Thesis, Universität Tübingen, Tübingen, Germany, 2019.
30. Kothari, R.S.; Bailey, R.J.; Kanan, C.; Pelz, J.B.; Diaz, G.J. EllSeg-Gen, towards Domain Generalization for Head-Mounted Eyetracking. *Proc. ACM Hum.-Comput. Interact.* **2022**, *6*, 1–17. [[CrossRef](#)]
31. Niehorster, D.C.; Santini, T.; Hessels, R.S.; Hooge, I.T.C.; Kasneci, E.; Nyström, M. The Impact of Slippage on the Data Quality of Head-Worn Eye Trackers. *Behav. Res.* **2020**, *52*, 1140–1160. [[CrossRef](#)]
32. Petersch, B.; Dierkes, K. Gaze-Angle Dependency of Pupil-Size Measurements in Head-Mounted Eye Tracking. *Behav. Res.* **2021**, *54*, 763–779. [[CrossRef](#)]
33. Kulkarni, O.N.; Patil, V.; Singh, V.K.; Atrey, P.K. Accuracy and Fairness in Pupil Detection Algorithm. In Proceedings of the 2021 IEEE Seventh International Conference on Multimedia Big Data (BigMM), Taichung, Taiwan, 14–17 November 2021; pp. 17–24.
34. Akinlar, C.; Kucukkartal, H.K.; Topal, C. Accurate CNN-Based Pupil Segmentation with an Ellipse Fit Error Regularization Term. *Expert. Syst. Appl.* **2022**, *188*, 116004. [[CrossRef](#)]
35. Xiang, Z.; Zhao, X.; Fang, A. Pupil Center Detection Inspired by Multi-Task Auxiliary Learning Characteristic. *Multimed. Tools Appl.* **2022**, *81*, 40067–40088. [[CrossRef](#)]
36. Bonteanu, P.; Bozomitu, R.G.; Cracan, A.; Bonteanu, G. A New Pupil Detection Algorithm Based on Multiple Angular Integral Projection Functions. In Proceedings of the 2021 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, 18–19 November 2021; pp. 1–4.
37. Cai, X.; Zeng, J.; Shan, S. Landmark-Aware Self-Supervised Eye Semantic Segmentation. In Proceedings of the 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Jodhpur, India, 15–18 December 2021; pp. 1–8.
38. Kothari, R.S.; Chaudhary, A.K.; Bailey, R.J.; Pelz, J.B.; Diaz, G.J. EllSeg: An Ellipse Segmentation Framework for Robust Gaze Tracking. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 2757–2767. [[CrossRef](#)]
39. Larumbe-Bergera, A.; Garde, G.; Porta, S.; Cabeza, R.; Villanueva, A. Accurate Pupil Center Detection in Off-the-Shelf Eye Tracking Systems Using Convolutional Neural Networks. *Sensors* **2021**, *21*, 6847. [[CrossRef](#)]
40. Shi, L.; Wang, C.; Jia, H.; Hu, X. EPS: Robust Pupil Edge Points Selection with Haar Feature and Morphological Pixel Patterns. *Int. J. Patt. Recogn. Artif. Intell.* **2021**, *35*, 2156002. [[CrossRef](#)]
41. Wan, Z.-H.; Xiong, C.-H.; Chen, W.-B.; Zhang, H.-Y. Robust and Accurate Pupil Detection for Head-Mounted Eye Tracking. *Comp. Electr. Eng.* **2021**, *93*, 107193. [[CrossRef](#)]
42. Wang, Z.; Zhao, Y.; Liu, Y.; Lu, F. Edge-Guided Near-Eye Image Analysis for Head Mounted Displays. In Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Bari, Italy, 4–8 October 2021; pp. 11–20.

43. Bonteanu, P.; Bozomitu, R.G.; Cracan, A.; Bonteanu, G. A Pupil Detection Algorithm Based on Contour Fourier Descriptors Analysis. In Proceedings of the 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME), Pitesti, Romania, 21–24 October 2020; pp. 98–101.
44. Fuhl, W.; Gao, H.; Kasneci, E. Tiny Convolution, Decision Tree, and Binary Neuronal Networks for Robust and Real Time Pupil Outline Estimation. In Proceedings of the ACM Symposium on Eye Tracking Research and Applications, Stuttgart, Germany, 2–5 June 2020; ACM: New York, NY, USA, 2020; pp. 1–5.
45. Han, S.Y.; Kwon, H.J.; Kim, Y.; Cho, N.I. Noise-Robust Pupil Center Detection Through CNN-Based Segmentation with Shape-Prior Loss. *IEEE Access* **2020**, *8*, 64739–64749. [[CrossRef](#)]
46. Manuri, F.; Sanna, A.; Petrucci, C.P. PDIF: Pupil Detection After Isolation and Fitting. *IEEE Access* **2020**, *8*, 30826–30837. [[CrossRef](#)]
47. Bonteanu, P.; Bozomitu, R.G.; Cracan, A.; Bonteanu, G. A New Pupil Detection Algorithm Based on Circular Hough Transform Approaches. In Proceedings of the 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), Cluj-Napoca, Romania, 23–26 October 2019; pp. 260–263.
48. Bonteanu, P.; Bozomitu, R.G.; Cracan, A.; Bonteanu, G. A High Detection Rate Pupil Detection Algorithm Based on Contour Circularity Evaluation. In Proceedings of the 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), Cluj-Napoca, Romania, 23–26 October 2019; pp. 264–267.
49. Bonteanu, P.; Cracan, A.; Bozomitu, R.G.; Bonteanu, G. A Robust Pupil Detection Algorithm Based on a New Adaptive Thresholding Procedure. In Proceedings of the 2019 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 21–23 November 2019; pp. 1–4.
50. Bozomitu, R.G.; Păsărică, A.; Tărniceriu, D.; Rotariu, C. Development of an Eye Tracking-Based Human-Computer Interface for Real-Time Applications. *Sensors* **2019**, *19*, 3630. [[CrossRef](#)] [[PubMed](#)]
51. Eivazi, S.; Santini, T.; Keshavarzi, A.; Kübler, T.; Mazzei, A. Improving Real-Time CNN-Based Pupil Detection through Domain-Specific Data Augmentation. In Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, Denver, CO, USA, 25–28 June 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1–6.
52. Han, S.Y.; Kim, Y.; Lee, S.H.; Cho, N.I. Pupil Center Detection Based on the UNet for the User Interaction in VR and AR Environments. In Proceedings of the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 23–27 March 2019; pp. 958–959.
53. Krause, A.F.; Essig, K. Boosting Speed- and Accuracy of Gradient Based Dark Pupil Tracking Using Vectorization and Differential Evolution. In Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, Denver, CO, USA, 25–28 June 2019; ACM: New York, NY, USA, 2019; pp. 1–5.
54. Miron, C.; Pasarica, A.; Bozomitu, R.G.; Manta, V.; Timofte, R.; Ciucu, R. Efficient Pupil Detection with a Convolutional Neural Network. In Proceedings of the 2019 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 21–23 November 2019; pp. 1–4.
55. Yiu, Y.-H.; Aboulatta, M.; Raiser, T.; Ophey, L.; Flanagan, V.L.; zu Eulenburg, P.; Ahmadi, S.-A. DeepVOG: Open-Source Pupil Segmentation and Gaze Estimation in Neuroscience Using Deep Learning. *J. Neurosci. Methods* **2019**, *324*, 108307. [[CrossRef](#)] [[PubMed](#)]
56. Fuhl, W.; Geisler, D.; Santini, T.; Appel, T.; Rosenstiel, W.; Kasneci, E. CBF: Circular Binary Features for Robust and Real-Time Pupil Center Detection. In Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, Warsaw, Poland, 14–17 June 2018; ACM: New York, NY, USA, 2018; pp. 1–6.
57. Fuhl, W.; Eivazi, S.; Hosp, B.; Eivazi, A.; Rosenstiel, W.; Kasneci, E. BORE: Boosted-Oriented Edge Optimization for Robust, Real Time Remote Pupil Center Detection. In Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, Warsaw, Poland, 14–17 June 2018; ACM: New York, NY, USA, 2018; pp. 1–5.
58. George, A.; Routray, A. ESCaF: Pupil Centre Localization Algorithm with Candidate Filtering. *arXiv* **2018**, arXiv:1807.10520.
59. Li, J.; Li, S.; Chen, T.; Liu, Y. A Geometry-Appearance-Based Pupil Detection Method for Near-Infrared Head-Mounted Cameras. *IEEE Access* **2018**, *6*, 23242–23252. [[CrossRef](#)]
60. Martinikorena, I.; Cabeza, R.; Villanueva, A.; Urtasun, I.; Larumbe, A. Fast and Robust Ellipse Detection Algorithm for Head-Mounted Eye Tracking Systems. *Mach. Vis. Appl.* **2018**, *29*, 845–860. [[CrossRef](#)]
61. Santini, T.; Fuhl, W.; Kasneci, E. PuReST: Robust Pupil Tracking for Real-Time Pervasive Eye Tracking. In Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, Warsaw, Poland, 14–17 June 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–5.
62. Santini, T.; Fuhl, W.; Kasneci, E. PuRe: Robust Pupil Detection for Real-Time Pervasive Eye Tracking. *Comput. Vis. Image Underst.* **2018**, *170*, 40–50. [[CrossRef](#)]
63. Vera-Olmos, F.J.; Pardo, E.; Melero, H.; Malpica, N. DeepEye: Deep Convolutional Network for Pupil Detection in Real Environments. *Integr. Comput.-Aided Eng.* **2018**, *26*, 85–95. [[CrossRef](#)]
64. Fuhl, W.; Santini, T.; Kasneci, G.; Rosenstiel, W.; Kasneci, E. PupilNet v2.0: Convolutional Neural Networks for CPU Based Real Time Robust Pupil Detection. *arXiv* **2017**, arXiv:1711.00112.
65. Topal, C.; Cakir, H.I.; Akinlar, C. APPD: Adaptive and Precise Pupil Boundary Detection Using Entropy of Contour Gradients arXiv 2018. arXiv:1709.06366.

66. Vera-Olmos, F.J.; Malpica, N. Deconvolutional Neural Network for Pupil Detection in Real-World Environments. In Proceedings of the Biomedical Applications Based on Natural and Artificial Computing, Corunna, Spain, 19–23 June 2017; Ferrández Vicente, J.M., Álvarez-Sánchez, J.R., de la Paz López, F., Toledo Moreo, J., Adeli, H., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 223–231.
67. Fuhl, W.; Santini, T.; Kasneci, G.; Kasneci, E. PupilNet: Convolutional Neural Networks for Robust Pupil Detection. *arXiv* **2016**, arXiv:1601.04902.
68. Fuhl, W.; Kübler, T.; Sippel, K.; Rosenstiel, W.; Kasneci, E. ExCuSe: Robust Pupil Detection in Real-World Scenarios. In *Computer Analysis of Images and Patterns*; Azzopardi, G., Petkov, N., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; Volume 9256, pp. 39–51. ISBN 978-3-319-23191-4.
69. Javadi, A.-H.; Hakimi, Z.; Barati, M.; Walsh, V.; Tcheang, L. SET: A Pupil Detection Method Using Sinusoidal Approximation. *Front. Neuroeng.* **2015**, *8*, 4. [[CrossRef](#)] [[PubMed](#)]
70. Świrski, L.; Bulling, A.; Dodgson, N. Robust Real-Time Pupil Tracking in Highly off-Axis Images. In Proceedings of the Symposium on Eye Tracking Research and Applications—ETRA’12, Santa Barbara, CA, USA, 28–30 March 2012; ACM Press: New York, NY, USA, 2012; p. 173.
71. Kassner, M.; Patera, W.; Bulling, A. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-Based Interaction. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Seattle, WA, USA, 13–17 September 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 1151–1160.
72. Li, D.; Winfield, D.; Parkhurst, D.J. Starburst: A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature-Based and Model-Based Approaches. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)—Workshops, San Diego, CA, USA, 21–23 September 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 3, p. 79.
73. Hammerla, N.Y.; Kirkham, R.; Andras, P.; Ploetz, T. On Preserving Statistical Characteristics of Accelerometry Data Using Their Empirical Cumulative Distribution. In Proceedings of the 2013 International Symposium on Wearable Computers, Zurich, Switzerland, 8–12 September 2013; ACM: New York, NY, USA, 2013; pp. 65–68.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.