Improving Real-Time CNN-Based Pupil Detection Through Domain-Specific Data Augmentation

Shaharam Eivazi
University of Tübingen, Perception Engineering
shahram.eivazi@uni-tuebingen.de

Thiago Santini
University of Tübingen, Perception Engineering
thiago.santini@uni-tuebingen.de

Alireza Keshavarzi
University of Tübingen, Perception Engineering
alireza.keshavarzi@student.uni-tuebingen.de

Thomas Kübler
University of Tübingen, Perception Engineering
thomas.kuebler@uni-tuebingen.de

Andrea Mazzei
Cortical Arts GmbH
amazzei@corticalarts.ch

ABSTRACT

Deep learning is a promising technique for real-world pupil detection. However, the small amount of available accurately-annotated data poses a challenge when training such networks. Here, we utilize non-challenging eye videos where algorithmic approaches perform virtually without errors to automatically generate a foundational data set containing subpixel pupil annotations. Then, we propose multiple domain-specific data augmentation methods to create unique training sets containing controlled distributions of pupil-detection challenges. The feasibility, convenience, and advantage of this approach is demonstrated by training a CNN with these datasets. The resulting network outperformed current methods in multiple publicly-available, realistic, and challenging datasets, despite being trained solely with the augmented eye images. This network also exhibited better generalization w.r.t. the latest state-of-the-art CNN: Whereas on datasets similar to training data, the nets displayed similar performance, on datasets unseen to both networks, ours outperformed the state-of-the-art by ≈27% in terms of detection rate.

CCS CONCEPTS

• **Computing methodologies** → **Image processing**; *Feature selection*; *Shape analysis*.

KEYWORDS

Pupil detection, Data augmentation, Deep learning

ACM Reference Format:
Shaharam Eivazi, Thiago Santini, Alireza Keshavarzi, Thomas Kübler, and Andrea Mazzei. 2019. Improving Real-Time CNN-Based Pupil Detection Through Domain-Specific Data Augmentation. In *2019 Symposium on Eye Tracking Research and Applications (ETRA '19), June 25–28, 2019, Denver , CO, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3314111.3319914

ETRA '19, June 25–28, 2019, Denver , CO, USA

ACM ISBN 978-1-4503-6709-7/19/06…$15.00
https://doi.org/10.1145/3314111.3319914

1 INTRODUCTION

Algorithmic approaches to pupil detection rely on traditional computer vision methods such as edge detection, intensity thresholds, and intensity gradient distribution. However, crafting a detector robust to the multitude of *eye-tracking challenges* (e.g., reflections, occlusion by eyelids, makeup) present in pervasive scenarios remains an elusive challenge. Conditions vary dramatically with scenario, for instance during driving [Kübler et al. 2015; Schmidt et al. 2017; Wood et al. 2017], museum visits [Santini et al. 2018a], shopping [Kasneci et al. 2014a], walking [Sugano and Bulling 2015], in an operating room [Tien et al. 2015], and during human-robot interaction [Aronson et al. 2018].

Similar to other computer vision tasks, as human-constructed algorithms' performance saturate, pupil detection is shifting towards data-driven approaches, as evidenced by recent machine-learning-oriented pupil detection studies such as [Chinsatit and Saitoh 2017; Fuhl et al. 2018a,b, 2016a, 2017b; Kan et al. 2018; Vera-Olmos and Malpica 2017; Vera-Olmos et al. 2019; Zhu et al. 2018]. The key issue with data-driven approaches is the requirement of *large amounts of labeled data*, which is a tedious, costly, slow, and error-prone manual process. Alternatively, previous work attempted to generate large amounts of data by using a reverse calibration approach[1] in combination with a refinement step [Tonsen et al. 2016]. This approach requires the subjects to be put under the specific challenges of interest, and the resulting labels are of lower quality than human-annotated ones. Wood et al. [Wood et al. 2015] proposed synthesizing perfectly labeled photo-realistic eye images for training data generation. Using a collection of dynamic eye-region models obtained from head scans they generated eye images to simulate various head poses, gaze directions, and illumination conditions. However, training on synthetic images did not result in a high inference performance due to differences between synthetic and real eye image distributions [Kan et al. 2018]. Kan et al. also found that simple generic affine transformation for data augmentation on real eye images lead to a better learning rate when compared to synthetic eye images. In this work, we:

[1] Estimating the pupil position from an automatically detected target in the field camera.

- propose multiple domain-specific data augmentation methods to enhance labeled data to create unique training sets containing desirable distributions of eye-tracking challenges (e.g., low contrast, reflections),
- demonstrate that by utilizing data recorded in constrained scenarios where state-of-the-art pupil trackers already perform virtually without errors, it is possible to generate a foundational data set with high-quality subpixel accuracy labels,
- combine the augmentation techniques and foundational data set to create labeled data sets with controlled distributions of eye-tracking challenges,
- demonstrate the feasibility and convenience of the proposed approach by training a CNN solely with the foundational data set images augmented on the fly. The resulting CNN outperformed the state-of-the-art for pupil detection in multiple publicly-available, realistic, and challenging data sets, despite being trained solely with the augmented images, displaying better generalization capabilities across all evaluated data sets.

2 TRAINING DATA GENERATION

2.1 Foundational Data Set

To construct a reliable and accurately labeled foundational data set, we propose employing existing state-of-the-art pupil detection methods paired with unchallenging data sets. Similarly to the work from [Tonsen et al. 2016], this also implies involving participants to record data. However, with our approach a) these recordings can be done virtually in any simple environment devoid of additional infrared illuminations, and b) no calibration or target following is required. Consequently, this translates into an additional advantage: Allowing one to reuse already recorded data that meets the unchallenging criteria. Thus, as base for our data set, we could simply take advantage of data already available from previous studies in a laboratory [Santini et al. 2016], supermarkets [Kasneci et al. 2014a; Sippel et al. 2014], museum [Santini et al. 2018a], and driving [Kasneci et al. 2014b; Kübler et al. 2015][2]. These data were collected using multiple eye trackers: Dikablis 1, Dikablis 2, and a Pupil Labs (with the second eye camera version – Pupil Cam2). We visually inspected these recordings and selected those that met our unchallenging criteria and are not part of existing data sets for pupil detection evaluation employed in Section 4. The selected recordings include participants with glasses, contact lenses, physiological anomalies, droopy eye lids, and bad eye camera position for which PuReST was still able to detected the pupil outline properly. For recordings containing binocular data, we used videos from both eyes as they represent distinct geometric configurations and appearances. In total, we selected more than 400 distinct eye recordings. The labeling method used PuReST [Santini et al. 2018c] to automatically detect the pupil center and outline for each frame in each recording. If the detected pupil had a confidence over 0.66 (following the threshold suggested by Santini et al. [Santini et al. 2018b]), it was included in the foundational data set, together with the pupil parameters as labels. Per recording, this was repeated until a thousand images were extracted or the video terminated, resulting in over 400 thousand automatically-labeled near-infrared eye images. The images were normalized to 192 px^2, randomly cropping patches away from the labeled pupil to maintain the original image aspect ratio if necessary.

[2]Part of these data are already publicly available, and the remaining ones were attained by contacting the authors of the respective papers.

2.2 Generic and Domain-Specific Data Augmentation

Multiple generic data augmentation techniques can be applied in the context of head-mounted eye tracking although their intensities should be constrained to match physical constrains. Examples of these augmentations are *affine transformations*, *blurring*, and *exposure level adaptation*. In this work, we additionally propose three domain-specific augmentations: mock-up glints, mock-up pupils, and reflections. In our data generation framework, most augmentation techniques are parameterized by 1) a *probability* (P), which defines how often the technique is applied, as well as 2) minimum (min) and maximum (max) magnitude parameters, controlling to which extent the technique is applied. These parameters allow one to fine tune the generated data to distinct distributions of challenges-of-interest. For the experiments in this work, we set all parameters empirically based on previous experience with head-mounted eye tracking data and visual inspection of generated data. These are reported in Table 1.

Table 1: Employed augmentation parameter set describing the distribution of challenges. Parameters marked with N/A are not applicable to the associated technique.

Technique	P	max	min
Downscale (Affine)	0.75	0.95	0.5
Crop (Affine)	0.5	0.95	0.5
Horizontal Flip (Affine)	0.5	N/A	N/A
Blur	0.25	N/A	N/A
Exposure	0.25	1.2	0.7
Mock-up Glints	0.5	0.25	0.05
Mock-up Pupil	0.25	N/A	N/A
Reflections	0.75	0.75	0.25

Affine transformations are employed to augment the images from a geometric perspective – i.e., distinct camera positions. Given the wide range of distinct geometric positions in the eye recordings already available in our foundational data set, we have limited affine transformations to downscaling, cropping, and horizontal flipping. The first two are used to distribute pupil sizes across the generated augmented data, whereas the last one balances the data between left and right eye cameras. Additionally, cropping (achieved by combining upscaling with translations) assists in distributing pupil position w.r.t. image more evenly.

Blurring is employed to augment the images from a camera focus perspective – i.e., to emulate effects resulting from recording with a fixed camera depth at different distances. We employ a Gaussian blur parameterized by a Gaussian kernel of size ks in the range [3, 9] and standard deviation (σ) in the range [0.25, 0.75] based on empirical tuning.

Exposure level adaptations augment the images from a camera exposure perspective – i.e., to emulate the amount of light received by the imaging sensor, which is usually controlled by shutter speed and illumination strength. This is achieved by multiplying all image pixels intensities by a certain factor.

Mock-up glints are used to deter the network from using glint position to infer pupil position. On a first glance, one might advocate that using the glint can improve network inference capabilities. Whereas this holds for a unique eye tracking device, distinct eye trackers produce different glint patterns. Thus, this can deteriorate network generalization. In our framework, we add up to six mock-up glints as high-intensity pixel blobs.

Mock-up pupils are utilized to guide the network to learn more global rather than local features. Learning relations between pupil position and, e.g., eyelids or the iris, is enforced this way. For instance, a network that learns to search for the pupil as a dark blob might easily be fooled by other dark blobs in the image such as those resulting from makeup, eye lashes, or shadows. This is achieved by creating pupil-like objects around the input image.

Reflections are one of the most challenging issues with state-of-the-art eye trackers [Fuhl et al. 2016c]. In order to achieve realistic reflections, we opted to superimpose real reflections. We recorded a set of more than fourteen thousand reflection frames while moving through indoors and outdoor environments using the device shown in Figure 1. For the outdoor environments, frames were recorded in

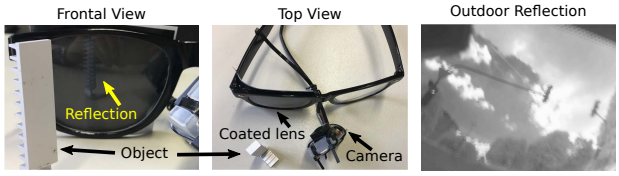

Figure 1: The device used to record reflections consisted of a store-bought reading glasses covered on one side by an ultra-pigmented black acrylic paint. An infrared camera was attached to the eyeglasses to capture images reflecting on the lenses. In practice, this results in images from a specular mirror from which reflection masks can be extracted.

several sessions on sunny and cloudy days while walking through a) coarsely and densely forested areas, b) human-constructed landscapes, and c) driving through the city. Considering each captured frame as an 8-bit grayscale image, superimposition is achieved by combining the input eye image (E) pixel (e) and reflection image (R) pixel (r) pixel-wise to produce the output image (O) pixel (o) as

$$o = e + \frac{\gamma r(255 - e)}{255}, \tag{1}$$

where

$$\gamma = max\left(min\left(0, \frac{r}{255} + \frac{e}{255} - \alpha\right), \beta\right). \tag{2}$$

α and β are hyperparameters weighting how much dark pixels in the input eye image should be affected by the reflections. These parameters were fixed to $\alpha = 0.5$ and $\beta = 0.75$ empirically, but examples for distinct values are shown in Figure 2.

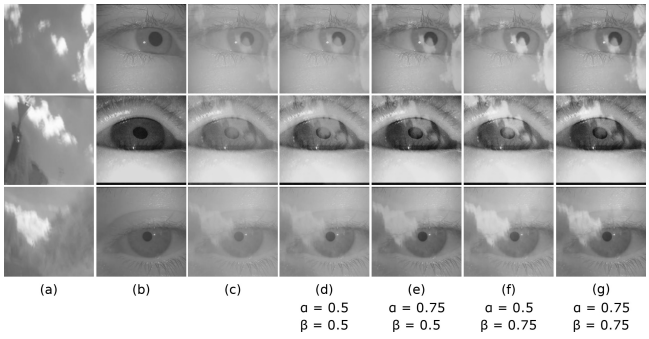

Figure 2: (a) Input reflection image (R), (b) input eye image (E), (c) superimposition with fixed $\gamma = 1$, and (d)–(g) resulting superimpositioned images for different values of α and β.

3 EMPLOYED NETWORK

Given that this network is merely used to demonstrate the feasibility of our proposed approach, we only give a brief description of its model and training here. We expect other networks to be able to attain similar performances when trained using data generated by our method. We employed an hybrid model based on Inception V4 [Szegedy et al. 2016], YOLO [Redmon and Farhadi 2017], and network in network (NIN) [Lin et al. 2013] models to achieve a high detection rate while maintaining low latency for real-time eye tracking. The resulting architecture is shown in Figure 4 and regresses the pupil center, width, and height.

The network core is similar to the Inception network. We made the network customizable by employing a) repeatable blocks and b) a coefficient for the number of filters in each convolutional layer. By default, the reduction blocks were applied between different block types. For example, when we used block A and B for our network, we have to use reduction block A in-between and respectively reduction-block B between B and C blocks. The final convolutional layer employs global average pooling (GAP) to reduce overfitting. Moreover, a 1×1 filter was applied to predict the bounding box parameters. All weights were initialized with the Xavier [Glorot and Bengio 2010] initializer, and L2 weight decay was employed to diminish over fitting. Furthermore, we used a dropout layer before the GAP layer. By default, all convolutional layers were utilized with batch normalization. Since the labels were generated automatically, we expect at least a small portion of the labels to be faulty. Thus, we employed Huber loss as cost function since it is less sensitive to such outliers. Adam optimizer was used for backpropagation with the learning rate starting at 0.001, scheduled to decrease every five epochs. To prevent gradient explosion, we used maximum gradient norm. During training step, we divided the data into the training and validation set with a ratio 90/10, respectively.

4 RESULTS

We measure pupil detection error by calculating the Euclidean distance between the predicted pupil center and the annotation. Similar to other studies [Fuhl et al. 2016a, 2017b, 2016b; Santini et al. 2018b; Vera-Olmos et al. 2019], detection with an error lower than

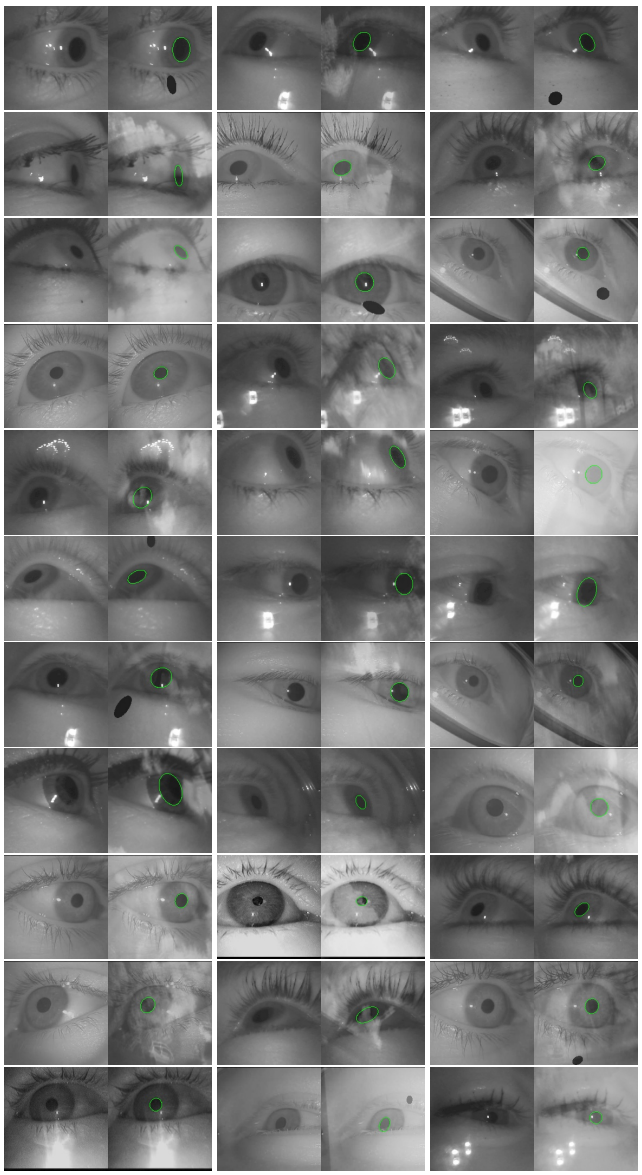

Figure 3: Example of pairs consisting of input images from non-challenging recordings (left) and the resulting automatically labeled and augmented image (right). The label is shown in green and better visualized in digital form. Our approach allows researchers to quickly generate arbitrarily large data sets with controlled distribution of eye-tracking challenges for training eye image related tasks such as pupil detection. For the sake of visualization, we have disabled affine transformations for these examples.

5 pixels was considered to be successful for detection rate estimation. For this evaluation, we have employed five publicly available data sets, namely Świrski [Świrski et al. 2012], ExCuSe [Fuhl et al. 2015], ElSe [Fuhl et al. 2016b], LPW [Tonsen et al. 2016], and PupilNet [Fuhl et al. 2016a].

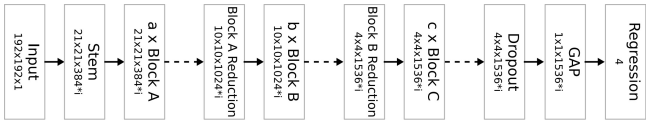

Figure 4: General architecture of the proposed hybrid model. Also, in our experiments, we found that dropout before GAP improve the results by 1.8%. The final change is the regression layer with Huber loss.

In our experiments, we found that a model with 3 block A and 4 blocks B with half of filters number (3A4BH) performs best in terms of detection rate and speed on the training data and, thus, select it as our final network. We compare our network with two algorithmic pupil detectors (ElSe [Fuhl et al. 2016b] and PuRe [Santini et al. 2018b]) and one pupil tracker (PuReST [Santini et al. 2018c]). These algorithms were chosen as they represent the state-of-the-art in pupil tracking, have been shown to significantly outperform competing commercial approaches (e.g., Pupil Labs) as well as other academic works [Santini et al. 2018c]. As representative of machine-learning approaches, we employ the first work to apply CNNs for pupil detection (PupilNet [Fuhl et al. 2017b]) as well as the most recent one (DeepEye [Vera-Olmos et al. 2019]).

PupilNet and DeepEye used the ElSe and ExCuSe data sets for training and testing (with cross-validation for evaluation). All of these data sets (as well as PupilNet's) were recorded using a Dikablis 1 eye tracker and are, thus, similar to some extent. Therefore, we have reasons to believe that these methods might have overfit to this specific type of data. It is worth noting that part of our foundational data set also includes data from this eye tracker. Results for these data sets are reported in Table 2. On average, our network outperforms algorithmic methods and PupilNet by at least 7.89 percentual points. Relative to the most recent CNN (DeepEye), our network produces similar results – losing by a difference of less than one percentual point on average. In fact, DeepEye was the best performer in 15 of the data sets, where as ours was the best performer in 12.

To test the hypothesis that DeepEye has overfit to the particular type of data of its training data sets, we have also evaluated it using the remaining two data sets (Świrski and LPW), which were recorded with distinct eye trackers from the ExCuSe, ElSe, and PupilNet data sets. We also included the best algorithmic detector (PuRe) and the tracker (PuReST) in this evaluation. *It is worth reiterating that neither DeepEye nor our network has seem data from these eye trackers before.* Results are shown in Table 3 and clearly demonstrate that our network has better generalization capabilities, overperforming DeepEye on average by 34 and 20 percentual points for the LPW and Świrski data sets, respectively. Moreover, detection rate for LPW was particularly similar to the one in the ExCuSe, ElSe, and PupilNet data sets, despite the large difference in appearance between the data sets, indicating that our network has learned eye tracker independent features. On the contrary, not only the performance dropped for the Świrski data sets, but both PuRe and PuReST have outperformed the machine-learning based approaches. This can be explained by the nature of that data set, which is composed by a small set of highly-off-axis images, which

Table 2: Detection rate (%) results considering a 5 px threshold. We report the performance reported by the original publications whenever possible. DeepEye performance for PN 1 to PN V were evaluated using the implementation provided by the authors (Available at github.com/Fjaviervera/DeepEye).

Data Set	Detection Rate (%)					
	ElSe	PuRe	PuReST	PupilNet	DeepEye	Ours
I	86	87	89	82	86	**90**
II	65	29	48	79	83	**88**
III	64	73	77	66	**93**	88
IV	83	89	90	92	**93**	**93**
V	85	87	87	92	**97**	**97**
VI	78	89	91	79	93	**94**
VII	60	68	73	73	**84**	81
VIII	68	54	60	81	87	**89**
IX	87	91	91	86	**92**	91
X	79	90	90	81	92	**94**
XI	75	88	87	91	94	**97**
XII	79	**88**	**88**	85	85	**88**
XIII	74	**85**	80	83	79	83
XIV	84	88	89	95	96	**97**
XV	57	62	71	81	**89**	77
XVI	60	79	66	80	82	**87**
XVII	90	95	95	**97**	95	96
XVIII	57	68	69	62	74	**75**
XIX	33	48	53	37	**78**	46
XX	78	83	86	79	**92**	89
XXI	47	70	81	83	88	**89**
XXII	53	62	72	58	**80**	76
XXIII	94	97	93	90	96	**100**
XXIV	53	60	65	55	55	**73**
PN I	62	**87**	72	69	64	83
PN II	26	29	48	45	**83**	60
PN III	39	73	57	49	**79**	69
PN IV	54	89	81	82	**90**	88
PN V	75	**87**	83	81	83	82
Average	67.6	75.7	76.96	76.3	85.6	84.8

are very unusual for practical eye tracking as only a small pupil range is visible. Thus, such data is under represented in our foundational data set, and, consequently, the network has not learned these highly-off-axial' parameter distribution adequatly.

Table 3: Detection rate of the proposed and DeepEye [Vera-Olmos et al. 2019] models on previously unseen LPW [Tonsen et al. 2016] and Swirski [Świrski et al. 2012] data sets.

Data Set	Detection Rate (%)			
	PuRe	PuReST	DeepEye	Ours
LPW	75	82	50	**84**
Swirski	80	**86**	54	74

5 CONCLUSION

Over four years now, researchers have demonstrated the potential of CNN-based techniques to provide reliable pupil detection. The quality and quantity of training data remarkably affects the performance of these techniques. Models trained earlier [Fuhl et al. 2016a, 2017b; Vera-Olmos et al. 2019] usually use the cross-validation technique with training and testing data coming from the same data set and, therefore, from a specific distribution. Therefore, these methods are quite specific to recording devices and, even worse, to the recording scenarios.

In this study, we demonstrate the feasibility of training a competitive network using automatically labeled data in combination with the proposed generic and domain-specific augmentation techniques. Importantly, we can demonstrate a high generalization performance to various data sets, devices and scenarios, unlike previous work. This implies that our training data contained sufficient variance to allow the model to fit parameters for the general pupil detection task, without substantial over-fitting to specific device properties, illumination conditions or scenarios. As a result we surpass state-of-the-art CNN-based pupil detection for unseen data (LPW and Swirski, see Table 3) by an average of 27%. This suggest that the proposed combination of easy-to-gather and automatically labeled data represent a significant step in achieving better pupil detection networks in the future. Furthermore, this approach can similarly be applied to other eye features such as the eyelids [Fuhl et al. 2017a] or iris [Abate et al. 2015]. Additionally, the trained CNN is real-time capable, achieving 34 and 124 FPS on a CPU (Intel i7-7700, 16GB RAM) and GPU (Nvidia GTX 1070), respectively (estimated based on network performance across more than four minutes of eye videos). In contrast, the most recent CNN-based approach (DeepEye) is limited to ≈30 FPS [Vera-Olmos et al. 2019].

To increase the detection rate further, we plan to extend our approach by the use of tracking techniques. Pupil tracking has shown remarkable ability to stabilize pupil detection and to solve hard tracking conditions for traditional computer vision approaches [Santini et al. 2018c] and could be incorporated in the deep learning approaches e.g. via Recurrent Neural Networks (RNNs).

ACKNOWLEDGMENTS

Work of the first and forth author is supported by the Institutional Strategy of the University of Tübingen (Deutsche Forschungsgemeinschaft, ZUK 63).

REFERENCES

Andrea F Abate, Maria Frucci, Chiara Galdi, and Daniel Riccio. 2015. BIRD: Watershed based iris detection for mobile devices. *Pattern Recognition Letters* 57 (2015), 43–51.

Reuben Aronson et al. 2018. Eye-Hand Behavior in Human-Robot Shared Manipulation. In *Proceedings of the 13th Annual ACM/IEEE International Conference on Human Robot Interaction (To appear)*.

Warapon Chinsatit and Takeshi Saitoh. 2017. CNN-Based Pupil Center Detection for Wearable Gaze Estimation System. *Applied Computational Intelligence and Soft Computing* 2017 (2017).

Wolfgang Fuhl, Shahram Eivazi, Benedikt Hosp, Anna Eivazi, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2018a. 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*. ACM, 48.

Wolfgang Fuhl, David Geisler, Thiago Santini, Tobias Appel, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2018b. CBF: circular binary features for robust and real-time pupil center detection. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ACM, 8.

Wolfgang Fuhl, Thomas Kübler, Katrin Sippel, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2015. ExCuSe: Robust Pupil Detection in Real-World Scenarios. In *Computer Analysis of Images and Patterns*, George Azzopardi and Nicolai Petkov (Eds.). Springer International Publishing, Cham, 39–51.

Wolfgang Fuhl, Thiago Santini, and Enkelejda Kasneci. 2017a. Fast and robust eyelid outline and aperture detection in real-world scenarios. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 1089–1097.

Wolfgang Fuhl, Thiago Santini, Gjergji Kasneci, and Enkelejda Kasneci. 2016a. PupilNet: Convolutional Neural Networks for Robust Pupil Detection. *CoRR* abs/1601.04902 (2016). arXiv:1601.04902 http://arxiv.org/abs/1601.04902

Wolfgang Fuhl, Thiago Santini, Gjergji Kasneci, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2017b. PupilNet v2.0: Convolutional Neural Networks for CPU based real time Robust Pupil Detection. *CoRR* abs/1711.00112 (2017). arXiv:1711.00112 http://arxiv.org/abs/1711.00112

Wolfgang Fuhl, Thiago C Santini, Thomas Kübler, and Enkelejda Kasneci. 2016b. Else: Ellipse selection for robust pupil detection in real-world environments. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. ACM, 123–130.

Wolfgang Fuhl, Marc Tonsen, Andreas Bulling, and Enkelejda Kasneci. 2016c. Pupil Detection for Head-mounted Eye Tracking in the Wild: An Evaluation of the State of the Art. *Mach. Vision Appl.* 27, 8 (Nov. 2016), 1275–1288. https://doi.org/10.1007/s00138-016-0776-4

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

Naoyuki Kan, Nagisa Kondo, Warapon Chinsatit, and Takeshi Saitoh. 2018. Effectiveness of Data Augmentation for CNN-Based Pupil Center Point Detection. In *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. IEEE, 41–46.

Enkelejda Kasneci et al. 2014a. Homonymous Visual Field Loss and Its Impact on Visual Exploration: A Supermarket Study. *Translational vision science & technology* 3, 6 (2014), 2–2.

Enkelejda Kasneci, Katrin Sippel, Kathrin Aehling, Martin Heister, Wolfgang Rosenstiel, Ulrich Schiefer, and Elena Papageorgiou. 2014b. Driving with binocular visual field loss? A study on a supervised on-road parcours with simultaneous eye and head tracking. *PloS one* 9, 2 (2014), e87470.

Thomas C Kübler et al. 2015. Driving with glaucoma: task performance and gaze movements. *Optometry & Vision Science* 92, 11 (2015), 1037–1046.

Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

Joseph Redmon and Ali Farhadi. 2017. YOLO9000: better, faster, stronger. *arXiv preprint* (2017).

Thiago Santini et al. 2018a. The Art of Pervasive Eye Tracking: Unconstrained Eye Tracking in the Austrian Gallery Belvedere. In *Proceedings of the 2018 ACM Eye Tracking Methods and Applications : Adjunct (PETMEI)*. ACM.

Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. 2018b. PuRe: Robust pupil detection for real-time pervasive eye tracking. *Computer Vision and Image Understanding* (2018).

Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. 2018c. PuReST: Robust Pupil Tracking for Real-time Pervasive Eye Tracking. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA '18)*. ACM, New York, NY, USA, Article 61, 5 pages. https://doi.org/10.1145/3204493.3204578

Thiago Santini, Wolfgang Fuhl, Thomas Kübler, and Enkelejda Kasneci. 2016. Bayesian identification of fixations, saccades, and smooth pursuits. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*. ACM, 163–170.

Jürgen Schmidt et al. 2017. Eye blink detection for different driver states in conditionally automated driving and manual driving using EOG and a driver camera. *Behavior Research Methods* (2017), 1–14.

Katrin Sippel, Enkelejda Kasneci, Kathrin Aehling, Martin Heister, Wolfgang Rosenstiel, Ulrich Schiefer, and Elena Papageorgiou. 2014. Binocular glaucomatous visual field loss and its impact on visual exploration-a supermarket study. *PloS one* 9, 8 (2014), e106089.

Yusuke Sugano and Andreas Bulling. 2015. Self-calibrating head-mounted eye trackers using egocentric visual saliency. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 363–372.

Lech Świrski, Andreas Bulling, and Neil Dodgson. 2012. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 173–176.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

Tony Tien et al. 2015. Differences in gaze behaviour of expert and junior surgeons performing open inguinal hernia repair. *Surgical endoscopy* 29, 2 (2015), 405–413.

Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2016. 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*. ACM, 139–142.

FJ Vera-Olmos and N Malpica. 2017. Deconvolutional neural network for pupil detection in real-world environments. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer, 223–231.

FJ Vera-Olmos, E Pardo, H Melero, and N Malpica. 2019. DeepEye: Deep convolutional network for pupil detection in real environments. *Integrated Computer-Aided Engineering* 26, 1 (2019), 85–95.

Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. 2015. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 3756–3764.

Joanne M Wood, Richard A Tyrrell, Philippe Lacherez, and Alex A Black. 2017. Nighttime pedestrian conspicuity: effects of clothing on driversâĂŹ eye movements. *Ophthalmic and physiological optics* 37, 2 (2017), 184–190.

Yinheng Zhu, Wanli Chen, Xun Zhan, Zonglin Guo, Hongjian Shi, and Ian G Harris. 2018. Head Mounted Pupil Tracking Using Convolutional Neural Network. *arXiv preprint arXiv:1805.00311* (2018).