

# 1000 Pupil Segmentations in a Second using Haar Like Features and Statistical Learning

Wolfgang Fuhl  
University Tübingen  
Sand 14, Tübingen, Germany  
wolfgang.fuhl@uni-tuebingen.de

Johannes Schneider  
University of Liechtenstein  
Fürst-Franz-Josef-Strasse, 9490 Vaduz, Liechtenstein  
johannes.schneider@uni.li

Enkelejda Kasneci  
University Tübingen  
Sand 14, Tübingen, Germany  
Enkelejda.Kasneci@uni-tuebingen.de

## Abstract

*In this paper we present a new approach for pupil segmentation. It can be computed and trained very efficiently, making it ideal for online use for high speed eye trackers as well as for energy saving pupil detection in mobile eye tracking. The approach is inspired by the BORE and CBF algorithms and generalizes the binary comparison by Haar features. Since these features are intrinsically very susceptible to noise and fluctuating light conditions, we combine them with conditional pupil shape probabilities. In addition, we also rank each feature according to its importance in determining the pupil shape. Another advantage of our method is the use of statistical learning, which is very efficient and can even be used online. <https://atreus.informatik.uni-tuebingen.de/seaf/file/d/8e2ab8c3fdd444e1a135/?p=%2FStatsPupil&mode=list>.*

## 1. Introduction

The plethora of image based eye tracking [6] applications has continued to rise in recent years. The most important areas of application are currently driver monitoring [2], virtual reality [43], augmented reality [47], medicine [8, 7, 1], market research [43, 49], remote support [51], human computer interaction [9, 39], supportive explanation models for computer vision models [60], and many more.

Those diverse application areas bring different image based challenges [26, 19, 28, 27, 18] and challenging resource restrictions [12, 15, 10] as well as different types of gaze estimation like appearance based approaches [61, 62].

Some of these image based challenges are changing illumination conditions, reflections on glasses, make up, recording errors, and high off axial pupil positions. In addition, the diversity of people using eye tracking devices also rise new challenges like deformed pupils [26, 19, 25, 24] which occurs after eye surgery [5].

Other challenges in eye tracking are different recording techniques like RGB and NIR imaging. While NIR is mostly used in head mounted [28] eye trackers, RGB imaging is still used in remote [16] eye tracking and especially web cam based eye tracking [48]. Due to the current situation with the Covid pandemic, web cam based eye tracking becomes more and more important for market research [48, 43] and scientific studies [48].

Nowadays, the gaze signal alone is also no longer sufficient, as the eye provides a variety of other sources of information. These are pupil response to cognitive load [4], pupil shape for eyeball regression [11], and eyelids to determine a person's fatigue [22, 21, 23]. The cognitive load is very interesting for the detection of mental disorders [46] or ranking a persons performance capability [40]. Eye ball regression is used to improve the robustness of eye trackers against drifts of the device and to improve the accuracy [52]. The fatigue detection of a person is important for critical applications like driving [58], flying [45], flight surveillance [42], and many more.

Due to the progress in eye tracker technology so far, mobile applications [37], long-term studies [55], high speed eye tracking for fundamental research [31], and the consumer market such as computer games [35] are becoming more and more important. For this it is necessary that the algorithms can be used as resource saving and robust as possible [15, 10, 12] to consume as little energy as possible in

mobile applications [37], to guarantee the real-time capability in high speed eye tracking [31], and not to waste computing capacity which is needed for computer games [44].

In this paper we present a resource-saving approach, which is inspired by CBF [15] and BORE [10]. Our approach was developed with the main features of cheap execution and easy training. The features used are Haar features [56] which can be computed very efficiently. Another important feature of our algorithm is the possibility of on-line learning and therefore, the usage together with deep neural networks (DNN). The combination with DNNs reduces the resource consumption and detection time of the pupil ellipse.

Contributions of this work to the state of the art are:

- Our approach is the first to use ellipse parameter conditional probability distributions for ellipse selection. This avoids unnecessary checks of ellipse points, as is the case in CBF [15] and BORE [10].
- Compared to BORE [10] and CBF [15] we use index tables whereby each feature has to be evaluated only once. This further reduces resource consumption and, in combination with the precomputed indexes already presented in CBF [15] and BORE [10], further speeds up the detection process.
- Our approach is simply trained on the occurrence statistics. This procedure is much more resource efficient than the unsupervised learning and evaluation of all possible combinations as done in BORE [10]. It is also much faster than the random combination evaluation used in CBF [15].
- Our approach can be trained online and therefore be used to speed up the pupil detection of large DNNs.

## 2. Related work

Since pupil detection has evolved in different directions, we divide the related work into three areas. These are classical computer vision approaches, deep neural networks, and resource saving machine learning approaches.

### 2.1. Classical Computer Vision Approaches

In the field of pupil detection and exact pupil center determination, the first major breakthrough came with the use of edge images [53]. Previously, adaptive thresholds were used [29]. A major disadvantage of edge images are their susceptibility to noise and motion blur. Therefore, edge filtering methods were introduced [26, 19, 27], which suppress noise and pass only relevant edge segments. In addition to this, angular integral projection function [19] and also blob detection [26] were used. Another improvement in pupil shape reconstruction was the evaluation of individual segments [34]. Alternative to edge detection, the radial symmetry transform was used to detect the pupil center [41].

### 2.2. Deep Neural Networks

With the advent of convolutions in neural networks and the success in the field of image processing, these CNNs were also used for pupil detection and segmentation. The first window-based approach was PupilNet [24, 25], running in real time on a single CPU core only. Later, large residual networks were also used [20] and published together with huge annotated datasets as well as generative adversarial networks [13] were used. The first U-Net with interconnections was proposed with DeepVOG [59]. New loss formulations regarding the pupil shape were suggested in [3]. Additional to those loss formulations an L1 loss connected to the central part of a fully connected convolutional network was proposed in [36].

### 2.3. Resource Saving Machine Learning

The first real-time machine learning methods combined with simple features were introduced by PupilNet [24, 25] and continued BORE [10] and CBF [15]. BORE [10] is capable of non-supervised learning and self-optimization. CBF [15], on the other hand, uses random ferns and pixel comparisons to determine the center of the pupil. Also the supervised decent method (SDM) was used for the regression of the pupil center in remote images [38]. However, this has the disadvantage of being highly dependent on the mean shape, which we will show in our later evaluation in combination with landmarks for segmentation on head mounted images. Another approach was created using the teacher, student training method [12, 17]. These tiny CNNs [12, 17] are very robust and were successfully used across datasets. In addition, they learn to evaluate the accuracy and give therefore a validity of the pupil ellipse [17].

The approach presented by us is also to be classified in this category. This is due on the one hand to the fact that our approach uses statistical learning and on the other hand to the resource-saving use of our method. In addition to these properties, our approach can also be trained very fast and resource-saving.

## 3. Method

Our approach uses statistical learning which needs to inspect each training sample only three times to create a detector. The first step for training our detector is to create the search area ( $x, y$  tuples) and the ellipses expected there ( $el_i$ ). This gives us the set  $EL$  which stores all ellipses  $el_i$  for each position  $x, y$ . Reformulated as a conditional probability distribution  $EL$  corresponds to the probability  $P$  of ellipse  $el_i$  under the condition to be centered at position  $x, y$  and thus Equation 1.

$$EL = P(el_i|x, y) \quad (1)$$

To calculate  $EL$ , we need one pass of the training data.

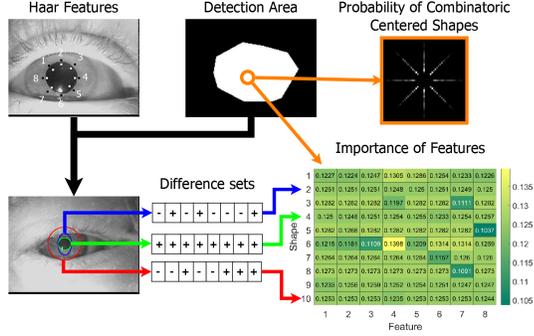


Figure 1. We introduce tiny Haar features which follow elliptical shapes for pupil segmentation. Together with a detection area, a shape conditioned probability distribution as well as statistical feature weighting.

In the second step, we reduce EL to speed up our detector and reduce over fitting. For this, we represent each ellipse as eight landmarks (See Figure 1) and round them to integers. For reduction, all ellipses with the same landmark distances are combined into one ellipse with a maximum deviation of one pixel per landmark. This gives us the reduced conditional probability distribution  $\widehat{EL}$ .

The next step is to create our feature extractors from the landmarks. For this we use Haar features. Instead of computing the area differences in the integral images, we use the difference of pixels in downscaled images. In the second pass of the training set, for each ellipse in  $\widehat{EL}$  we store all occurrences of the eight differences  $d_j$ . Since the set of eight differences to each ellipse is very large, we want to reduce it. To reduce this set, we compute the best five difference sets  $d_j$  for each positive probability in  $\widehat{EL}$ , noting here that one  $d_j$  are eight differences one for each landmark. For this we use the mean shift clustering with a maximum of five clusters. Reformulated as a conditional probability distribution, we thus have positive probabilities for five difference sets  $d_j$  under the condition of the probability of an ellipse at some position ( $\widehat{EL} = P(\widehat{el}_i|x, y)$ ) and thus Equation 2.

$$D = (d_j|P(\widehat{el}_i|x, y)) \quad (2)$$

In the last run, the individual differences or landmarks are now weighted with respect to their robustness. These feature weights  $f_j$  are computed in the third run of the training set. For this we use the difference set  $d_j$  with the minimum distance of the landmark differences and weight  $f_j$  positively if the sign matches and negatively if the sign differs. Based on this we statistically weight the reliability of each feature. After the pass each feature  $f_j$  is normalized to sum to one and to form a probability distribution. This gives us a similar conditional probability distribution as for the difference sets and thus Equation 3.

$$F = (f_j|P(\widehat{el}_i|x, y)) \quad (3)$$

To use the detector, the landmark differences to the ellipses must be calculated at all possible positions. Then the minimum difference to the difference sets  $d_j$  is calculated and the deviation is weighted by the feature weights  $f_j$ . The final ellipse and position is then the global minimum and described in Equation 4.

$$argmin_{P(\widehat{el}_i|x, y)} (argmin_j \sum_{LM=1}^8 abs(d_j(LM) - \tilde{d}(LM)) * f_j(LM)) \quad (4)$$

In Equation 4  $LM$  are the Haar features,  $d_j$  is the difference set,  $\tilde{d}$  is the set of differences in the input image for ellipse  $\widehat{el}_i$  at position  $x, y$  and  $f_j$  is the corresponding feature weight. Overall Equation 4 searches for the minimum difference of the eight landmark positions in the entire input image. If there are multiple equally good positions we use the conditional probability distribution  $\widehat{EL}$  to select the most probable ellipse and position. While this already provides a very efficient detector, further optimizations are necessary like the precalculation of all indexes in the image, as it was already presented for BORE [10] and CBF [15]. Also, all differences are indexed to calculate differences at each position only once.

Since these conditional probabilities and occurrence statistics are easily extensible as well as online data can be added, this learning approach is able to learn in parallel to other approaches like DNNs. Thus, it is possible to use a DNN for detection and learn the statistics online. Once our approach has learned some examples, it can be used before the DNN, saving resources, energie, and computation time. To update our statistics online we weighted new data based on the number of data already used to compute the statistics. This is similar to the use of moving averages but with the simplification that we only have to remember how many data samples were used to compute the current statistical values.

#### 4. Evaluation of our approach trained offline

The data used for the offline training evaluation are the segmented pupils of [20]. The dataset consists of two files "p1\_image.mp4" and "p2\_image.mp4" with an image resolution of  $192 \times 144$ . The first file contains data taken in a driving simulator and the second file contains images from real world driving. Because of this there are no reflections or strong light fluctuations in the data of the first file, so it contains much simpler images. Therefore, we decided to use the first file with more than 500,000 frames as the training dataset. The second file with more than 350.000 frames

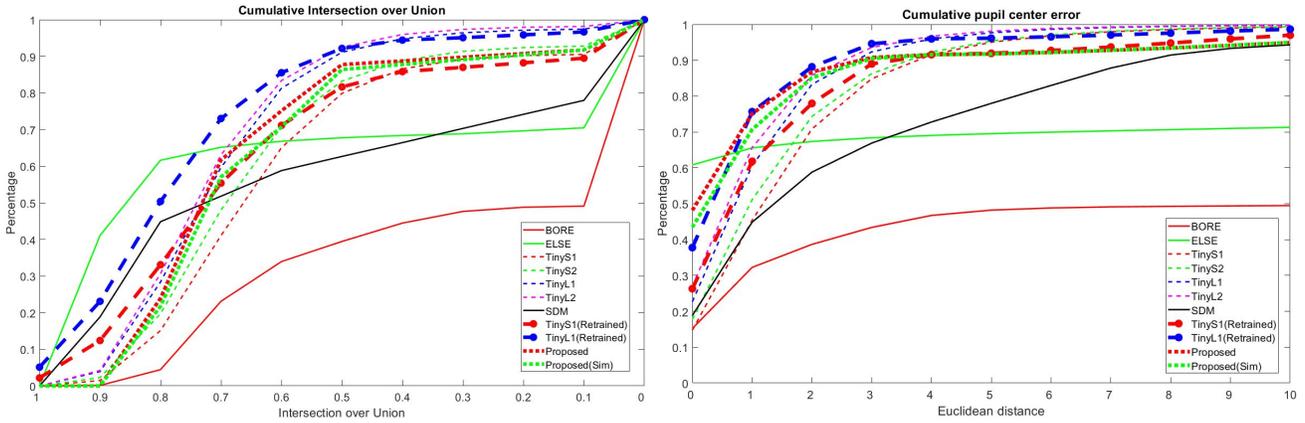


Figure 2. The cumulative intersection over union results for multiple approaches on the left. On the right is the cumulative euclidean pupil distance error for multiple approaches.

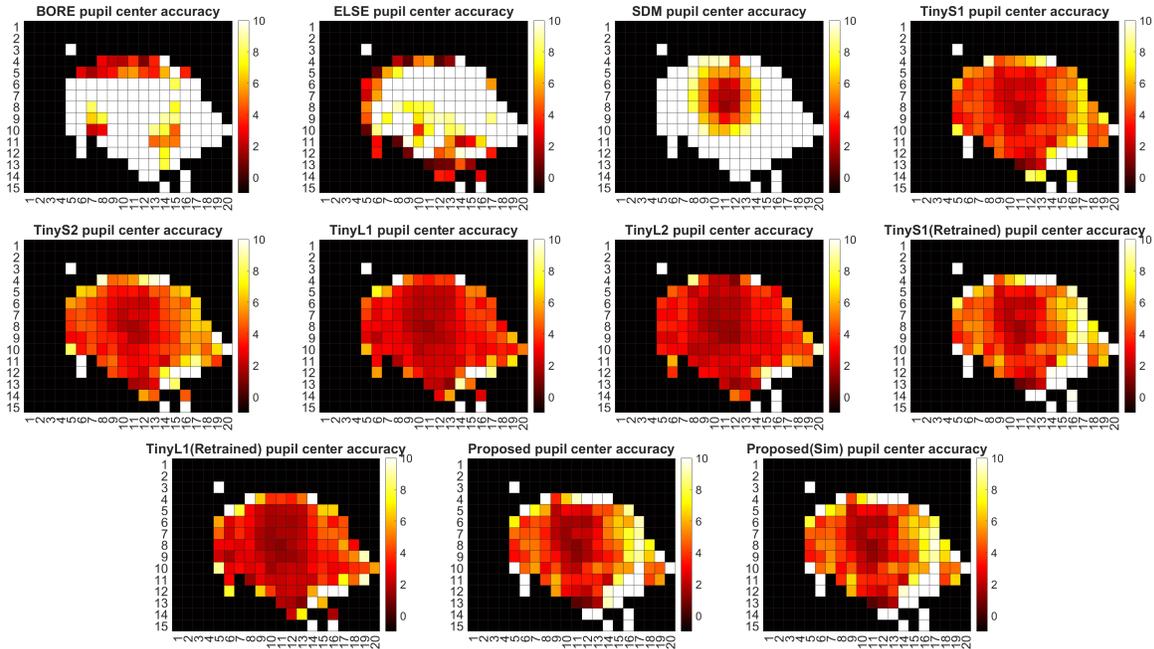


Figure 3. The mean euclidean pixel distance mapped to  $10 \times 10$  cells on the image space. Higher values or brighter colors are a worse results. We have limited the maximum error to 10 pixels for a uniform display of the color scaling.

and the much more challenging images is used as the evaluation dataset.

For the data augmentation, we used up to 20% random noise, as well as reflections with an intensity up to 20%, where the reflections are calculated from randomly selected images. Also, we randomly changed the contrast of the image in the range of -40 to 40. In addition, we shifted the image randomly in a range of -10 to 10 pixels as well as we used zooming with a random factor in the range of 0.8 to 1.2. For the TinyCNNs [12], this was done online during training. For all other approaches, the data augmentation was computed in advance resulting in five images from each frame. Of course, the image could also occur without

augmentation.

Since we trained our approach once with the real data ("p1\_image.mp4" for the driving simulator recordings) and once with the simulator data, we also give here the details for our approach. First, we used the simulator of [11] and inverted the images so that the pupil is dark. Then we selected the data based on the pupil ellipse, which matched the pupil ellipses in the normal training set. For the data augmentation, we used the same approaches as for the training on the real data except for adjusting the contrast of the background and the pupil of the simulated images. Here we used the differences from the training set first to adjust the contrast.

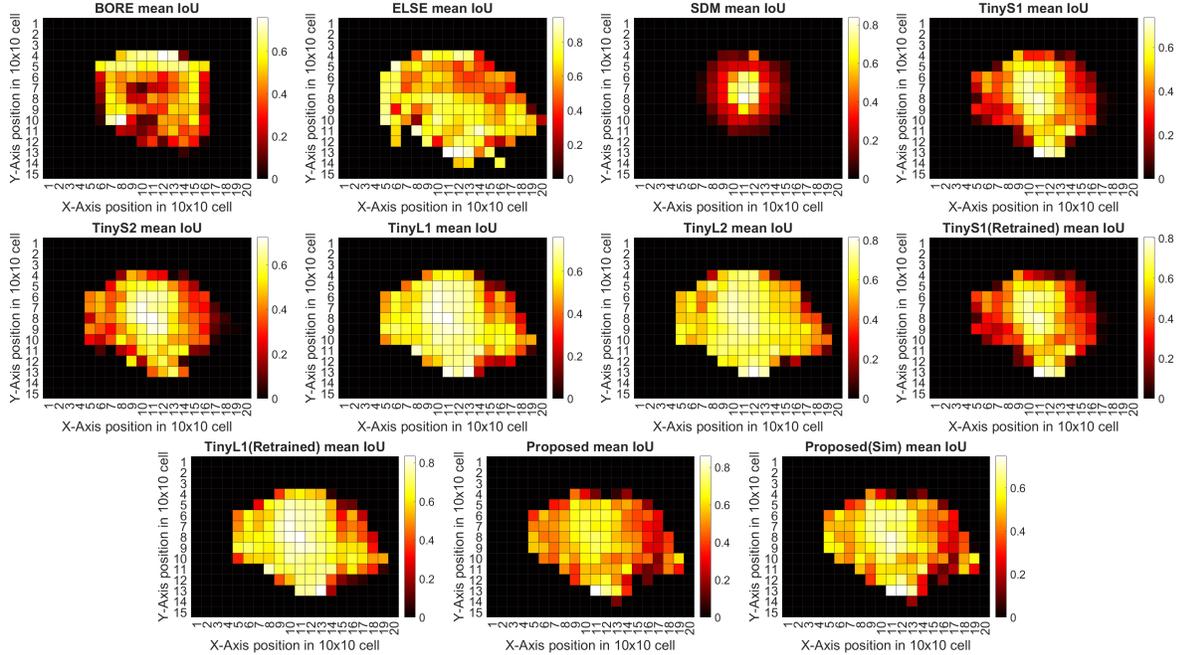


Figure 4. The mean intersection over union mapped to  $10 \times 10$  cells on the image space. Higher values or brighter colors are better results. Please note that although the maximum is 1, the color scaling for all plots is not uniform with respect to the maximum. The minimum, on the other hand, is uniform at zero for all plots.

Table 1. Runtime for training and execution of the evaluated approaches on more than 2.500.000 images (5 augmented versions of each image) with data augmentation. The execution time was always evaluated with a single CPU core.

Method	Training time (h)	Execution (ms)	Note
TinyCNN S1	40 (With GPU)	3.7	With teacher network training.
TinyCNN S2	40 (With GPU)	3.7	With teacher network training.
TinyCNN L1	44 (With GPU)	7.9	With teacher network training.
TinyCNN L2	42 (With GPU)	5.8	With teacher network training.
SDM (HOG+SVM)	17	4.2	With parameter grid search.
BORE	8	1.1	Grid search for optimal pupil sizes.
EISe	0	6.6	No training necessary.
Proposed	1	0.9	

The hardware used for training and running the final models consists of an Intel i5-4570 CPU running at 3.2 GHz. The system has 16GB of DDR4 memory and an NVIDIA 1050 TI with 4GB of GDDR5 memory. The GPU was only used for training the TinyCNNs [12]. All runtime analyses were performed on one CPU core.

For a comparison with the state of the art, we use EISe as a representative of edge-based approaches, BORE as a resource saving alternative, the TinyCNNs pre-trained on LPW [54] and provided by the authors as well as two newly trained TinyCNNs on the presented training data, and SDM [57] for landmark detection also as a resource saving alternative.

Figure 2 on the left shows the cumulative mean intersection over union. This metric holds the information about

the segmentation quality. On the right side of Figure 2 is the cumulative euclidean pupil center pixel error which is important for the gaze estimation accuracy. As can be seen on both plots in Figure 2, SDM and BORE perform worse. BORE cannot handle the reflections very well as can be seen especially in Figure 3 where a high mean pupil center error is present nearly everywhere on the image space. For SDM this is different since the method performs well in the near area of the mean shape (Figure 3). The best performance regarding a cumulative pupil center error of zero has EISe (Figure 2 right). It is also reaches the highest values for the cumulative intersection over union for a value of 0.9 (Figure 2 left). Apart from this the TinyCNNs and the proposed approach are more robust and reaching nearly 90% at a pixel error of two (Figure 2 right). After the pixel error of

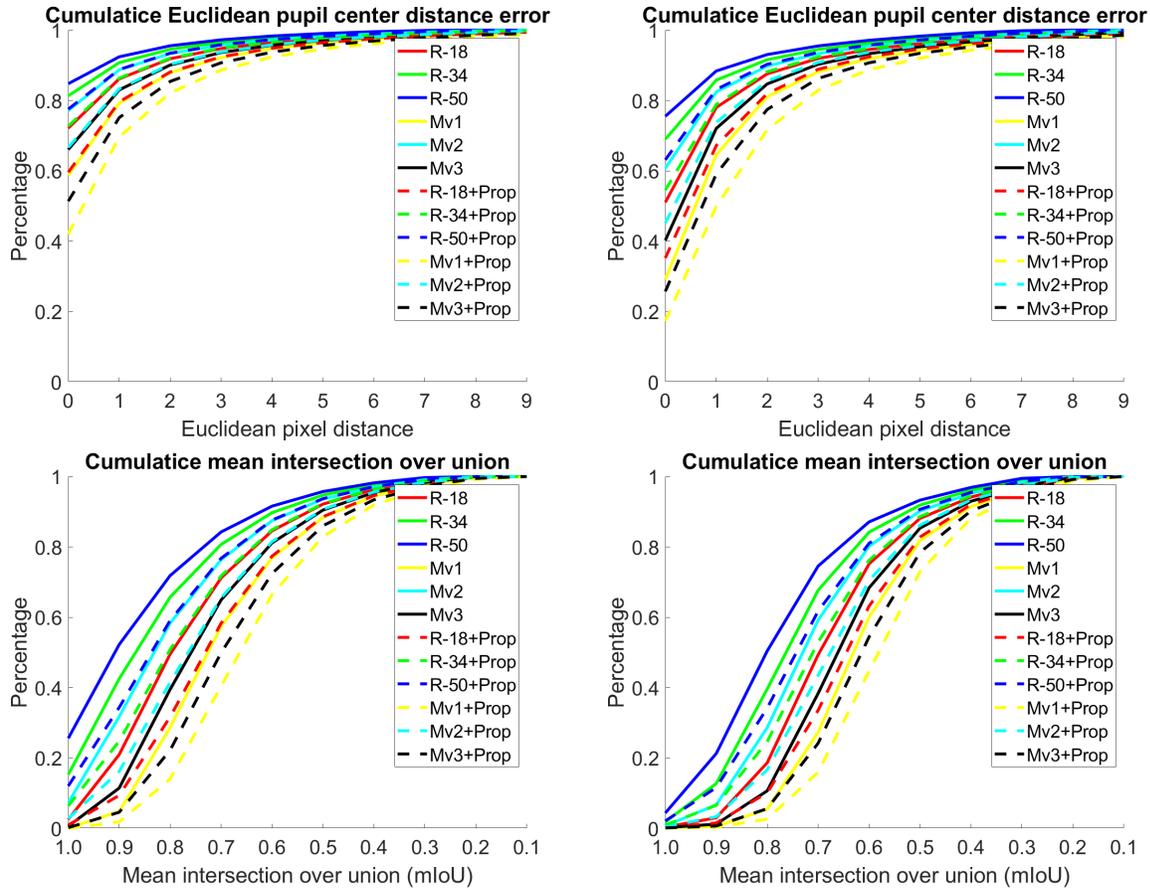


Figure 5. The cumulative euclidean distance error (top) and mean intersection over union (bottom) for the dataset [20] (left) and [14] (right). The dashed lines are the results of our approach in combination with the DNN model.

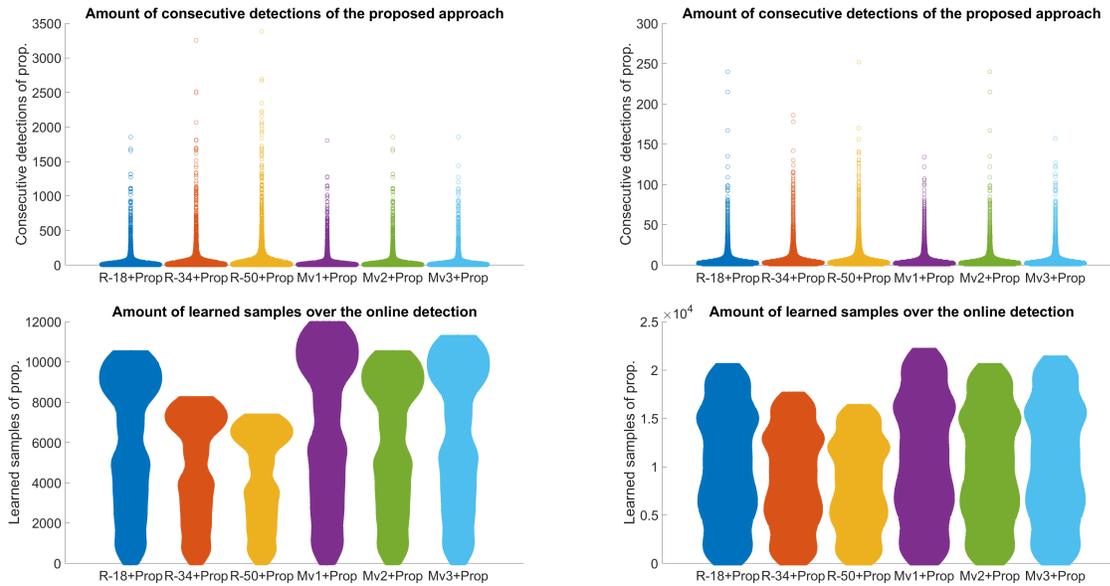


Figure 6. Consecutive detections of our approach (top) and the amount of learned samples (bottom) over the online learning experiment for the dataset [20] (left) and [14] (right). The width of the distribution in both plots corresponds to the occurrence.

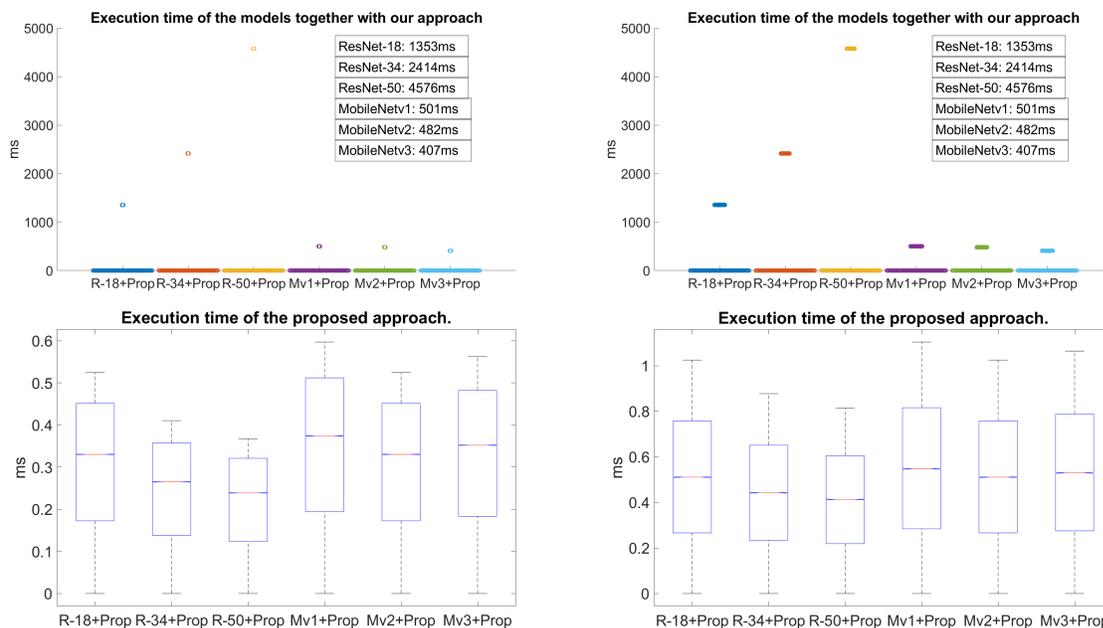


Figure 7. Distribution plots of the execution time per model in combination with the proposed approach (top) and the execution time of our approach alone without as whisker plot (bottom) on a single CPU core. The execution times in the textboxes in the two top plots are the runtimes of each model without the proposed approach on a single CPU core.

two, our approach is outperformed by the TinyCNNs but our approach needs only a fraction of computation time (See Table 1). For the segmentation quality, our approach keeps up with the TinyCNNs whereas the newly trained ones perform significantly better (Figure 2 left). This is due to the reduction of possible ellipses as described in the method section. In addition, it can be seen in Figure 2 that our approach trained on the simulated data performs only slightly worse compared to the one trained on the real data.

If Figure 3 and Figure 4 are compared for each approach, it can be seen that EISe has a lot of invalid detections over the entire image space (Figure 3). This stems from heavy reflections which make edge detection not applicable. In Figure 4 on the other hand, EISe has a good average segmentation quality over the image space with the exception of the upper right area where occlusions by the eyelid occurred. Another important information is the clear center bias which can be seen for SDM by comparing Figure 3 and Figure 4. Looking now at our approach and the TinyCNNs, we notice that they have good coverage of the entire image space (Figure 3 and Figure 4). In terms of segmentation, however, our approach is significantly worse in the outer areas (Figure 4).

Table 1 shows the training time in hours as well as the execution time in milliseconds on a single CPU core. As can be seen, our approach outperforms the other approaches in terms of execution time. For the training time EISe is the fastest approach due to it has not to be trained. Combining

the detection, segmentation, training time, and execution time results we think the proposed approach is a valuable contribution to the eye tracking community.

#### 4.1. Evaluation of the online learning

For online learning and application, we used the same dataset as for offline evaluation with the same training and test split [20]. Additionally, we used the segmented pupils provided with [14] for the LPW [54] images. This dataset consists of 66 videos from different individuals with an average length of 2000 frames. In this dataset, each video contains different challenges making it less optimal for our approach which can be seen in our results. For training and testing, we made a 50% to 50% split based on the videos and used each split once for training and once for evaluation. The training parameters for all networks and the data augmentation are the same as in the offline evaluation. To show the combination of our approach with different DNNs, we used the models ResNet-18 [30], ResNet-34 [30], ResNet-50 [30], MobileNetv1 [33], MobileNetv2 [50], and MobileNetv3 [32]. For all following plots the left side corresponds to the [20] dataset and the right plots to the dataset from [14].

Figure 5 shows the cumulative Euclidean error in the upper plots and the mean intersection over union (mIoU) in the two lower plots. The left side shows the results for the [20] data and the right side shows the results for the [14] dataset. The dashed lines are the results in combination with

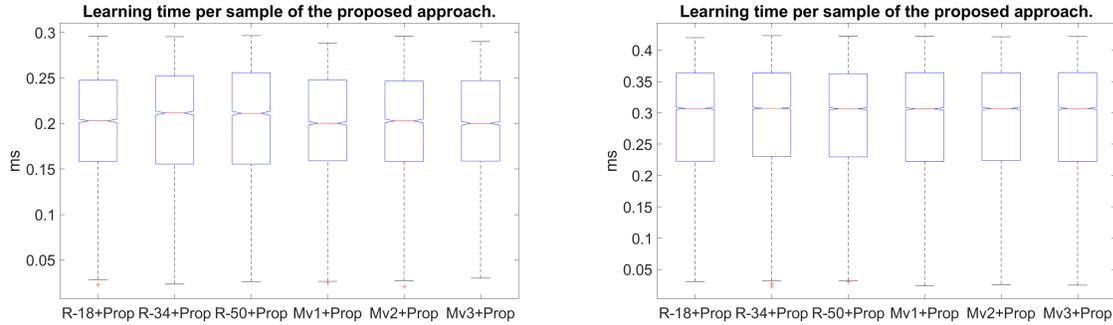


Figure 8. Training time per sample for each model of the proposed approach on a single CPU core.

our approach and the solid lines are the results of the models alone. The first thing to notice is that the [14] dataset is slightly more challenging. Additionally, it can be seen that our approach makes the results slightly less accurate. This is especially true for the segmentation results (mIoU). In contrast, our approach requires only one CPU core with a runtime below one millisecond (See Figure 7).

Figure 8 shows in the top two plots the distribution of the number of continuously detected pupils of our approach and in the bottom two plots the growing size of the learned statistical model. The membership with respect to the datasets is [20] on the left and [14] on the right. Comparing the top two plots, we can directly see that on the left side 10 times more images were detected in one sequence from our approach. This is because the dataset on the left ([20]) consists of multiple continuous long-term recordings and the dataset on the right ([14]) consists of small individual videos. This also leads to a larger statistical model as can be seen in the bottom two plots by the scaling of the y-axis.

Figure 7 shows the execution time of the models together with our approach in the upper plots. As can be seen, in the case of the dataset with the driving data from long-term recordings (Plot on the left), the DNN needs to be executed only very rarely (Single points above the flat distributions on the bottom). This shows the effectiveness of our approach. On the right, one can see the combined run times for the dataset with the 66 short videos. Here we also see that our approach drastically reduces the runtime, but the DNN must be run more often because new videos always contain new pupil positions and shapes. In the bottom two plots in Figure 7, the execution time of the presented approach alone can be seen. This is below one millisecond in almost all cases.

Figure 8 shows the required time of our approach to learn a new pupil ellipse. Since this only involves updating the statistics and moving the clusters, the execution time is relatively constant and very resource efficient as well as very fast to perform. In Figure 8 it can also be seen very clearly that the training of our approach does not depend on the model or the accuracy of the extracted pupil ellipse.

## 5. Limitations

While the presented approach with Haar features in combination with statistical learning has a very low training time and a very low runtime and can also be trained on simulated data, this approach of course also has disadvantages. The first disadvantage is the search area. This means that no pupils or ellipses can be found outside this area. Of course, this limitation can be easily circumvented by arbitrarily extending the search area, but this has a negative impact on both the detection rate and the runtime. Another disadvantage of the presented approach, is the statistic itself, which in the case of feature weighting weights frequent occurrences of valid features higher. This means that large datasets of similar images lead to features that are weighted more important than others. This results in an overfitting to these images. Also, the presented approach only recognizes shapes which are present in the training data. This is because unknown shapes are not sampled and have no probability of occurrence. This can be easily fixed by simulated data, data manipulation or the combination with DNNs as shown in the second part of our evaluation.

## 6. Conclusion

In this work, we have presented a new approach to efficiently train and segment pupils. While it is not able to segment pupils as accurately as, for example, edge-based approaches, it is comparatively robust and very efficient to compute. In addition, we showed that our approach can be trained online and used together with different DNN models. This allows to save energy, since no GPU/TPU is needed for our approach. This allows longer recordings to be made. If a user has already used the device for a longer period of time, as is the case for the data from [20] (Left Plots), the DNN only needs to be used sporadically.

## References

- [1] H. Bahmani, W. Fuhl, E. Gutierrez, G. Kasneci, E. Kasneci, and S. Wahl. Feature-based attentional influences on the ac-

- commodation response. In *Vision Sciences Society Annual Meeting Abstract*, 2016.
- [2] Thomas Brandt, Ralf Stemmer, and Andry Rakotonirainy. Affordable visual driver monitoring system for fatigue and monotony. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 7, pages 6451–6456. IEEE, 2004.
- [3] Aayush K Chaudhary, Rakshit Kothari, Manoj Acharya, Shushil Dangi, Nitinraj Nair, Reynold Bailey, Christopher Kanan, Gabriel Diaz, and Jeff B Pelz. Ritnet: real-time semantic segmentation of the eye for gaze tracking. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3698–3702. IEEE, 2019.
- [4] Siyuan Chen and Julien Epps. Using task-induced pupil diameter and blink rate to infer cognitive load. *Human-Computer Interaction*, 29(4):390–413, 2014.
- [5] L Dhir, NE Habib, DM Monro, and S Rakshit. Effect of cataract surgery and pupil dilation on iris pattern recognition for personal authentication. *Eye*, 24(6):1006–1010, 2010.
- [6] Andrew T Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34(4):455–470, 2002.
- [7] S Eivazi, W Fuhl, and E Kasneci. Towards intelligent surgical microscopes: Surgeons gaze and instrument tracking. In *Proceedings of the 22st International Conference on Intelligent User Interfaces, IUI*, 2017.
- [8] Shahram Eivazi, Michael Slupina, Wolfgang Fuhl, Hoorieh Afkari, Ahmad Hafez, and Enkelejda Kasneci. Towards automatic skill evaluation in microsurgery. In *Proceedings of the 22st International Conference on Intelligent User Interfaces, IUI 2017*. ACM, 03 2017.
- [9] Wolfgang Fuhl. From perception to action using observed actions to learn gestures. *User Modeling and User-Adapted Interaction*, pages 1–18, 08 2020.
- [10] W. Fuhl, S. Eivazi, B. Hosp, A. Eivazi, W. Rosenstiel, and E. Kasneci. Bore: Boosted-oriented edge optimization for robust, real time remote pupil center detection. In *Eye Tracking Research and Applications, ETRA*, 2018.
- [11] W. Fuhl, H. Gao, and E. Kasneci. Neural networks for optical vector and eye ball parameter estimation. In *ACM Symposium on Eye Tracking Research & Applications, ETRA 2020*. ACM, 01 2020.
- [12] W. Fuhl, H. Gao, and E. Kasneci. Tiny convolution, decision tree, and binary neuronal networks for robust and real time pupil outline estimation. In *ACM Symposium on Eye Tracking Research & Applications, ETRA 2020*. ACM, 01 2020.
- [13] W. Fuhl, D. Geisler, W. Rosenstiel, and E. Kasneci. The applicability of cycle gans for pupil and eyelid segmentation, data generation and image refinement. In *International Conference on Computer Vision Workshops, ICCVW*, 11 2019.
- [14] Wolfgang Fuhl, David Geisler, Wolfgang Rosenstiel, and Enkelejda Kasneci. The applicability of cycle gans for pupil and eyelid segmentation, data generation and image refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [15] W. Fuhl, D. Geisler, T. Santini, T. Appel, W. Rosenstiel, and E. Kasneci. Cbf:circular binary features for robust and real-time pupil center detection. In *ACM Symposium on Eye Tracking Research & Applications*, 06 2018.
- [16] W. Fuhl, D. Geisler, T. Santini, and E. Kasneci. Evaluation of state-of-the-art pupil detection algorithms on remote eye images. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct publication – PETMEI 2016*, 09 2016.
- [17] W. Fuhl and E. Kasneci. Learning to validate the quality of detected landmarks. In *International Conference on Machine Vision, ICMV*, 11 2019.
- [18] W. Fuhl, T. C. Kübler, D. Hospach, O. Bringmann, W. Rosenstiel, and E. Kasneci. Ways of improving the precision of eye tracking data: Controlling the influence of dirt and dust on pupil detection. *Journal of Eye Movement Research*, 10(3), 05 2017.
- [19] W. Fuhl, T. C. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci. Excuse: Robust pupil detection in real-world scenarios. In *16th International Conference on Computer Analysis of Images and Patterns (CAIP 2015)*, 09 2015.
- [20] W. Fuhl, W. Rosenstiel, and E. Kasneci. 500,000 images closer to eyelid and pupil segmentation. In *Computer Analysis of Images and Patterns, CAIP*, 11 2019.
- [21] W. Fuhl, T. Santini, D. Geisler, T. C. Kübler, and E. Kasneci. Eyelad: Remote eye tracking image labeling tool. In *12th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, 02 2017.
- [22] W. Fuhl, T. Santini, D. Geisler, T. C. Kübler, W. Rosenstiel, and E. Kasneci. Eyes wide open? eyelid location and eye aperture estimation for pervasive eye tracking in real-world scenarios. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct publication – PETMEI 2016*, 09 2016.
- [23] W. Fuhl, T. Santini, and E. Kasneci. Fast and robust eyelid outline and aperture detection in real-world scenarios. In *IEEE Winter Conference on Applications of Computer Vision (WACV 2017)*, 03 2017.
- [24] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci. Pupilnet: Convolutional neural networks for robust pupil detection. In *CoRR*, 2016.
- [25] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci. Pupilnet v2.0: Convolutional neural networks for robust pupil detection. In *CoRR*, 2017.
- [26] W. Fuhl, T. Santini, T. C. Kübler, and E. Kasneci. Else: Ellipse selection for robust pupil detection in real-world environments. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pages 123–130, 03 2016.
- [27] W. Fuhl, T. Santini, C. Reichert, D. Claus, A. Herkommer, H. Bahmani, K. Rifai, S. Wahl, and E. Kasneci. Non-intrusive practitioner pupil detection for unmodified microscope oculars. *Elsevier Computers in Biology and Medicine*, 79:36–44, 12 2016.
- [28] Wolfgang Fuhl, Marc Tonsen, Andreas Bulling, and Enkelejda Kasneci. Pupil detection for head-mounted eye tracking in the wild: An evaluation of the state of the art. In *Machine Vision and Applications*, pages 1–14, 06 2016.

- [29] Antonio Haro, Myron Flickner, and Irfan Essa. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 163–168. IEEE, 2000.
- [30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [31] Benedikt Hosp, Shahram Eivazi, Maximilian Maurer, Wolfgang Fuhl, David Geisler, and Enkelejda Kasneci. Remote-eye: An open-source high-speed remote eye tracker. *Behavior Research Methods*, pages 1–15, 2020.
- [32] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.
- [33] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [34] Amir-Homayoun Javadi, Zahra Hakimi, Morteza Barati, Vincent Walsh, and Lili Tcheang. Set: a pupil detection method using sinusoidal approximation. *Frontiers in neuroengineering*, 8:4, 2015.
- [35] Erika Jönsson. If looks could kill—an evaluation of eye tracking in computer games. *Unpublished Master’s Thesis, Royal Institute of Technology (KTH), Stockholm, Sweden*, 2005.
- [36] Rakshit S Kothari, Aayush K Chaudhary, Reynold J Bailey, Jeff B Pelz, and Gabriel J Diaz. Ellseg: An ellipse segmentation framework for robust gaze tracking. *arXiv preprint arXiv:2007.09600*, 2020.
- [37] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.
- [38] Andoni Larumbe, Rafael Cabeza, and Arantxa Villanueva. Supervised descent method (sdm) applied to accurate pupil detection in off-the-shelf eye tracking systems. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 1–8, 2018.
- [39] Päivi Majaranta and Andreas Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*, pages 39–65. Springer, 2014.
- [40] Vladimir A Maksimenko, Alexander E Hramov, Nikita S Frolov, Annika Lüttjohann, Vladimir O Nedaivozov, Vadim V Grubov, Anastasia E Runnova, Vladimir V Makarov, Jürgen Kurths, and Alexander N Pisarchik. Increasing human performance by sharing cognitive load using brain-to-brain interface. *Frontiers in neuroscience*, 12:949, 2018.
- [41] Ion Martinikorena, Rafael Cabeza, Arantxa Villanueva, Iñaki Urtasun, and Andoni Larumbe. Fast and robust ellipse detection algorithm for head-mounted eye tracking systems. *Machine Vision and Applications*, 29(5):845–860, 2018.
- [42] R Andy McKinley, Lindsey K McIntire, Regina Schmidt, Daniel W Repperger, and John A Caldwell. Evaluation of eye metrics as a detector of fatigue. *Human factors*, 53(4):403–414, 2011.
- [43] Martin Meißner, Jella Pfeiffer, Thies Pfeiffer, and Harmen Oppewal. Combining virtual reality and mobile eye tracking to provide a naturalistic experimental environment for shopper research. *Journal of Business Research*, 100:445–458, 2019.
- [44] Xiaoxu Meng, Ruofei Du, Matthias Zwicker, and Amitabh Varshney. Kernel foveated rendering. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(1):1–20, 2018.
- [45] Salem Naeri, Saptarshi Mandal, and Zihong Kang. Analyzing pilots’ fatigue for prolonged flight missions: Multimodal analysis approach using vigilance test and eye tracking. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 63, pages 111–115. SAGE Publications Sage CA: Los Angeles, CA, 2019.
- [46] Sadia Najmi, Nader Amir, Kristen E Frosio, and Catherine Ayers. The effects of cognitive load on attention control in subclinical anxiety and generalised anxiety disorder. *Cognition and Emotion*, 29(7):1210–1223, 2015.
- [47] Simona Naspetti, Roberto Pierdicca, Serena Mandolesi, Marina Paolanti, Emanuele Frontoni, and Raffaele Zanolini. Automatic analysis of eye-tracking data for augmented reality applications: A prospective outlook. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 217–230. Springer, 2016.
- [48] Alexandra Papoutsaki, James Laskey, and Jeff Huang. Searchgazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, pages 17–26, 2017.
- [49] Thammathip Piumsomboon, Gun Lee, Robert W Lindeman, and Mark Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 36–39. IEEE, 2017.
- [50] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [51] Nelson Silva, Tanja Blascheck, Radu Jianu, Nils Rodrigues, Daniel Weiskopf, Martin Raubal, and Tobias Schreck. Eye tracking support for visual analytics systems: foundations, current applications, and research challenges. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, pages 1–10, 2019.
- [52] Yusuke Sugano and Andreas Bulling. Self-calibrating head-mounted eye trackers using egocentric visual saliency. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 363–372, 2015.
- [53] Lech Świrski, Andreas Bulling, and Neil Dodgson. Robust real-time pupil tracking in highly off-axis images. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 173–176, 2012.

- [54] Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 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*, pages 139–142, 2016.
- [55] MéLodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. Wearable eye tracking for mental health monitoring. *Computer Communications*, 35(11):1306–1311, 2012.
- [56] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.
- [57] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013.
- [58] Junli Xu, Jianliang Min, and Jianfeng Hu. Real-time eye tracking for the assessment of driver fatigue. *Healthcare technology letters*, 5(2):54–58, 2018.
- [59] Yuk-Hoi Yiu, Moustafa Aboulatta, Theresa Raiser, Leoni Ophhey, Virginia L Flanagan, Peter zu Eulenburg, and Seyyed-Ahmad Ahmadi. Deepvog: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of neuroscience methods*, 324:108307, 2019.
- [60] Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J Zelinsky, and Tamara L Berg. Studying relationships between human gaze, description, and computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 739–746, 2013.
- [61] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.
- [62] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4511–4520, 2015.