Eye Localisation using Cascaded U-Net for Autism Spectrum Disorder

[1,2]INES RAHMANY, [1]SABRINE BRAHMI, [2]NAWRES KHLIFA
[1]Faculty of Sciences and Techniques, University of Kairouan, Kairouan, TUNISIA
[2]Universit´e de Tunis El Manar, Laboratoire Biophysique et Technologies Medicales, ISTMT, Tunis, TUNISIA.

Abstract: Many computer vision applications Computer Aided Diagnosis require an accurate and efficient eye detector. We represent, in this work, an efficient approach for determining the position of the eye in images presenting faces. First, a series of candidate regions are proposed; next, a set of cascaded U-net is used to detect the eye regions; then edge detection methods are used to detect eyes lids and iris boundary and thus helping determining the gaze direction of person having ASD. Our proposed approach achieved a precision of detection that is better than the current most recent methods in trials utilizing GI4E, BioID, and columbiaGaze datasets. The proposed approach is robust to picture alterations, such as changes in external illumination, facial occlusion, and image modality

Keywords: ASD, Eye Detection, Segmentation, Deep Learning , U-net.

Received: March 21, 2021. Revised: April 17, 2022. Accepted: July 10, 2022. Published: Augustu 4, 2022.

1. Introduction

Eye detection has become a hot topic in the computer vision and the pattern recognition field during the recent period [1], [2], because the locations of human eyes are crucial for a variety of applications, such as psychology, facial expression identification, medicine, and auxiliary driving, [3]. Yet, practically, eye detection is rather difficult. Since cameras are sensitive to the fluctuations of light and the distance of shooting, human eyes in a facial image prove to be quite eccentric. Sometimes face is partially obscured which affects certain existing eye detection methods that depend on facial model detection [4]. An eye detector has also to perform effectively in a variety of image modalities, such as visible and infrared images. Furthermore, the eye identification method has to be quick since it is expected to be used online in various scenarios. Even though, several approaches for detecting the eyes from facial images have been presented, finding the approach that best performs regarding accuracy, reliability, and efficiency is tough. As a result, we aim to create an efficient and robust eye detection method. This work is a part of the gaze tracker project for Autism Spectrum Disorder (ASD) Diagnosis. The organization of the remainder of the paper will be as follows. A review of state of the art methods is presented in section 2 Related Work. In section 3, the Proposed Method is found; we present a method for eye detection that includes the generation of the candidate region, eye region identification and detection. Then, in Section 4, dataset, the achieved results, evaluation and discussions are exposed. Finally, in the last section, conclusions are presented.

2. Related Work

Image-based eye detection algorithms are classified into three types: classical or traditional eye detection methods, machine learning eye detection methods, and deep learning eye detection methods. The classical eye detection methods which are based on the geometric characteristics of the eye can be categorized into two subcategories; the geometric model and the template matching. In the first subclass Valenti and Gevers [5] designed a voting mechanism for the localization of the eye pupil based on the curvature of isophotes. Markus et al. [6] suggested an ensemble of randomized regression trees-based approach for eye pupil localization. To detect the pupils, Timm and Barth [7] proposed a method based on the curvature of isophotes and squared dot products.

While in the second subclass, the RANSAC [8] technique was used to construct an elliptic equation to fit the pupil center. Based on correlation filters, Araujo et al. [9] talked about an Inner Product Detector for the localization of the eye. Although the traditional eye detectors can sometimes prove their efficiency, they do not perform with the same efficiency when the external light or facial occlusion changes. Machine learning eye detectors have two essential bases which are the feature extraction that is followed by a cascaded classifier. For quick classification, Chen and Liu [10] used wavelet technique (Haar) and a support vector machine (SVM). To create an efficient eye detector, Sharma and Savakis [11] suggested a method based on oriented gradients (HOG) features in learning histogram combined with SVM classifiers. For eye center detection, Leo et al. [12], [13] opted for self-similarity information paired with shape analysis. D'orazio et al. [14]introduced a geometrical parameter restriction and a neural classifier meant to identify eye regions. For simultaneous eye localization and eye state assessment, Gou et al. [15] developed a cascade regression framework. Kim et al. [16] proposed using multiscale iris shape characteristics to generate eye candidate regions, which would then be verified using HOG descriptor and the features of mean intensity. Since deep learning methods have become popular [17], some researchers have employed (Convolution Neural Network) CNN for training eye detectors, and this is what the second subcategory consists in. A CNN-based pupil center identification approach is presented by Chinsatit and Saitoh in [18]. In Fuhl's [19] study, which employed two similar convolutional neural networks to distinguish between coarse and fine pupil location, the authors suggested that subregions be computed using a downscaled input picture. Amos et al. [20] used 68 feature points to train a facial landmark detector to define the face model, with 12 feature points describing the eye shape. When compared to traditional methods, deep learning-based methods have proved to be more resilient and accurate. However, efficiency remains a problem. Facial images are typically larger than 640 x 480 pixels. Only the chosen candidate areas are entered into the CNNs, hence it is crucial to propose candidate regions quickly and constructively. The classification of the left and right eyes and the eye center location are also important in some applications, such as eye tracking systems and illness detection. On the other hand, the majority of eye detectors that are used nowadays are unable to identify the eye regions, split apart the left and right eyes, and identify the center of the eye in an one round.As a result, our goal is to provide an innovative way that presents a solution for the problems present in the old methods. The Figure presents a taxonomy of eye detection methods.

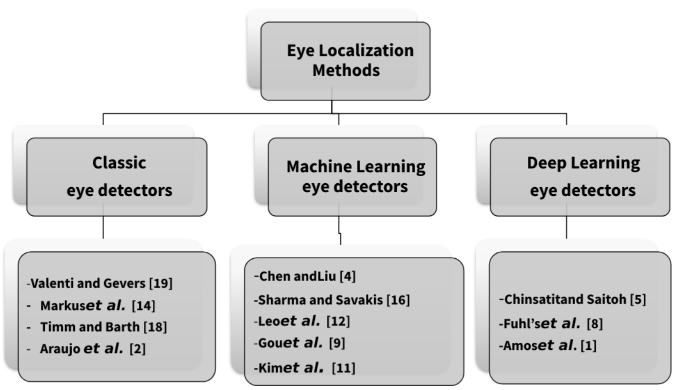

Fig. 1: **Taxonomy of eye detection methods.**

3. The Proposed Methodology

Figure 2 presents the overall flowchart of the proposed 4 phases method using cascaded U-net. In the first phase, in the complete facial image we swiftly produce a number of eye region candidates. The second U-net examined these possible eye regions in the second stage to identify the regions of the eye and the eye lids. The third U-net is used to pinpoint the iris in the third phase. Then edge detection methods are used to detect eyes lids and iris boundary and thus helping determining the gaze direction of person having ASD.

The architecture of the U-net is shown in Figure 3. It consists of an expanded path on the right and a contracting path on the left (left side). The contracting route of the convolutional network obeys the standard architecture. It consists of two 3x3 convolutions (unpadded convolutions) that are performed twice, with a rectified linear unit (ReLU) applied after each one, and a 2x2 max pooling action and downsampling with stride 2. The number of feature channels is quadrupled with each down- sampling step. After the

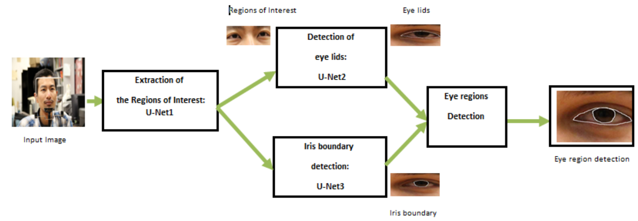

Fig. 2: **The overall flowchart of the proposed method.**

upsampling of the feature map a 2x2 convolution is found (up-convolution) which halves the number of feature channels, two 3x3 convolutions, each followed by a ReLU in the expanding route, and a concatenation with the proportionately cropped feature map from the contracting path. Cropping is necessary since every convolution results in the loss of border pixels. In order to convert each 64-component feature vector to the proper number of classes, a 1x1 convolution is used as the final layer. There are a total of 23 convolutional layers in the network.

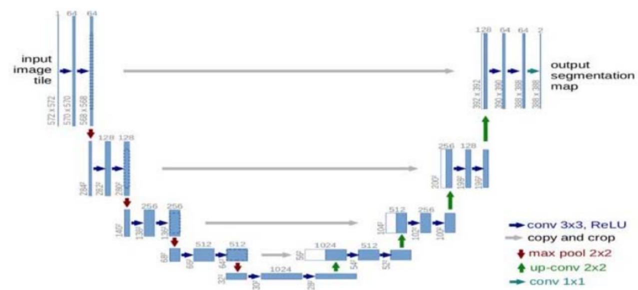

Fig. 3: **The U-net architecture.**

4. Results and Evaluations

4.1. Dataset

In this work, we consider a public database of images of people: CGDS (Columbia Gaze Data Set). CGDS and a publicly available gaze dataset: 5,880 images of 56 people (32 males, 24 females) presenting various gaze directions and poses of the head, and each image has a resolution of 5184 x 3456 pixels. For each subject, there are 5 head poses and 21 gaze directions per head pose at the time of its publication. The presented subjects come from different ethnicities and 21 of them wore glasses. A sample of columbia Gaze dataset is presented in Figure 4.

Every combination of five horizontal head postures (0°, ±15°, ±30°), seven horizontal gaze directions (0°, ±5°, ±10°, ±15°), and three vertical gaze directions (0°, ±10°) resulted in the acquisition of one picture for each individual. Table I presents the attributes of the images used in the dataset.

4.2. Eye Detection Results

Figures in this subsections display some qualitative results of our proposed cascaded U-net method.

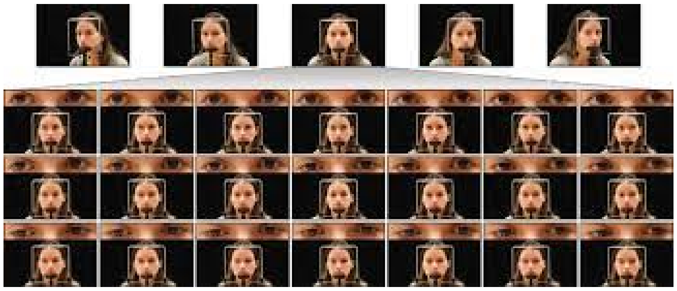

Fig. 4: Sample of Columbia Gaze dataset [21].

TABLE I: CGDS dataset attributes.

Number of subjects:	56
Gaze targets by subject and head pose:	21
Fixed head poses :	5
Head Pose Calibration :	oui
Resolution (px):	5184 * 3456
Total images:	5,880

1) Phase 1 Results : Extraction of Region of Interest: The 1st U-net inputs and outputs for the extraction of region of interest are given in Figures 5 and 6.

Fig. 5: Input Images of the U-net 1.

2) Phase 2 Results : Detection of eye lids: We use eye regions of interest provided by the U-net 1 as inputs (Figure 7) to second U-net which segments and detects (locate) the ocular eye area (Figure 8). Sobel filter is applied in order to detect the eye lids (Figure 9)

3) Phase 3 Results : Iris boundary detection: We use the ocular eye area provided by the U-net 2 as inputs (Figure 10) to third U-net which segments and detects (locate) the iris (Figure 11). Sobel filter is applied in order to detect the iris boundary (Figure 12).

4) Phase 4 Results : Eye regions Detection: In the last step, we made the fusion of U-Net2 and U-Net3 which results of the localization of different eye regions (Figure 13), thus helping determining the gaze direction.

We presented here the sets of U-nets outputs for detecting and segmenting ocular areas in order to locate the eye. Our approach is effective and demonstrates that it can successfully detect eye positions. visible. We used the normalized error to test the eye classification and detection capabilities,

4.3. Evaluation

Our proposed method was tested on Google Colab environment. It is a free cloud-based online environment for Jupyter notebooks, to train our deep learning and machine learning models on GPUs. The specifications used runtime offered by Google Colab is presented in the Table II.

TABLE II: The specifications of runtime offered by Google Colab.

GPU	Up to Tesla K80 with 12 GB of GDDR5 VRAM, Intel Xeon Processor with two cores @ 2.20 GHz and 13 GB RAM

The outputs of the three stages of cascaded U-nets were examined individually during the evaluation on the Columbia Gaze dataset. For the U-nets' training and testing, we randomly separated the dataset into two portions. we randomly choose 80% of images from each class as training set, while the rest (20%) are used for test.

Currently, there are no analytical approaches to design the parameters in U-net networks. Therefore, to get an optimal result from the algorithm, we need to find the best parameters empirically. The model parameters are presented in Table III.

TABLE III: Model Parameters.

Nb Total Images	5880
Nb Train Images	4 704
Nb Test Images	1 176
Batch size	16
Epochs	16
Optimizer	ADAM
Activation	softmax

Figure 14 shows that the two curves of loss and accuracy with an epoch number equal to 16 is near the boundary of the rectangle. The accuracy is 98.61%. This shows that the proposed model presents high performance (a good image segmentation model for eye detection).

We examined the average time of each phase. The Table IV below presents the overall performance of the proposed method.

TABLE IV: Results of the eye localization phases by the proposed cascaded U-net model.

Proposed method	Accurracy	Execution times
U-Net 1	100%	18s
U-Net 2	95.19%	11s
U-Net 3	97.61%	8s

The comparison between the suggested method and state of the art techniques used on the BioID dataset and Columbia Gaze dataset is shown in Table V.

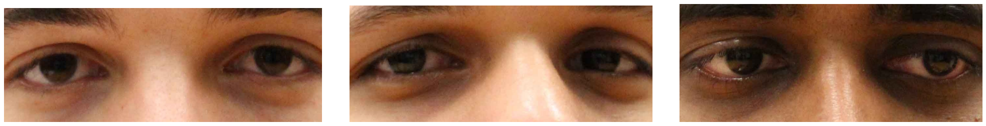

Fig. 6: Phase 1 Results : Extraction of Region of Interest.

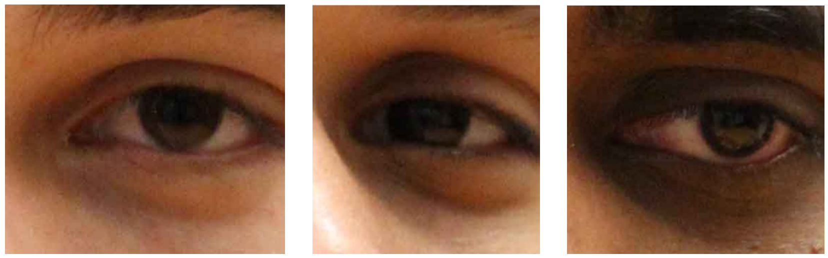

Fig. 7: Input Images of the U-net 2.

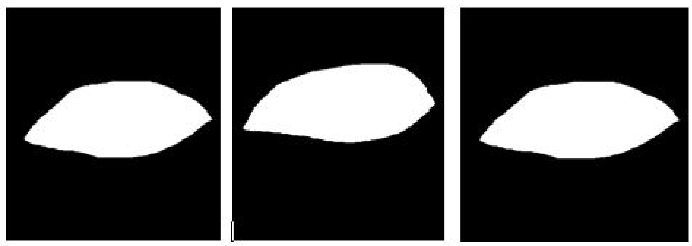

Fig. 8: Ocular eye area.

Fig. 9: Phase 2 Results : Detection of eye lids.

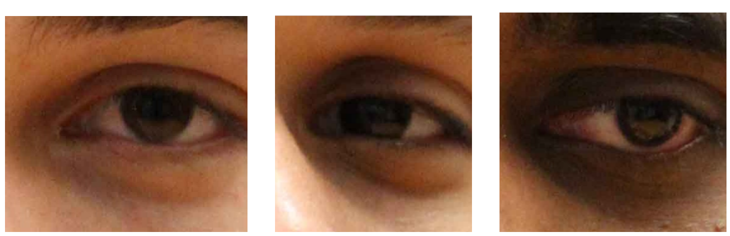

Fig. 10: Input Images of the U-net 3.

[5] [9] [15]

In comparison to Valenti and Gevers [5] (86.1%), Araujo et al. [9] (88.3%), and Gou et al. [15] (91.2%) approaches, our suggested method performs better (97.61%). Additionally, our suggested approach is robust even when the subject of the picture test is wearing spectacles and does not need clustering. The approaches used by Valenti and Gevers [5] and Araujo et al. [9] are less reliable than ours.

5. Conclusions

In this paper, we developed an efficient cascaded U-net method for locating the eye in face photos. Images captured using visible or infrared light have no impact on the suggested strategy. Additionally, the face detector is not dependent on the placement of the eyes. We tested our approach using more than 5,000 facial photos for the evaluation and found that our suggested eye detector was helpful and efficient. To produce

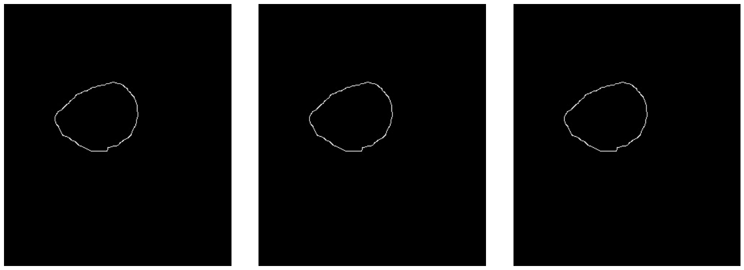

Fig. 11: Segmentation of the iris

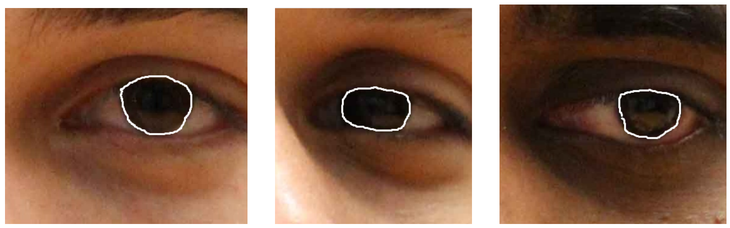

Fig. 12: Phase 3 Results : Iris boundary detection

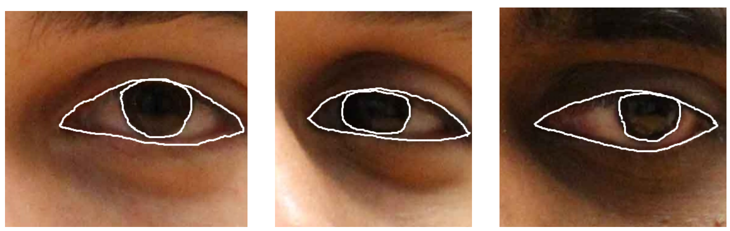

Fig. 13: Eye location results.

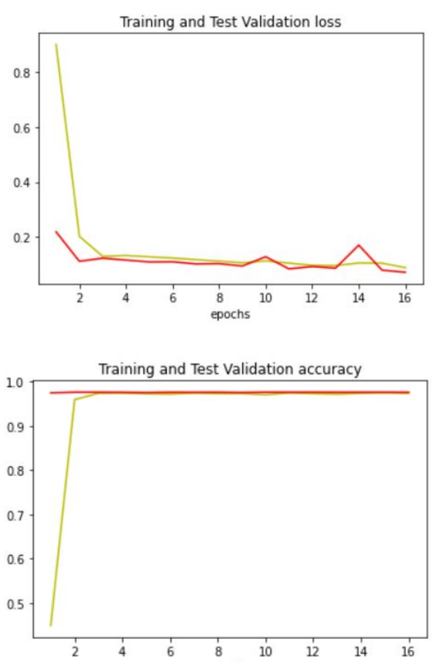

Fig. 14: Accuracy and loss curve of the proposed U-net model.

TABLE V: Comparison of the achieved results of the proposed approach with other state-of-the-art methods.

Technique	Databases	Accurracy
George et al. [22]	BioID: 1521 images GI4E: 1380 images	94.74%
Yiu et al. [23]	BioID: 1521 images	96%
Valenti et al. [5]	BioID: 1521 images	86.1%
Araujo et al. [9]	BioID: 1521 images	88.3%
Gou et al. [15]	BioID: 1521 images	91.2%
Proposed method	BioID: 1521 images	97.61%

good segmentation results and a noticeably high efficiency. The gaze tracker project for the diagnosis of autism spectrum disorder includes this study.

References

[1] H. Fu, Y. Wei, F. Camastra, P. Arico, and H. Sheng, “Advances in eye tracking technology: theory, algorithms, and applications,” *Computational intelligence and neuroscience*, vol. 2016, 2016.

[2] L. Zhang, Y. Cao, F. Yang, and Q. Zhao, “Machine learning and visual computing,” 2017.

[3] A. H. Mosa, M. Ali, and K. Kyamakya, “A computerized method to diagnose strabismus based on a novel method for pupil segmentation,” 2013.

[4] L. Birgit and M. Brodsky, “Pediatric ophthalmology, neuro-ophthalmology, genetics: Strabismus-new concepts in pathophysiology, diagnosis, and treatment,” 2010.

[5] R. Valenti and T. Gevers, "Accurate eye center location through invariant isocentric patterns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1785–1798, 2011.
[6] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer, "Eye pupil localization with an ensemble of randomized trees," *Pattern recognition*, vol. 47, no. 2, pp. 578–587, 2014.
[7] F. Timm and E. Barth, "Accurate eye centre localisation by means of gradients." *Visapp*, vol. 11, pp. 125–130, 2011.
[8] L. Świrski, A. Bulling, and N. Dodgson, "Robust real-time pupil tracking in highly off-axis images," in *Proceedings of the symposium on eye tracking research and applications*, 2012, pp. 173–176.
[9] G. M. Araujo, F. M. Ribeiro, E. A. Silva, and S. K. Goldenstein, "Fast eye localization without a face model using inner product detectors," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 1366–1370.
[10] S. Chen and C. Liu, "Eye detection using discriminatory haar features and a new efficient svm," *Image and Vision Computing*, vol. 33, pp. 68–77, 2015.
[11] R. Sharma and A. Savakis, "Lean histogram of oriented gradients features for effective eye detection," *Journal of Electronic Imaging*, vol. 24, no. 6, p. 063007, 2015.
[12] M. Leo, D. Cazzato, T. De Marco, and C. Distante, "Unsupervised approach for the accurate localization of the pupils in near-frontal facial images," *Journal of Electronic Imaging*, vol. 22, no. 3, p. 033033, 2013.
[13] ——, "Unsupervised eye pupil localization through differential geometry and local self-similarity matching," *PloS one*, vol. 9, no. 8, p. e102829, 2014.
[14] T. D'orazio, M. Leo, and A. Distante, "Eye detection in face images for a driver vigilance system," in *IEEE Intelligent Vehicles Symposium, 2004*. IEEE, 2004, pp. 95–98.
[15] C. Gou, Y. Wu, K. Wang, K. Wang, F.-Y. Wang, and Q. Ji, "A joint cascaded framework for simultaneous eye detection and eye state estimation," *Pattern Recognition*, vol. 67, pp. 23–31, 2017.
[16] H. Kim, J. Jo, K.-A. Toh, and J. Kim, "Eye detection in a facial image under pose variation based on multi-scale iris shape feature," *Image and Vision Computing*, vol. 57, pp. 147–164, 2017.
[17] D. Xie, L. Zhang, and L. Bai, "Deep learning in visual computing and signal processing," *Applied Computational Intelligence and Soft Computing*, vol. 2017, 2017.
[18] W. Chinsatit and T. Saitoh, "Cnn-based pupil center detection for wearable gaze estimation system," *Applied Computational Intelligence and Soft Computing*, vol. 2017, 2017.
[19] W. Fuhl, T. Santini, G. Kasneci, and E. Kasneci, "Pupilnet: Convolutional neural networks for robust pupil detection," *arXiv preprint arXiv:1601.04902*, 2016.
[20] B. Amos, B. Ludwiczuk, M. Satyanarayanan *et al.*, "Openface: A general-purpose face recognition library with mobile applications," *CMU School of Computer Science*, vol. 6, no. 2, 2016.
[21] B. Smith, Q. Yin, S. Feiner, and S. Nayar, "Gaze Locking: Passive Eye Contact Detection for Human?Object Interaction," in *ACM Symposium on User Interface Software and Technology (UIST)*, Oct 2013, pp. 271–280.
[22] A. George and A. Routray, "Fast and accurate algorithm for eye localisation for gaze tracking in low-resolution images," *IET Computer Vision*, vol. 10, no. 7, pp. 660–669, 2016.
[23] Y.-H. Yiu, M. Aboulatta, T. Raiser, L. Ophey, V. L. Flanagin, P. Zu Eulenburg, and S.-A. Ahmadi, "Deepvog: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning," *Journal of neuroscience methods*, vol. 324, p. 108307, 2019.