Iris Center Localization Using Geodesic Distance and CNN

Radovan Fusek(✉) and Eduard Sojka

FEECS, Department of Computer Science, Technical University of Ostrava,
17. listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic
{radovan.fusek,eduard.sojka}@vsb.cz

Abstract. In this paper, we propose a new eye iris center localization method for remote tracking scenarios. The method combines the geodesic distance with CNN-based classification. Firstly, the geodesic distance is used for fast preliminary localization of the regions possibly containing the iris. Then a convolutional neural network is used to carry out the final decision and to refine the final position of the iris center. In the first step, the areas that do not appear to contain the eyeball are quickly filtered out, which makes the whole algorithm fast even on less powerful computers. The proposed method is evaluated and compared with the state-of-the-art methods on two publicly available datasets focused to the remote tracking scenarios (namely BioID [9], GI4E [15]).

Keywords: CNN · Iris detection · Geodesic distance · Deep learning

1 Introduction

In the area of recognition of eye movements, the remote and head-mounted eye-tracker systems have been widely deployed in recent years. The head-mounted eye-tracker systems are represented by the devices that are very often attached to the user's head. These systems can be used to obtain accurate information on the eye movements, such as gaze direction, or iris and pupil positions. However, these systems are more intrusive for the users than the remote eye-tracker systems. The remote trackers can be created by a single camera or by multiple cameras located away from the user. For example, these kinds of trackers are used inside the vehicle cockpits to recognize fatigue of the driver or blinking frequency. The remote systems can also be used for iris and pupil localization, however, due to the fact that the images provided by the remote systems have usually a low resolution, recognition of the eye parts represents a challenging task.

In this paper, we propose a method for localization of iris center for the remote tracking scenarios. The method is based on the geodesic distance combined with a convolutional neural network (CNN). In [6], the authors show that the geodesic distance can be used for pupil localization. We experimented with that method and we observed detection shortcomings, which became the motivation for this paper. However, we found that the method can be useful, especially,

A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11868, pp. 76–85, 2019.
https://doi.org/10.1007/978-3-030-31321-0_7

for fast detecting the coarse position of iris. Our new method runs in two steps. In the first step, we use the ideas presented in [6] for preliminarily estimating the candidate areas. The final determination of iris position is done by making use of CNN in the second step. The second step extends and improves the original method, which is the main contribution of this paper. The presented experiments show that the proposed method outperforms the original method [6] and the state-of-the-art methods in this area.

The rest of the paper is organized as follows. The previously presented papers from the area of eye analysis are mentioned in Sect. 2. In Sect. 3, the main steps of the proposed method are described. In Sect. 4, the results of experiments are presented.

2 Related Work

In the area of iris and pupil detection, many different approaches have been presented. In [13], a method designed for head-mounted eye-tracking systems for pupil localization was proposed. The main steps include: removing the corneal reflection, pupil edge detection using a feature-based technique, and the ellipse fitting step using RANSAC. Swirski et al. [14] presented the method that is based on a Haar-like feature detector to roughly estimate the pupil location in the first step. In the next step, the potential pupil region is segmented using k-means clustering to find the largest black region. In the final step, the edge pixels of region are used for ellipse fitting using RANSAC. Exclusive Curve Selector or *ExCuSe* was proposed in [2]. This method is based on the histogram analysis combined with the Canny edge detector and ellipse estimation using the direct least squares method. In [8], another pupil detection method known as *SET* is proposed. The method is based on thresholding, segmentation, border extraction using the convex hull method, and selection of the segment with the best fit. In [5], another approach known as *ElSe* is presented. The method uses edge filtering, ellipse evaluation, and pupil validation. Another method for determining the iris centre in low-resolution images is proposed in [7]. In the first step, the coarse location of iris centre is determined using a novel hybrid convolution operator. In the second step, the iris location is further refined using boundary tracing and ellipse fitting. In [10], the pupil localization method based on the training process and the Hough regression forest was proposed. The method based on a convolutional neural network is proposed in [3,4]. An evaluation of the state-of-the-art pupil detection algorithms is presented in [1].

3 Proposed Method

In many iris or pupil detection methods, the coarse position of iris or pupil is localized in the first step. For example, a circle-shaped (due to the shape of pupil) convolution filter is used in [7]. In [14], the approximate pupil region is localized using a Haar-like center-surround feature.

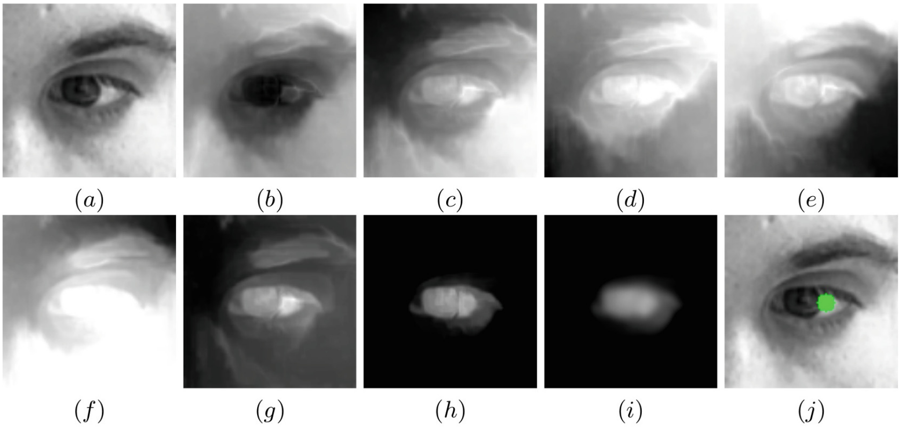

Fig. 1. The steps of eyeball and iris center localization using Geodesic distances. The input image (a). The visualization of the distance function from the centroid (b) and from particular corners (c, d, e, f). The mean of all corner distances (g). The difference (h) between (g) and (b) (only the non-zero distances are shown). The result of convolution step (i). The final position of iris center (j). The values of distance function are depicted by the level of brightness.

In this paper, we adopt the coarse localization of iris (eyeball) presented in [6]. For convenience of the reader, we briefly mention this approach. The approach is based on the geodesic distance that is used in the following way. Suppose that the image of eye region (Fig. 1(a)) is obtained beforehand (e.g. using facial landmarks or eye detector). In the first step, the geodesic distance is computed from the centroid (the point located in the center of the eye region) to all other points inside the eye image (Fig. 1(b)). The geodesic distance between two points computes the shortest curve that connects both points along the image manifold. Since the values of distance function are high in the area of eyebrow, this step is useful for its removing. It can be clearly seen that the areas with low distances represent the potential location of pupil and iris.

In the next step, the geodesic distance is also computed from each image corner to all other points inside the image (Fig. 1(c–f)). Then, the mean of all corner distances is calculated (Fig. 1(g)). Thereafter, for automatic extraction of eyeball area, the difference between Fig. 1(g) and (b) is carried out. In the image that shows this difference (Fig. 1(h)), it can be seen that the eyebrow area is removed and the potential area of iris is localized. In [6], the authors used the convolution with the Gaussian kernel in the last step (Fig. 1(i)). Then, the final iris position is determined as the location with the maximum value. In Fig. 1(j), the iris center position obtained using this approach is shown. In this particular case, it can be seen that the method fails to find the correct pupil and iris center (position) due to the fact that the iris is gently off-centered. Figure 1(a) is taken from the GI4E dataset [16] that contains many similar off-center iris and pupil images. We observed that these kinds of images cause difficulties for the method

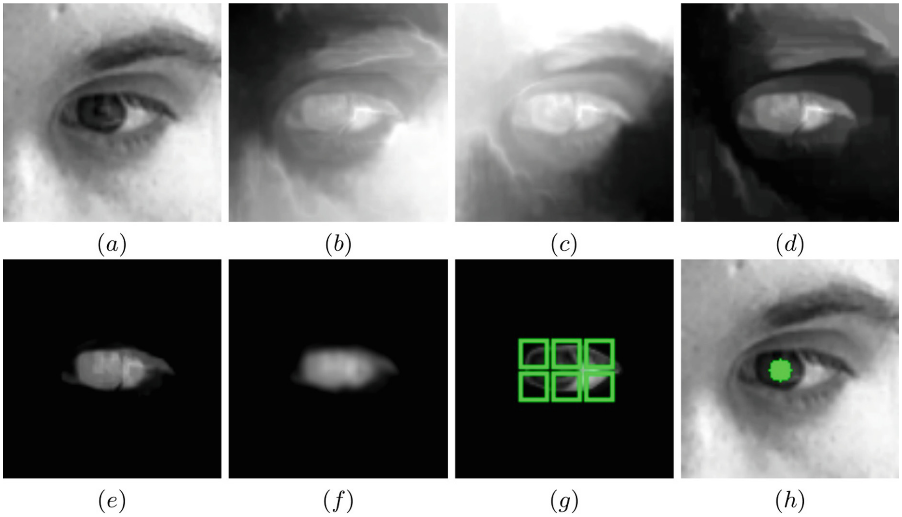

Fig. 2. The steps of iris center localization using the proposed approach. The input image (a). The visualization of the distance function from the two corners (top left (b) and bottom right (c)). The mean of two corner distances (d). An example of extracted preliminary iris region (e) using the difference step between (d) and Fig. 1(b). The result of convolution step (f). An example of cropped images (windows) that are used as an input for the CNN-based detector (g). The final position of iris center obtained using the proposed approach (h). The values of distance function are depicted by the level of brightness.

that was presented in [6] due to the fact that the final detection is based on finding one point only with a maximum distance, which does not seem to be reliable enough.

In contrast to the approach from [6], the main steps of our new approach are as follows. In the first step, the candidates for iris center are quickly determined. In the second step, the most probable centre is determined among the candidates by making use of a traditional convolutional neural network. Rapidly filtering out the points that do not have a chance to become the iris center speeds up the whole algorithm, which is often required. In addition to this, the first step also contributes to the successfulness of recognition since the neural network is asked to decide only certain specific pixel configurations in image. In the subsequent paragraphs, this general idea is presented in more details.

In the first step, we follow the approach presented in [6] that has been briefly repeated at the beginning of this section. Since, in the case of the method presented here, the goal of the first step is only to determine the candidates (not to determine the final position of the iris center directly), we may simplify the algorithm presented in [6], which is desirable since the first step should be fast. We do the following: Instead of measuring the distances from the four corners, which was done in the original method, we compute the distances only from two

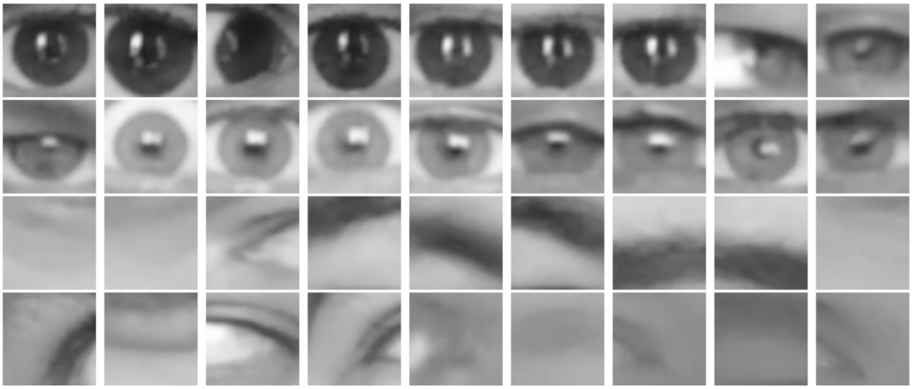

Fig. 3. An example of iris and non-iris images.

cornes with the hope that the subsequent use of CNN will compensate for this simplification. We use the top left and bottom right corner, see Fig. 2(b), (c). For the same reason, a smaller kernel size may be used in convolution smoothing the difference between the distances from the center and the mean of the distances from the corners (see Fig. 2 again), i.e. less aggressive smoothing is used. We note that the expectations we mention here will also be confirmed experimentally in Sect. 4.

Before carrying out the second step, suppose that the CNN-based classifier is trained with a sufficient amount of training iris and non-iris images (Fig. 3). In the second step, the distance differences produced in the first step are subjected to thresholding. It means that the position is verified by CNN only if the distance value is big enough at that point; a window (centered at the point that is being verified) of the gray-scale image is used by CNN (Fig. 2(g)). Finally, the location with the best response of CNN-based detector represents the final iris position (Fig. 2(h)).

The main advantages of this approach can be summarized as follows. Since, the original method uses only the maximum distance value for determining the final position (i.e. feature vector with one value), the combination with CNN-based detector has a positive effect on detection accuracy due to the fact that the model of iris is now described using a more sophisticated feature vector. With the use of coarse iris localization, the CNN classification is carried out only in the neighborhood of points with high distance values to fine-tune the position of iris. This step positively influences the speed of the whole method. Moreover, a smaller number of negative training images can be used if the iris position is approximately detected in advance (CNN will decide only certain specific situations).

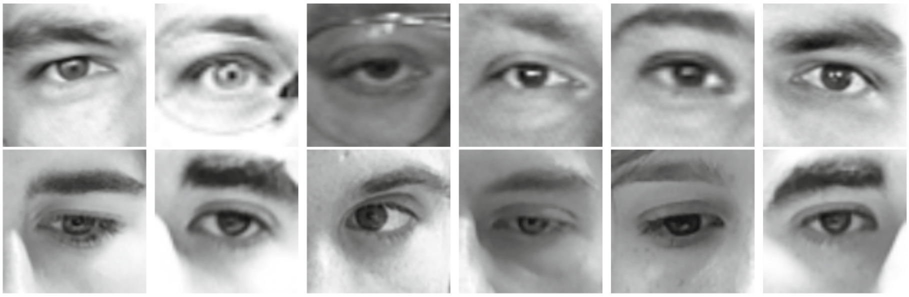

Fig. 4. Examples of eye images used in experiments. The BioID images are in the first row. The GI4E images are in the second row.

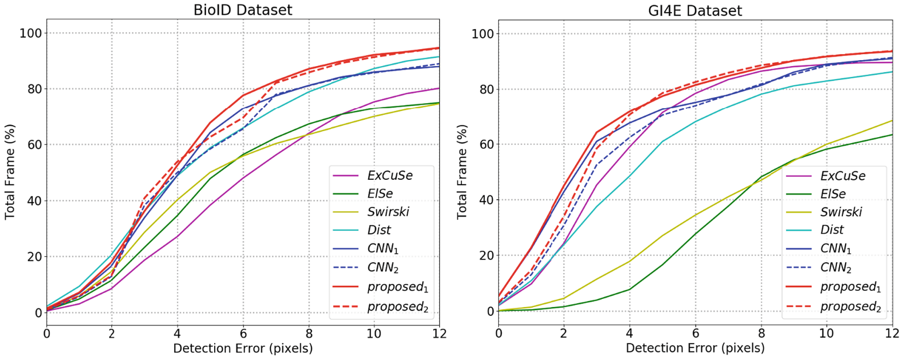

Fig. 5. The cumulative distribution of detection error. The error that is calculated as the Euclidean distance (in pixels) is in the x-axis. The y-axis shows the percentage of frames with the detection error smaller or equal to a specific error. The names of datasets are placed above the pictures.

4 Experiments

As we described in the previous section, after detection of the approximate iris area based on the geodesic distance, the potential points that are selected using the appropriate threshold are further evaluated with the use of CNN. Based on our experiments, we observed that 85% of all points in the eye image can be discarded based on their low distance values. It means that we examine only 15% of all points in the image (the locations with the highest distance values) using CNN. Since we would like to keep a fast computational time of the approach, we use a general architecture of LeNet [12] network for CNN. The network consists of two convolutional layers with the depth of 6 and 16, respectively, and a 5×5 filter size with a 1×1 stride. Each of the layers is followed by a rectified linear activation function. Thereafter, a max pooling layer with a window size of 2×2 and with a 2×2 stride is added; the last two layers are fully connected. We used stochastic gradient descent with the learning rate of 0.01 annealed to 0.0001

To compute the recognition score (confidence), we use the soft-max layer, and 32 × 32 grayscale images are used as an input. The implementation of CNN is based on Dlib [11]. The training set consists of 4600 iris images and 4600 non-iris images that were manually extracted from our eye image data (Fig. 3). It is important to note that the number of training images is low due to the fact that the geodesic distance is used to find the preliminary iris location, and the CNN-based detector is used to refine the final iris position. Therefore, the negative training data were obtained around the iris location only.

We examine two configurations of the presented approach. In the first configuration, we use the CNN detector that evaluates the neighborhood of every point after the distance thresholding (15% of all points). The method with this configuration is denoted as $proposed_1$ in the following experiments. We also created a faster version of our method in which only every fourth point is examined after distance thresholding. This method is referred to as $proposed_2$. The size of extracted area around each point is 32 × 32 pixels in both variants.

To compare the proposed algorithm to the state-of-the-art methods, we have chosen the following methods. Namely *ElSe*, *ExCuSe*, *Swirski*, the original distance method (denoted as *Dist*), and two CNN-based iris detectors: CNN_1 and CNN_2. In the first CNN-based detector (CNN_1), we used a sliding window technique applied to the entire input eye image with one pixel stride, and the stride of four pixels is used in the second detector (CNN_2). The size of sliding window is 32 × 32 pixels in both variants (i.e. 32 × 32 grayscale images are used as an input). The architecture and training process of networks are the same as in the proposed method. It is worth mentioning that *ElSe*, *ExCuSe*, and *Swirski* were primarily developed to work with images acquired by head-mounted cameras, however, the experiments in [1] show that the methods can be used in the images captured with the use of remote sensors as well. We also experimented with the parameters of particular methods. For *ElSe*, we directly used the setting for remotely acquired images published by the authors of the algorithm.

To evaluate the methods, we used two public datasets; BioID [9] and GI4E [15]. The BioID dataset contains 1521 images with the resolution of 384 × 286 pixels. The GI4E database contains 1339 images with the resolution of 800 × 600. From both datasets, the eye regions are selected based on the provided ground truth data of eye corner positions. It is important to mention that the eye images from datasets are purposely extracted with the eyebrow to test the methods in complicated conditions. The size of each extracted eye image (from both datasets) is 100 × 100 pixels in the following experiments. Example images of the GI4E and BioID datasets that are used for experiments are shown in Fig. 4.

In Table 1, the detection results and average times of methods are shown. We note that the average time for processing one eye region was measured on an Intel core i3 processor (3.7 GHz) with NVIDIA GeForce GTX1050. The errors are calculated as the Euclidean distance between the ground truth of iris center and the center provided by the particular detection method. In Fig. 5, we also provide the resulting plots of detection results. In the plots, the cumulative

Table 1. The detection results of methods.

	BioID Mean error (pixels)	GI4E Mean error (pixels)	Time per region (ms)
$proposed_1$	4.97	4.09	18
$proposed_2$	5.36	4.35	9
$Dist$	5.51	5.58	10
CNN_1	6.41	4.65	240
CNN_2	6.34	4.92	15
$ElSe$	10.50	12.72	16
$ExCuSe$	11.00	7.10	8
$Swirski$	10.43	11.10	10

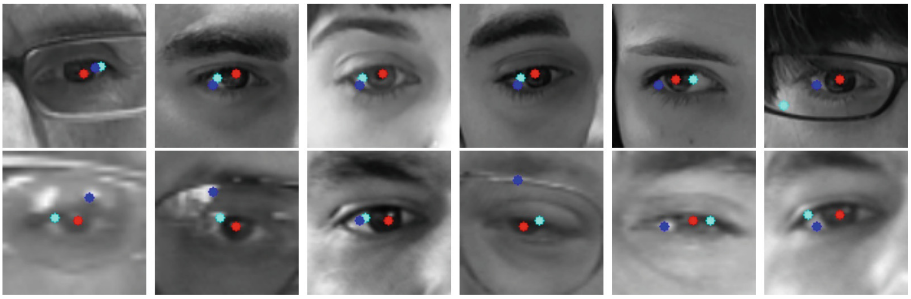

Fig. 6. Examples of images in which the proposed method performs better compared to other tested methods. The results of methods are distinguished by color: $proposed_2$ - red, CNN_2 - blue, $Dist$ - cyan. The first row: GI4E dataset, the second row: BioID dataset. (Color figure online)

distribution of detection error is shown (i.e. the figures show the percentage of frames with the detection error smaller or equal to a specific value).

Based on the results, we can conclude that the proposed method achieved very stable results and outperforms all methods in the images of both datasets. For BioID datasets, the average detection error of proposed method ($proposed_1$) is 4.97 pixels. It means that the presented method also outperforms the original method ($Dist$) in the area of detection accuracy (4.97 vs. 5.51). The faster variant of our method ($proposed_2$) also achieved promising results (5.36). It is worth mentioning that the CNN-based detectors achieved good detection score (6.41 and 6.34), however, the detection time is unnecessarily long in the first variant of CNN (CNN_1). The situation is better in the second faster variant of CNN detector (CNN_2), unfortunately, the detection error is bigger than in the faster variant of proposed approach (6.34 vs 5.36). Based on the results in Fig. 5, it can be observed that the proposed method is able detect approximately 90% of all frames with detection error smaller than 8 pixels. Even in the case of GI4E datasets, the proposed detectors achieved smaller errors than all tested methods (4.09 and 4.35). This situation can also be seen in Fig. 5.

In summary, our results show that the proposed method outperforms the main competitors: the original method presented in [6] and the iris detectors based on CNN. The proposed method that combines CNN with the distance-based preprocessing also achieved the promising time needed for processing one eye region (9 ms in $proposed_2$). Figure 6 shows several cases in which our method works better compared to other tested methods (namely, the main competitors: CNN_2 and $Dist$). Based on the results in Fig. 6, it may be said that the common errors are caused by the presence of glasses and reflections. However, the proposed method is better in such cases than the other tested methods.

5 Conclusion

In this paper, we proposed a new approach for iris center localization. The approach combines the geodesic distance with a convolutional neural network. Firstly, the geodesic distance is used to determine the areas possibly containing the iris. CNN is then used for the final decision. The proposed approach was evaluated and compared with the state-of-the-art methods on two publicly available datasets. Based on the experimental results, we can conclude that the proposed method achieved better recognition performance and a reasonable computational time when compared to the existing methods. We leave the deeper experiments with another architectures of CNN for future work.

Acknowledgments. This work was partially supported by Grant of SGS No. SP2019/71, VŠB - Technical University of Ostrava, Czech Republic.

References

1. Fuhl, W., Geisler, D., Santini, T., Rosenstiel, W., Kasneci, E.: Evaluation of state-of-the-art pupil detection algorithms on remote eye images. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, UbiComp 2016, pp. 1716–1725. ACM, New York (2016). https://doi.org/10.1145/2968219.2968340. http://doi.acm.org/10.1145/2968219.2968340
2. Fuhl, W., Kübler, T., Sippel, K., Rosenstiel, W., Kasneci, E.: ExCuSe: robust pupil detection in real-world scenarios. In: Azzopardi, G., Petkov, N. (eds.) CAIP 2015. LNCS, vol. 9256, pp. 39–51. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23192-1_4
3. Fuhl, W., Santini, T., Kasneci, G., Kasneci, E.: PupilNet: convolutional neural networks for robust pupil detection. CoRR abs/1601.04902 (2016). http://arxiv.org/abs/1601.04902
4. Fuhl, W., Santini, T., Kasneci, G., Rosenstiel, W., Kasneci, E.: PupilNet v2.0: convolutional neural networks for CPU based real time robust pupil detection. CoRR abs/1711.00112 (2017). http://arxiv.org/abs/1711.00112
5. Fuhl, W., Santini, T.C., Kübler, T.C., Kasneci, E.: Else: ellipse selection for robust pupil detection in real-world environments. CoRR abs/1511.06575 (2015). http://arxiv.org/abs/1511.06575

6. Fusek, R.: Pupil localization using geodesic distance. In: Bebis, G., et al. (eds.) ISVC 2018. LNCS, vol. 11241, pp. 433–444. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03801-4_38
7. George, A., Routray, A.: Fast and accurate algorithm for eye localisation for gaze tracking in low-resolution images. IET Comput. Vis. **10**(7), 660–669 (2016). https://doi.org/10.1049/iet-cvi.2015.0316
8. Javadi, A.H., Hakimi, Z., Barati, M., Walsh, V., Tcheang, L.: Set: a pupil detection method using sinusoidal approximation. Front. Neuroeng. **8**, 4 (2015). https://doi.org/10.3389/fneng.2015.00004. https://www.frontiersin.org/article/10.3389/fneng.2015.00004
9. Jesorsky, O., Kirchberg, K.J., Frischholz, R.W.: Robust face detection using the hausdorff distance. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 90–95. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45344-X_14
10. Kacete, A., Royan, J., Seguier, R., Collobert, M., Soladie, C.: Real-time eye pupil localization using Hough regression forest. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–8, March 2016. https://doi.org/10.1109/WACV.2016.7477666
11. King, D.E.: Dlib-ml: a machine learning toolkit. J. Mach. Learn. Res. **10**, 1755–1758 (2009)
12. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
13. Li, D., Winfield, D., Parkhurst, D.J.: Starburst: a hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005) - Workshops, pp. 79–79, June 2005. https://doi.org/10.1109/CVPR.2005.531
14. Świrski, L., Bulling, A., Dodgson, N.: Robust real-time pupil tracking in highly off-axis images. In: Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA 2012, pp. 173–176. ACM, New York (2012). https://doi.org/10.1145/2168556.2168585. http://doi.acm.org/10.1145/2168556.2168585
15. Villanueva, A., Ponz, V., Sesma-Sanchez, L., Ariz, M., Porta, S., Cabeza, R.: Hybrid method based on topography for robust detection of iris center and eye corners. ACM Trans. Multimedia Comput. Commun. Appl. **9**(4), 25:1–25:20 (2013). https://doi.org/10.1145/2501643.2501647. http://doi.acm.org/10.1145/2501643.2501647
16. Zhang, X., Sugano, Y., Fritz, M., Bulling, A.: Appearance-based gaze estimation in the wild. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4511–4520, June 2015. https://doi.org/10.1109/CVPR.2015.7299081