End-to-End Iris Segmentation Using U-Net

Conference Paper · July 2018

DOI: 10.1109/IWOBI.2018.8464213

4 authors:

Jus Lozej

3 PUBLICATIONS **6** CITATIONS

SEE PROFILE

Blaž Meden

University of Ljubljana

17 PUBLICATIONS **47** CITATIONS

SEE PROFILE

Vitomir Štruc

University of Ljubljana

138 PUBLICATIONS **1,267** CITATIONS

SEE PROFILE

Peter Peer

University of Ljubljana

118 PUBLICATIONS **1,250** CITATIONS

SEE PROFILE

End-to-End Iris Segmentation using U-Net

Juš Lozej*, Blaž Meden*, Vitomir Štruc†, Peter Peer*
*Faculty of Computer and Information Science, University of Ljubljana, Slovenia
E–mail: jlozej@gmail.com, blaz.meden@fri.uni-lj.si, peter.peer@fri.uni-lj.si
†Faculty of Electrical Engineering, University of Ljubljana, Slovenia
E-mail: vitomir.struc@fe.uni-lj.si

***Abstract*—Iris segmentation is an important research topic that received significant attention from the research community over the years. Traditional iris segmentation techniques have typically been focused on hand-crafted procedures that, nonetheless, achieved remarkable segmentation performance even with images captured in difficult settings. With the success of deep-learning models, researchers are increasingly looking towards convolutional neural networks (CNNs) to further improve on the accuracy of existing iris segmentation techniques and several CNN-based techniques have already been presented recently in the literature. In this paper we also consider deep-learning models for iris segmentation and present an iris segmentation approach based on the popular U-Net architecture. Our model is trainable end-to-end and, hence, avoids the need for hand designing the segmentation procedure. We evaluate the model on the CASIA dataset and report encouraging results in comparison to existing techniques used in this area.**

***Index Terms*—Iris, segmentation, deep learning, convolutional neural networks (CNN), U-Net**

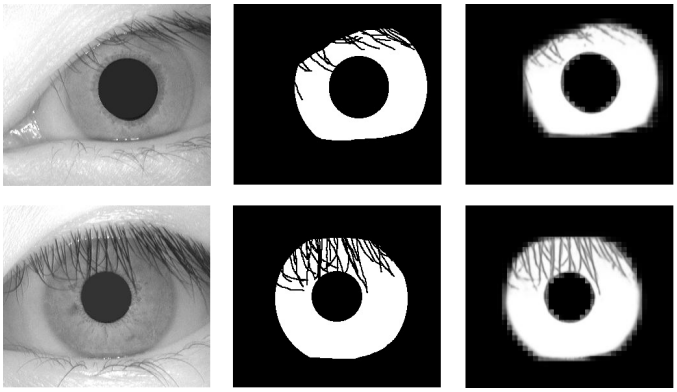

Fig. 1: Sample segmentation results obtained with the U-Net model. The figure shows (from left to right): sample input iris images, the ground truth, the results generated by U-Net.

I. Introduction

Biometric systems based on iris recognition have proven to be very reliable and accurate. Despite the rising popularity of deep learning models, iris recognition systems that utilize deep learning are still relatively rare. Current approaches use deeply learned features in conjunction with a conventional pipeline [1], [2] or exploit a pre-trained model that is fine tuned to be used for iris recognition purposes [3].

One crucial step in iris recognition systems is the segmentation of the iris region from the input image. This step has traditionally been solved using manually designed segmentation techniques [4] and considerable performance has already been achieved on numerous datasets of variable quality. However, with the success of deep-learning models for other vision problems, researchers are increasingly looking into convolutional neural networks (CNNs) to further improve on the performance of existing iris segmentation techniques [5]–[7].

In this paper we contribute to the recent body of work that aims to develop deep learning models for iris-recognition pipelines and study the utility of U-Net [8], a deep convolutional neural network (CNN) commonly used for image translation tasks, for the problem of iris segmentation. Specifically, we develop an end-to-end iris segmentation procedure based on the U-Net model and explore how different model depth and the use of batch normalization layers affects segmentation performance. We conduct experiments on the CASIA 1 dataset [9] and show that U-Net not only ensures highly competitive segmentation results, but also outperforms competing techniques from the literature. A few sample results obtained with the trained U-Net model are presented in Fig. 1.

The main contributions of this paper are:

- We present an end-to-end deep-learning model for iris segmentation based on the U-Net architecture and make it publicly available to the research community through https://github.com/jus390/U-net-Iris-segmentation, and
- We study the impact of different hyper-parameters on the segmentation performance of the trained U-Net model.

The rest of the paper is structured as follows: In Section II we briefly review the related work of relevance to this paper. In Section III we describe the U-Net architecture and procedure used to train the model parameters. In Section IV we present the results of our experiments and discuss our main findings. We conclude the paper in Section V with some final comments and directions for future work.

II. Related work

Iris segmentation techniques represent an active topic of research within the research community [4]. The interest in iris segmentation is fueled by iris recognition technology, where the detection of the region-of-interest (ROI) is the first (and of the most important) step is the overall processing pipeline. By segmenting the iris from the input image, irrelevant data that would otherwise interfere with the success of the recognition process is removed. Additionally, the segmentation step makes

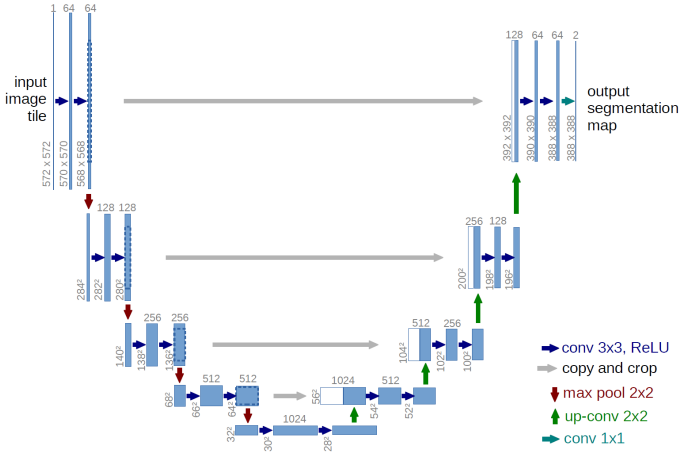

Fig. 2: Illustration of the U-Net architecture with a depth of 4. The model relies on an encode-decoder architecture and uses copy-and-crop operations to propagate information from the encoder layers to the corresponding layers in the decoder. Image taken from [8].

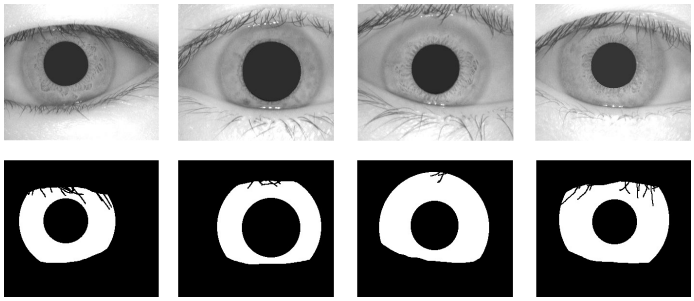

Fig. 3: Illustration of the training data. The top row shows sample images from the CASIA dataset, the bottom row shows the annotated (binary) segmentation ground truth.

it possible to normalize the iris region and extract discriminative features from well aligned iris samples.

As suggested by Rot et al. [10], existing approaches to iris segmentation include Daugman's integro-differential operator [11], active contour models [12] and clustering algorithms [13] as well as techniques exploiting gradient (edge) information [14]–[16], variants of the Hough transform [17], [18] and others [4].

Recent approaches for iris segmentation, which are also of relevance to this work, consider iris-segmentation as a classification problem, where the goal is to assign each pixel either to the *iris* or *non-iris* class - see, for example, [5]–[7]. Our model is related to these techniques and also uses CNNs to segment the input images and identify pixels that belong to the iris. A desirable characteristics of our model is that the need for hand-crafted features, information fusion approaches and specific image transforms is bypassed. Instead the entire model is trained end-to-end based solely on appropriately annotated training data.

III. Methods

In this section we describe the U-Net model, it's architecture and procedure used to train the model for iris segmentation.

A. The U-net model

Overview: The U-Net [8] model represents a popular CNN architecture for solving biomedical problems (e.g. segmenting different kinds of cells and detecting boundaries between very dense cell structures) and other image translation tasks [19], [20]. The main advantage of this model is it's ability to learn relatively accurate models from (very) small datasets, which is a common problem for data-scarce computer-vision tasks, including iris segmentation.

Model architecture: U-Net uses an encoder-decoder architecture as illustrated in Fig. 2. The architecture is devided into corresponding encoder and decoder convolutional layers. In Fig. 2 the left side of the model is the encoder path and the left side is the decoder path. The encoder follows a typical CNN architecture, as popularized by the VGG network [21] consisting of two 3×3 convolutional layers followed by a ReLu activation layer with max pooling. Each down-sampling step of the encoder doubles the amount of feature channels and decreases the image resolution by half. The decoder part than up-samples the feature maps of the lower layer while also concatenating and cropping the output of the encoder part of the same depth. This process ensure that information is propagated from the encoder to the decoder at all scales and no information is lost during the down-sampling operations in the encoder. The final layer of the network is a 1×1 convolutional layer that mixes the output channels of the preceding layer and produces the segmentation maps (one per class - iris vs. non-iris) that represent the output of the U-Net model (Note that for binary segmentation problems the masks are complements of each other).

Training details: To train the model for iris segmentation, we first manually annotate a small set of iris images and then learn the parameters of the U-Net model using adaptive moment estimation (Adam) and binary cross-entropy as our training objective. For the annotation procedure we use an in-house tool that first uses a pair of ellipses to mark the iris (and pupil) region and then relies on a manual markup at the pixel level to account for the eyelids, eyelashes and other eye details that do not belong to the iris. The result of this annotation procedure is a detailed (pixel-level) markup of the iris, where the main iris region is bounded by smooth parameterized second-order curves, while eye artifacts are masked with detailed masks as shown in Fig. 3. Once the model is trained, it takes iris images at the input and returns corresponding segmentation masks at the output.

IV. Experiments and Results

In this section we present the results of our experiments. We first discuss the experimental dataset and protocol used for the experiments, then elaborate on the network training and finally comment on the results of our assessment.

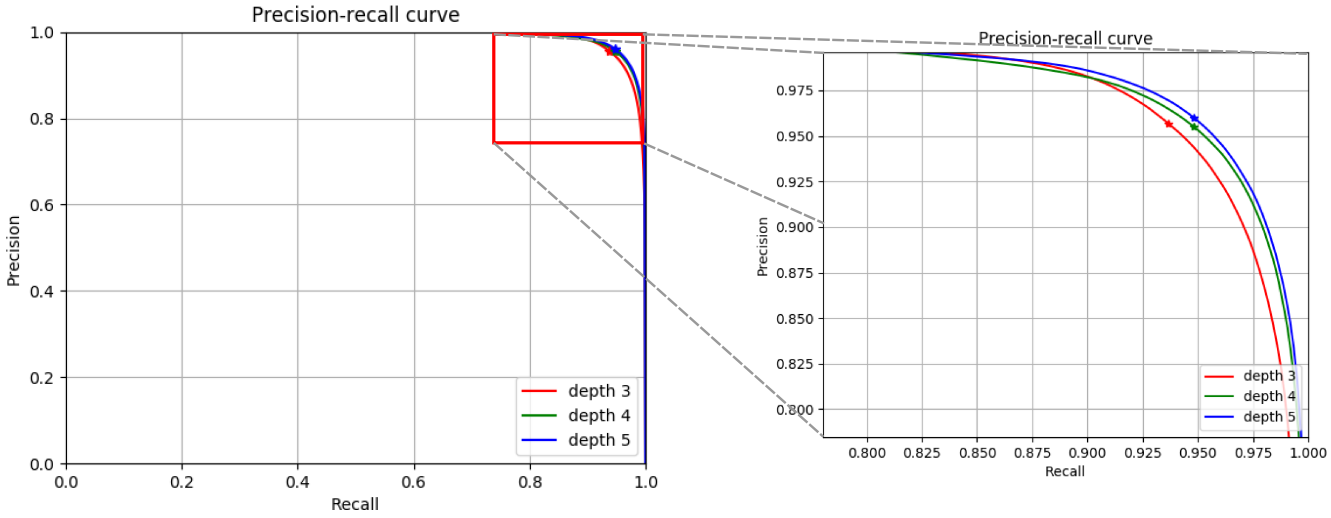

Fig. 4: Precision-recall curves for U-Net models with different depths (without batch normalization). Stars indicate the threshold with the best precision-recall ratio. The right side figure shows a zoomed-in version of the figure on the left. Note that increasing the depth of the model contributes towards better segmentation performance. The figure is best viewed in color.

A. *Dataset and experimental protocol*

For our experiments we annotate 200 images from the original CASIA Ver. 1 database [9]. The images correspond to 107 distinct subject and are of size 320×280 pixels. We use a 80% and 20% split to construct train and test datasets, thus, 160 images are used for training the U-Net model, and the remaining 40 images are used to compute performance metrics. We report results in terms of precision, recall and the intersection-over-union, similar to [22], and also present precision-recall curves where applicable. Since, iris segmentation is treated as a binary classification problem by the U-Net model, we also report accuracy values for the segmentation procedure.

B. *Training details*

Prior to training all images are reshaped into square form of 320×320 pixels through padding. The ground truth masks are padded with zeros and the actual iris images are padded with the mean intensity value. Both the masks and the images are also normalized to values between 0 and 1. All models are trained using the Adam optimizer with a learning rate of 10^{-4} and zero decay. During training no augmentations are used. The models are trained for 10 epochs with the deepest (at depth 5) model taking roughly 15 min to finish training.

The models were implemented in python using Keras [23] high-level neural network API with Tensorflow [24] as its backend. The memory consumption was limited to 95% of memory. As mentioned before the best performing model is readily available through https://github.com/jus390/U-net-Iris-segmentation.

C. *Experiments*

Performance and hyper-parameter impact: In our first series of experiments we first evaluate the (iris) segmentation performance of the trained U-Net model and assess the impact of different hyper-parameters, i.e., the impact of model depth and use of batch normalization. Increasing the depth of the model corresponds to adding an additional convolutional layer to the encoder as well as to the decoder, thus, for a depth of 5 the model comprises 5 layers in the encoder and 5 in the decoder. We use the code provided by the authors of U-Net for all experiments.

TABLE I: Accuracy for different deep U-net architectures with and without batch normalization (BN)

Depth	Batch normalization (BN)	Accuracy
3	No	97.77%
3	Yes	96.89%
4	No	97.83%
4	Yes	96.85%
5	No	97.79%
5	Yes	96.90%

The results of the first series of experiments are generated on our test set of 200 images and presented in the form of precision-recall curves in Fig. 4 and in the form of accuracy values in Table I. We see that all models exhibit increased performance (in terms of accuracy) with the increase of depth. A similar behavior can be observed from the precision-recall curve. Here, the model at depth 5 performs the best, followed closely by the model at depth 4. As expected, U-Net at depth 3 performs the worst, however, the performance of this model is slightly better then the performance of the model of depth 4 at lower thresholds. From this we can conclude that deeper architectures should give slightly better performance, however, the performance differences we observe are minimal. Because of hardware constraints we weren't able to evaluate

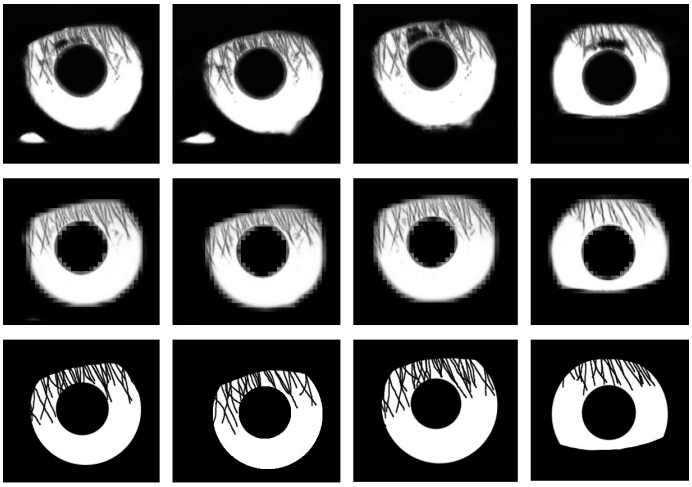

Fig. 5: Comparison of segmentation results with batch normalization(top), without batch normalization (middle) and the ground truth (bottom). Results are shown for U-Net model of depth 4. The models without batch normalization perform slightly better than the models without batch normalization.

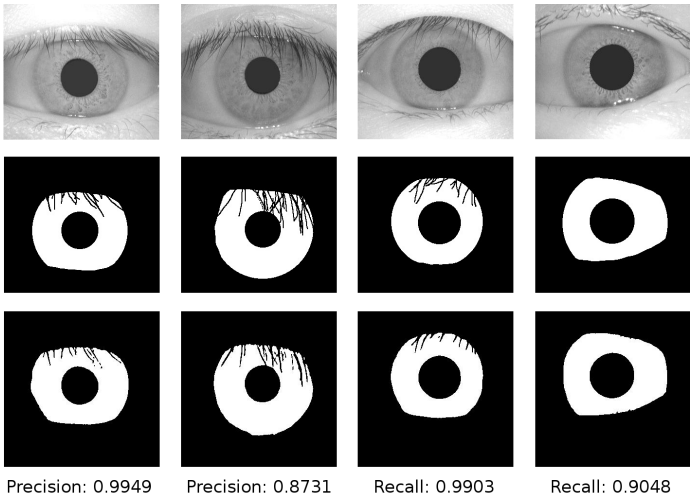

Fig. 6: Selected examples of the best and worst segmentation result obtained with the U-Net model of depth 5. Original iris images (top), ground truth annotations (middle), segmentation results (bottom).

TABLE II: Time and space complexity for U-Net models of different depths.

Depth	Time/image	Maximum memory used
3	45 ms/image	5173 MB
4	60 ms/image	5181 MB
5	102 ms/image	5185 MB

the architecture at higher depths.

The models without batch normalization generally perform slightly better then those with batch normalization. When visual inspecting the results of the models with batch normalization we observe noticeable smudges on the surface of the iris, which are the cause of the lower performance. These anomalies can be seen in Fig. 5.

What is interesting to note is that the best performing models without batch normalization generate iris regions that seem to be bounded (roughly) by second-order curves (i.e., ellipses), which is consistent with the way the images were annotated. This empirical results suggests that the network implicitly learned to estimate ellipses and circles as boundaries for the iris region.

Time and space efficiency: We use a desktop PC for the experiments running an Intel I7-2600k processor with 8 GB of RAM and a Nvidia GTX-1060 6 GB GPU. All models are tested on the GPU with the use of CUDA. Using this hardware configuration we assess the speed of each model by feeding them the test images in a loop one at a time. During this experiment we also measure the utilization of the GPU's resources. The results are shown in Table II

As can be seen all models used about the same amount of memory. The difference being the time it took for each model to segment the sample images. The time expectedly increased with depth. During testing we also monitored the GPU usage. All models reached a 100% GPU usage.

Qualitative evaluation: We visually examined some of the best and worst cases in terms of precision and recall values for the best performing U-Net model (i.e., depth 5, without batch normalization) using a preselected threshold of 0.7 that provided the best average ratio between recall and precision. The example with the worst precision proved to be difficult for the model as it has a complex pattern of intertwining eyelashes, while the example with the worst recall seemed to have left out some parts of the iris and also had a less stable edge.

Comparison with competing methods: We compare our method to four baseline methods. The first baseline method (denotes as Masek [25]) uses the Hough transform to localize the iris boundaries, then uses thresholding to remove eyelashes and specular reflections and lastly a horizontal line detector, which also relies on the Hough transform, to detect the eyelid boundaries. These method was chosen as its source code is publicly available. The other baseline methods are part of the University of Salzburg Iris Toolkit [26]. The toolkit contains many different iris preprocessing, feature extraction and feature comparison methods that can be interchanged and combined in order to create a multitude of Iris pipelines. Naturally the toolkit also contain a number of segmentation methods, which we utilized as baseline methods. The method (Caht) uses a contrast adjusted Hough transform. It modifies the contrast of the input image in order to increase the variation between the iris and the sclera making edges more apparent. Similarly to Masek's method it also exploits Hough transform to localize the limbic and papillary boundaries. The other two methods (Wahet and Ifpp) are divided into two steps: center localization and boundary localization. Both methods first localize the center of the pupil using the adaptive Hough transform. This center is then used as an origin point from

TABLE III: Comparison to competing methods from the literature. Average precision, recall and Intersection over Union (IoU) at threshold with the best ratio between precision and recall

Method	Depth	Threshold	Avg. Precision	Avg. Recall	Avg. IoU
Masek [25]	n/a	n/a	0.892 ± 0.072	0.876 ± 0.105	0.792 ± 0.107
Caht [26]	n/a	n/a	0.899 ± 0.059	0.889 ± 0.073	0.807 ± 0.074
Ifpp [26]	n/a	n/a	0.909 ± 0.060	0.856 ± 0.076	0.788 ± 0.094
Wahet [26]	n/a	n/a	0.920 ± 0.057	0.872 ± 0.076	0.809 ± 0.076
U-net (this work)	3	0.66	0.937 ± 0.030	0.957 ± 0.021	0.899 ± 0.035
U-net (this work)	4	0.66	0.948 ± 0.026	0.955 ± 0.022	0.908 ± 0.031
U-net (this work)	5	0.70	$\mathbf{0.948 \pm 0.025}$	$\mathbf{0.960 \pm 0.019}$	$\mathbf{0.912 \pm 0.031}$

which both methods start their search. Wahet first transforms the image into polar coordinates. Then an Ellipso-polar transform is used to derive the limbic and papillary boundary candidates. The transformation first determines the maximum energy horizontal line, while maximizing the vertical polar gradient. This results in a smoothed curve, which is projected back onto Cartesian coordinates and points are fitted to an oriented ellipse. This is done twice, once for each boundary. In contrast Ifpp uses twofold pulling and pushing with the use of the Fourier-based trigonometry for boundary localization. The toolkits implementations of these methods segment the eyelids, eyelashes and reflections after normalization. As our test required non-normalized binary masks we post-processed the exported masks using Masek's method e.g. line detection for eyelids and thresholding for reflection and eyelash segmentation.

As we can see from Table III, the U-Net model performs the best in terms of average precision, average recall and average intersection-over-union (IoU). Even the models with the lowest depth outperforms all considered baseline techniques suggesting that convolutional neural networks represent a viable alternative to established techniques from the literature.

V. Conclusion

We have presented a U-Net based procedure for iris segmentation. The CNN-based segmentation model proved to be very successful at segmenting the iris, while also outperforming all considered baseline methods. The model didn't require a great amount of training data and worked well without the use of data augmentation during training. In conclusion the use of deep learning methods in iris recognition may provide an increase in performance in an area of biometrics compared to conventional methods.

Acknowledgements

This research was supported in parts by ARRS (Slovenian Research Agency) Research Program P2-0250 (B) Metrology and Biometric Systems, ARRS Research Program P2-0214 (A) Computer Vision, and the RS-MIZŠ and EU-ESRR funded GOSTOP. One of the GPUs used for this research was donated by the NVIDIA Corporation.

References

[1] Z. Zhao and A. Kumar. Towards more accurate iris recognition using deeply learned spatially corresponding features. In *ICCV*, pages 22–29, 2017.

[2] X. Tang, J. Xie, and P.a Li. Deep convolutional features for iris recognition. In *CCBR*, pages 391–400. Springer, 2017.

[3] M. Arsalan, H.G. Hong, R.A. Naqvi, M.B. Lee, M.C. Kim, D.S. Kim, C.S. Kim, and K.R. Park. Deep learning-based iris segmentation for iris recognition in visible light environment. *Symmetry*, 9(11):263, 2017.

[4] I. Nigam, M. Vatsa, and R. Singh. Ocular biometrics: A survey of modalities and fusion approaches. *Information Fusion*, 26:1–35, 2015.

[5] E. Jalilian, A. Uhl, and R. Kwitt. Domain adaptation for cnn based iris segmentation. *BIOSIG*, 2017.

[6] M. Arsalan, H. Gil Hong, R.A. Naqvi, M. B. Lee, M. C. Kim, D. S Kim, C. S. Kim, and K. R. Park. Deep learning-based iris segmentation for iris recognition in visible light environment. *Symmetry*, 9(11):263, 2017.

[7] E. Jalilian and A. Uhl. Iris segmentation using fully convolutional encoder–decoder networks. In *Deep Learning for Biometrics*, pages 133–155. Springer, 2017.

[8] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[9] Casia Iris V1. http://biometrics.idealtest.org/dbDetailForUser.do?id=1. Accessed: 2018-05-06.

[10] P. Rot, V. Štruc, and P. Peer. Deep multi-class eye segmentation for ocular biometrics. In *IWOBI*, 2018.

[11] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE TPAMI*, 15(11):1148–1161, 1993.

[12] J. Daugman. New methods in iris recognition. *IEEE TSMC-B*, 37(5):1167–1175, 2007.

[13] T. Tan, Z. F. He, and Z. Sun. Efficient and robust segmentation of noisy iris images for non-cooperative iris recognition. *IVC*, 28(2):223–230, 2010.

[14] M. De Marsico, M. Nappi, and R. Daniel. Is_is: Iris segmentation for identification systems. In *ICPR*, pages 2857–2860, 2010.

[15] G. Sutra, S. Garcia-Salicetti, and B. Dorizzi. The viterbi algorithm at different resolutions for enhanced iris segmentation. In *ICB*, pages 310–316, 2012.

[16] H. Li, Z. Sun, and T. Tan. Robust iris segmentation based on learned boundary detectors. In *ICB*, pages 317–322, 2012.

[17] J. Koh, V. Govindaraju, and V. Chaudhary. A robust iris localization method using an active contour model and hough transform. In *ICPR*, pages 2852–2856, 2010.

[18] A. Uhl and P. Wild. Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation. In *ICB*, pages 283–290, 2012.

[19] Satellite image segmentation: a workflow with u-net. https://vooban.com/en/tips-articles-geek-stuff/satellite-image-segmentation-workflow-with-u-net/. Accessed: 2018-05-06.

[20] Practical image segmentation with unet. https://tuatini.me/practical-image-segmentation-with-unet/. Accessed: 2018-05-06.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] Ž. Emeršič, L. Gabriel, V. Štruc, and P. Peer. Convolutional encoder–decoder networks for pixel-wise ear detection and segmentation. *IET Biometrics*, 7(3):175–184, 2018.
[23] Keras python library. https://keras.io/. Accessed: 2018-05-06.
[24] TensorFlow neural network api. https://www.tensorflow.org/. Accessed: 2018-05-06.
[25] L. Masek. Matlab source code for a biometric identification system based on iris patterns. *http://people. csse. uwa. edu. au/pk/studentprojects/libor/*, 2003.
[26] C. Rathgeb, A. Uhl, . Wild, and H. Hofbauer. Design decisions for an iris recognition sdk. In *Handbook of Iris Recognition*, pages 359–396. Springer, 2016.