Unsupervised iris image segmentation

Elizaveta Maksimenko, Elena Pavelyeva

Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow, Russia
elizaveta.a.maksimenko@gmail.com, pavelyeva@cs.msu.ru

Keywords: Iris segmentation, Neural networks, Image processing, W-Net, Computer vision, Biometrics.

Abstract

Iris biometrics is considered one of the most accessible and effective biometric characteristics due to its unique patterns and stability over the time. This paper presents a neural network method based on W-Net architecture for iris image segmentation. The use of Daugman’s integro-differential operator enhances the precision and effectiveness of iris image segmentation. The W-Net architecture, utilizing deep learning, accurately isolates the iris region, while the Daugman’s operator ensures robust boundary detection. The model is trained on the CASIA-IrisV4-Interval dataset and effectively handles challenges such as eyelash occlusions. This method opens new possibilities for developing more reliable and accurate biometric identification systems with applications in access control and identity verification.

1. Introduction

Nowadays the world is facing a growing need for security and protection of personal data. In this context, biometric technologies play an increasingly important role. Iris biometrics is one of the most promising biometric technologies, which allows to identify a person by unique iris characteristics (Bhattacharyya et al., 2009). Its advantages include the uniqueness of each iris, its invariability throughout life, non-invasive biometric data collection, high accuracy and ease of use. Due to these advantages, iris biometrics can be integrated into various devices such as smartphones and laptops, making it convenient to use in daily life. The technology uses infrared light to capture the iris image. Infrared light passes through the pupil and reflects off the iris. Since infrared radiation has wavelengths that are outside the visible light spectrum, the dependence on eye color is eliminated. In order to use the iris as a biometric characteristics, it is necessary to accurately extract the iris from the eye image (Fig. 1). Iris segmentation is the process of extracting an iris region from an eye image by defining its outer and inner boundaries while excluding glares, eyelashes, eyelids and other details that may be present in the eye image. Iris image segmentation is a crucial step in iris recognition systems, significantly affecting the overall accuracy and reliability. Despite the effectiveness of current iris-based human recognition methods, their accuracy is highly dependent on the quality of segmentation, as evidenced by many studies. So the development of iris image segmentation algorithms is an important task to achieve higher accuracy and reliability in biometrics.

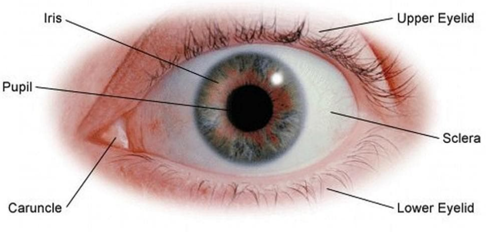


Figure 1. Structure of the eye.

The most common methods for solving the iris boundary extraction problem are the following approaches.

1. Machine learning methods: these methods use a dataset to train machine learning algorithms, which can then classify pixels as iris or background.
2. Geometry-based methods: these methods use geometric properties of the iris to segment the iris. For example, the iris boundary can be described as a circle or ellipse.
3. Active contour methods: in these methods, the iris contour is determined by minimizing the energy associated with the contour.

Generally, the simplest and most effective methods to localize iris boundaries are to find circular regions in the image using the Hough transform (Illingworth and Kittler, 1988) and the integro-differential Daugman’s operator (Daugman, 1993).

The main difficulties in iris segmentation are as follows.

1. The presence of elements that partially conceal the iris (eyelids or eyelashes).
2. The iris may vary in width and shape depending on the lighting and viewing angle.
3. Noise, reflections, and other artifacts may appear in the iris image.
4. In images taken with an infrared camera, the iris may not be very different in intensity from the sclera.

In light of these limitations, neural network architectures such as convolutional neural networks (CNNs) and deep neural networks (DNNs) provide a promising solution for iris segmentation. Deep neural networks have become a key component in many computer vision tasks, including unsupervised learning for semantic image segmentation (Badrinarayanan et al., 2017). They have the ability to automatically extract high-level features from images, allowing them to efficiently adapt to different textures, illumination, and eye structure. Unlike classical methods, neural network architectures minimize the need for

Publisher's note: Copernicus Publications has not received any payments from Russian or Belarusian institutions for this paper.

This contribution has been peer-reviewed.

manual parameter selection and tuning, providing a flexible and more accurate means for iris segmentation under a variety of imaging conditions. This becomes especially critical in modern biometric systems where accuracy plays a key role in successful personal identification.

2. Iris extraction method based on Daugman's integro-differential operator and Hough transform

Images from the (CASIA Iris Image Database, n.d.) database containing 2639 iris images of 280x320 pixels (Fig. 2) were used to implement the neural network method.

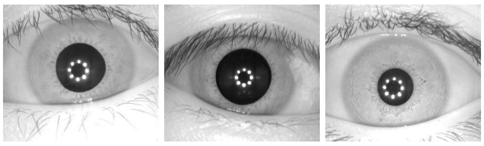
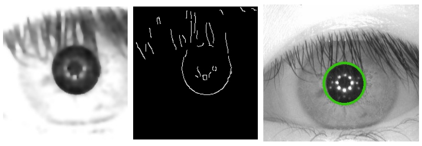


Figure 2. Examples of CASIA-IrisV4-Interval dataset images.

In order to avoid artifacts such as backlight reflection, glares and noise, morphological operations were applied to grayscale images in several stages (erosion and delatation). Further, taking into account the limited size of the pupil and its probable location in the image, an image of the size $(h \cdot 0.2; h \cdot 0.7) \times (w \cdot 0.2; w \cdot 0.7)$, where h and w are the height and width of the image, respectively, and the Canny edge detection operator is applied. Then the Hough transform (Illingworth and Kittler, 1988) was applied to search for the inner boundary circle, and the radius and coordinates of the center of the pupil were obtained (Fig. 3).



(a) (b) (b)

Figure 3. (a) Pre-processed image; (b) pre-processed image after applying the Canny operator; (c) results of pupil boundary extraction.

The outer boundary was searched using the Daugman's (Daugman, 1993) integro-differential operator:

$$\max_{r,x_0,y_0}\left|G_\sigma(r) * \frac{\partial}{\partial r}\int_{r,x_0,y_0}\frac{I(x,y)}{2\pi r}ds\right|, \qquad (1)$$

where $G_\sigma(r) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{r^2}{2\sigma^2}}$ = Gaussian function
$I(x,y)$ = iris image intensity function
s = contour of a circle
(x_0, y_0) = assumed coordinates of the circle center
r = assumed radius of the circle
σ = standard deviation of the Gaussian function
$*$ = convolution operation

The algorithm searches for circular regions in the image. It calculates the mean sum of intensity values lying on the circle as the radius changes. Since often the centers of the inner and outer iris borders do not coincide, the search for the radius and center of the outer border was performed within 10 pixels from the pupil center. Also, knowing that

$$1.25r < R < 5r, \qquad (2)$$

where r = radius of the pupil
R = radius of the outer iris boundary

we applied integration only over the right and left arcs, which are divided into two parts. The size of each arc is calculated using the formula:

$$\gamma = \frac{I \times 90^\circ}{\sum_i I_i}, \qquad (3)$$

where $i \in \{1,2,3,4\}$
I_i = average intensity of the image in the corresponding region

In this way, the overlap of eyelashes and eyelids on the iris image is taken into account, and the regions containing them are less involved in the integration (Fig. 4).

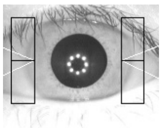
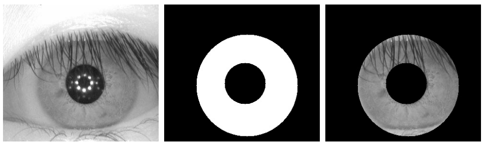


Figure 4. Definition of the outer border of the iris.

The method creates a mask that covers not only the iris, but also parts of eyelashes and eyelids (Fig. 5). This problem may increase the probability of personal identification errors in biometric authentication systems.



(a) (b) (c)

Figure 5. Result of Daugman's method: (a) input image; (b) mask resulting from the method; (c) segmentation result.

3. W-Net architecture

W-Net (Xia and Kulis, 2017) is an autoencoder that is formed by the composition of two networks of the U-Net architecture, namely, the encoder U_{Enc} and the decoder U_{Dec}. In a standard autoencoder configuration, the encoder creates a compact representation of the features of the input image, while the decoder is responsible for reconstructing the original image from this encoded representation. In W-Net, the encoder maps each pixel of the input image into a pixel in the segmentation layer, and the image dimensions do not change during the conversion process. For experiments, the images are scaled to a size of 320x320x1, so the outputs of the encoder are 320x320xK and contain predictions of the probability of each pixel belonging to a class $k \in \{1,2,3,\dots,K\}$.

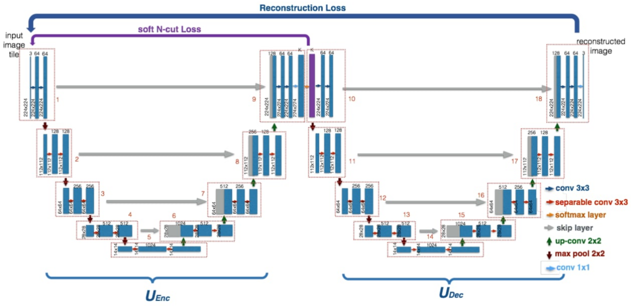


Figure 6. W-Net architecture.

3.1 Modules

The whole architecture consists of 18 modules, highlighted in the image by red rectangles. The first 9 modules belong to the U_{Enc} encoder, the rest belong to the U_{Dec} decoder. Each of these modules consists of two 3x3 convolutional layers followed by a ReLU activation function and packet normalization.

The structure of the two networks of the U-Net architecture represented in the W-Net architecture is almost identical. In general, it is divided into a compressive path and an extensible path (hence the term U-Net). In the compressive path, the modules are connected by 2x2 subsampling layers and the number of channels is doubled at each step of size reduction. In the expanding path, the size increase is performed using 2x2 inverse convolution, which increases the size of each pixel by taking into account the pixel values in the neighborhood. In doing so, the number of channels is halved when moving from one module to another. Finally, to reduce the loss of spatial information due to size reduction, the input of each module in the compressive path is combined with the output of the corresponding module in the expanding path (skip layers).

The differences between the encoder and decoder are as follows: U-Enc is terminated with a 1x1 convolution layer that maps each 64-component feature vector into a vector of K components, where K is the number of classes for image segmentation. The softmax layer ensures that the values are in the range (0,1) and sum to give 1. U-Dec is completed with a 1x1 convolution layer that maps 64-component feature vectors into vectors of 3 components to return an output of the same size as the images in the training set. The goal of W-Net is to generate high quality image segmentations without the presence of labeled data. To achieve this goal, two loss functions are jointly minimized during training to train the network without a teacher. We propose to realize the method based on a convolutional neural network W-Net (Fig. 6) trained without a teacher, using an integro-differential Daugman’s operator.

3.2 Soft Normalized Cut Loss function

The output of U_{Enc} is a normalized dense prediction of size 320 × 320 × K. By using the argmax operation, the prediction for each pixel in K classes is obtained. The global criterion for segmentation is $Ncut_K(V)$ (Shi and Malik, 2000) is calculated as follows:

$$Ncut_K(V) = \sum_{k=1}^{K} \frac{\mathrm{cut}(A_k, V - A_k)}{\mathrm{assoc}(A_k, V)}, \tag{4}$$

$$\mathrm{cut}(A_k, V - A_k) = \sum_{u \in A_k, v \in V - A_k} w(u, v), \tag{5}$$

is the sum of weights of all edges that connect vertices inside the subset A_k with vertices outside this subset;

$$\mathrm{assoc}(A_k, V) = \sum_{u \in A_k, t \in V} w(u, t), \tag{6}$$

where A_k = set of pixels in the segment k
V = set of all pixels
$w(u, v)$ = weight between the pixels u and v

However, since the argmax function is non-differentiable, it is not possible to compute the corresponding gradient during error back propagation. Instead, a Soft Normalized Cut Loss function is defined that is differentiable so that the gradients during error back propagation can be updated.

$$\begin{aligned} J_{\text{soft N-cut}}(V, K) &= \sum_{k=1}^{K} \frac{\mathrm{cut}(A_k, V - A_k)}{\mathrm{assoc}(A_k, V)} \\ &= K - \sum_{k=1}^{K} \frac{\sum_{u \in V} \sum_{v \in V} w(u, v) p(u = A_k) p(v = A_k)}{\sum_{u \in V} p(u = A_k) \sum_{t \in V} w(u, t)}, \end{aligned} \tag{7}$$

where $p(u = A_k)$ = probability that pixel u belongs to class $k \in \{1, 2, 3, \ldots, K\}$

The weight w_{ij} is defined by the following expression:

$$w_{ij} = e^{-\frac{\|F(i) - F(j)\|_2^2}{\sigma_I^2}} \cdot \begin{cases} e^{-\frac{\|X(i) - X(j)\|_2^2}{2\sigma_X^2}}, & \|X(i) - X(j)\|_2^2 < r \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $F(i)$, $F(j)$ = latent representation values for pixels i and j
$X(i)$, $X(j)$ = coordinates of pixels i and j
σ_I^2,σ_X^2 = dispersion parameters
r = threshold value chosen empirically

The weights take into account both the intensity functions of the pixels and their spatial location, which allows us to determine the degree of interaction between them in the context of an image processing task.

3.3 Reconstruction Loss Function

The Reconstruction Loss function trains the encoder to generate representations that contain as much information as possible about the input image. It does this by minimizing the distance between the source image X and the decoder output.

Reconstruction Loss $J_{reconstr}$ is given by the formula:

$$J_{reconstr} = \|X - U_{Dec}(U_{Enc}(X; W_{Enc}); W_{Dec})\|_2^2 \tag{9}$$

where X = original input image
U_{Enc} = encoding function
U_{Dec} = decoding function
W_{Enc} = encoder parameters
W_{Dec} = decoder parameters

In our work, W-Net was trained in two ways:

1. Sequential minimization $J_{\text{soft-N-cut}}$ and J_{reconstr}
2. Minimization of functionality

$$F_1 = \alpha \cdot J_{\text{soft-N-cut}} + J_{reconstr} \tag{10}$$

where $\alpha = \frac{0.01}{K}$

4. Realization

Images were transformed to the size of 320x320x1 and fed to the input of the neural network. Results are shown in Fig. 7.

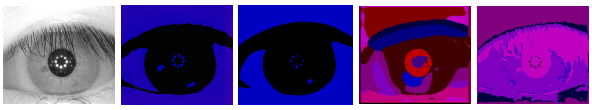


Figure 7. Segmentation results without additional loss: (a) input image; (b) segmentation map with minimization 1, K=2; (c) segmentation map with minimization 2, K=2; (d) segmentation map when minimizing 1, K=16; (e) segmentation map when minimizing 2, K=16.

Successive minimization of loss functions slows down the convergence rate significantly. Moreover, the recovery accuracy decreases and the image is almost not recoverable (Fig. 8). In addition to the iris, eyelashes and other side elements are isolated.

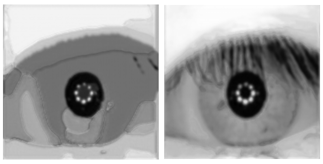


Figure 8. Reconstructed image: (a) with minimization 1; (b) with minimization 2.

4.1 Additional loss function

To improve the segmentation quality, it was proposed to add an additional loss function based on the Daugman’s integro-differential operator.

Binary iris mask images obtained by applying the method described above were prepared for the experiments. Examples of the images are presented in Fig. 9.

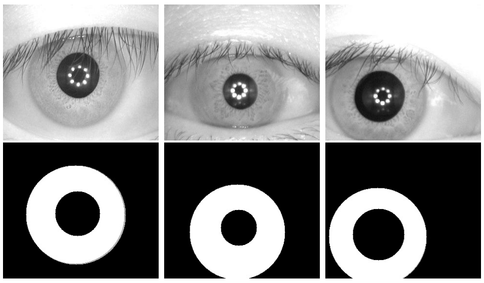


Figure 9. Source images and their corresponding masks.

The input of the neural network is the original image and its mask. Since W_{Enc} produces a segmentation map for K classes, we can attribute the iris class to the first class and consider the loss function at the output of the encoder between the first class and the mask. We will call this loss function D_{Loss}.

The first attempt was to add D_{Loss} to $J_{\text{soft-N-cut}}$ with weight β to minimize sequentially

$$J_{\text{soft-N-cut}} + \beta \cdot D_{Loss}, \text{ then } J_{reconstr} \tag{11}$$

where $\beta \in (0, 10]$ = empirically selected coefficient

To compute D_{Loss}, the following loss functions were considered to evaluate the degree of match between predicted and true masks:

1. Categorical Cross Entropy (CCE)

$$H(y, p) = -\sum_i y_i \cdot \log(p_i), \tag{12}$$

where y = vector of true labels
p = vector of predicted probabilities for each category
i = index by category

2. Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{13}$$

where n = number of observations (sample size)
y_i = true value for the i-th observation
$\hat{y}_i$ = predicted value for the i-th observation

3. Binary Cross-Entropy Loss with Logits (BCEWithLogitsLoss)

$$-\frac{1}{N}\sum_{i=1}^{N}(y_i \cdot \log(\sigma(\hat{y}_i)) + (1 - y_i) \cdot \log(1 - \sigma(\hat{y}_i))) \tag{14}$$

where N = number of items in the batch
y_i = true label for item i
$\hat{y}_i$ = raw values that the model generates before deciding on class membership using the activation function for item i
σ = sigmoidal activation function

For K = 2 and any loss function, the network is overfitted and images close to binary masks are obtained (Fig. 10), i.e., part of the eyelashes and part of the eyelid are misclassified as iris.

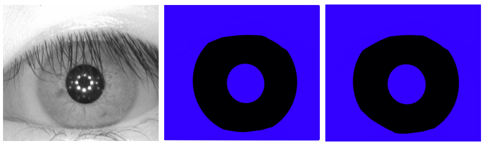


Figure 10. K = 2 and Categorical Cross Entropy: (a) input image; (b) β = 1; (c) β = 0.1.

It was concluded that the number of classes K = 2 is insufficient for this task, and that it will work better at $K > 2$ with the subsequent possibility of combining some classes. We empirically chose K = 16. Therefore, we modified the loss function 2 and the problem is reduced to minimization of the functional

$$F_2 = \alpha \cdot J_{\text{soft-N-cut}} + \beta \cdot D_{Loss} + J_{reconstr} \tag{15}$$

Segmentation results when minimizing 15 in are shown in Fig. 11.

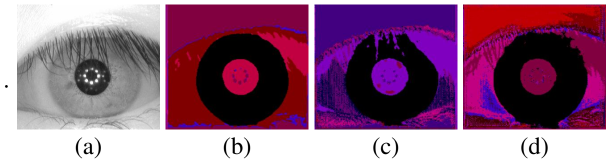


Figure 11. Demonstration of results when K = 16, $\beta = 0.5$ and minimizing 15: (a) input image; (b) CCE; (c) BCEWithLogits; (d) MSE;

BCEWithLogits performed better than other loss functions for this task while minimizing (15), partially separating the eyelashes from the iris, while CCE and MSE lead to overfitting and large segmentation errors. In order to increase the accuracy of the model it was necessary to use fine-tuning tools.

4.2 Fine-tuning

Fine-tuning is a neural network training method that adapts a pre-trained model to a new task. This method is particularly useful when a limited amount of training data is available for a new task. In its basic form, advanced customization involves two key steps:

1. Pre-training: the model is trained on a large dataset that may not fully fit the target problem.
2. Additional tuning: Some layers of the network are frozen while others are actively trained to fine-tune model parameters to meet specific requirements.

The following three works have been considered to realize the additional customization. In (Amiri et al., 2020) additional tuning significantly improved results compared to a model trained on natural images alone. The specificity was to modify the initial layers, which capture more general features, and to freeze deeper layers targeting more specific details. This approach showed that early layers contribute more to the success of medical image processing because of their ability to highlight unique textural features. In (Baur et al., 2017) the use of semi-supervised learning improved the model’s adaptation to the task of segmenting multiple sclerosis lesions in MR images. The use of random inclusion of additional features in combination with additional tuning allowed more efficient use of unlabeled data to improve the generalization ability of the model. The additional adjustment was applied to the last layers of the model, which improved the accuracy of lesion localization. The (Li et al., 2019) examines the use of pseudo-labels for semi-supervised learning. The model was first trained in fully supervised mode and then adapted with additional tuning to a larger set of unlabeled data. Pseudo-labels generated by the model itself were used to further refine the parameters during iterative training. This approach showed a significant improvement in performance over traditional training methods, especially when the amount of labeled data was limited. Based on research, it was decided to further train W-Net to recognize eyelash class.

4.3 Fine-tuning for W-Net

First, the data was carefully prepared for training. Data augmentation including rotations, luminance changes, and reflections was performed, which greatly increased the volume and variety of the training dataset and improved the generalization ability of the model.

To train W-Net to recognize eyelash and pupil class, binary masks for 2100 images were trained. For this purpose, the previously written Daugman’s method was used to find the center and radius of the pupil, and the intensity threshold was automatically adjusted according to the iris intensity. Then the image was cut out at a size $\frac{h}{2} \times w$, which was subjected to a binarization process based on the intensity threshold, which was necessary for eyelash extraction. Next, the pupil mask was superimposed on the obtained binary image with eyelashes. As a result, the original image and its corresponding eyelash and pupil mask were input to the model (Fig. 12).

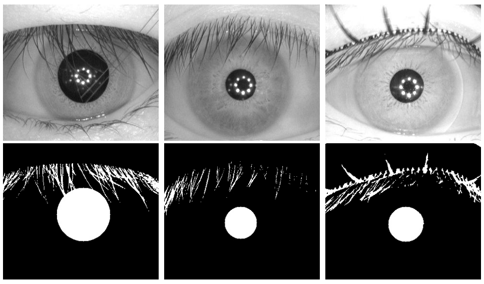


Figure 12. Source images and their corresponding masks.

.

Training occurred only for W_{Enc}, resulting in a segmentation map for K classes. The first class was still the iris class, the second class was the eyelash and pupil class. The loss function was computed for the predicted second class and the prepared mask using the formula:

$$\alpha \cdot J_{\text{soft-N-cut}} + \beta \cdot L_{Loss}, \tag{16}$$

where L_{Loss} = loss calculated according to equation (14)

Training resulted in predictions of the “eyelash and pupil” class and predictions of the iris class (Fig. 13). The remaining 14 classes were unsupervised and the model predicted them automatically.

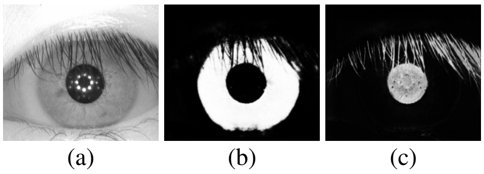


Figure 13. (a) Original image; (b) prediction for iris class; (c) prediction for eyelash and pupil class.

5. Post-processing

Once the predictions were obtained, it was necessary to perform subsequent processing in several steps:

1. The predicted segmentation for the eyelash and pupil class was subjected to morphological processing (erosion and dilatation in several steps) to remove falsely segmented elements from the lower half of the image and noise. The resulting image was then reduced to binary form.

2. Predicted segmentation for the iris class was similarly morphologically processed to remove falsely segmented elements on the iris region.
3. The inverted mask of the eyelash and iris class and superimposed on the binary mask of the iris.

The post-processing process is shown in Fig.14.

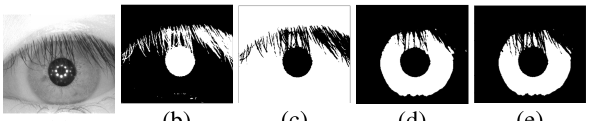

(a) (b) (c) (d) (e)

Figure 14. (a) Original image; (b) binary prediction for iris class; (c) inverted image b); (d) binary prediction for eyelash and pupil class; (e) final result.

Examples of segmentation results can be seen in Fig. 15.

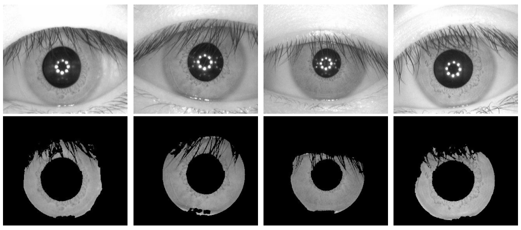


Figure 15. Examples of segmentation results.

6. Iris Normalization

After iris segmentation, the iris image is normalized for subsequent calculation of the iris proximity measure. For this purpose, a pseudo-polar coordinate system centered at the pupil center is introduced (the circles formed by the iris boundaries are non-concentric). In most identification methods, the iris is translated into a rectangular normalized image (Fig. 16).

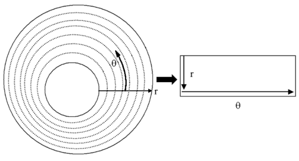


Figure 16. Iris normalization.

The formula for converting an iris point (x, y) to polar coordinates (r, θ) is given as follows:

$$x(\theta, r) = x_p + (R + r_i \cdot (r - R)) \cdot \cos(\theta) \tag{17}$$

$$y(\theta, r) = y_p + (R + r_i \cdot (r - R)) \cdot \sin(\theta)$$

where x_p, y_p = coordinates of the pupil center
r = radius of the pupil
R = radius of the iris outer border
$r_i \in [0, 1]$
$\theta \in [0, 2\pi]$

Thus, the mask obtained after post-processing is overlaid on the original image, the coordinates of the centers and radii of the circles describing the pupil and iris are found, and then the normalization process follows (Fig. 17).

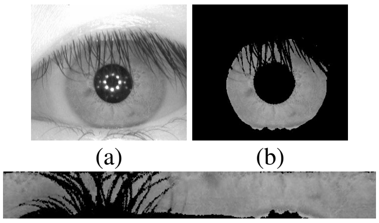

(a) (b)
(c)

Figure 17. (a) Original image; (b) segmentation result; (c) normalized iris image.

7. Segmentation Quality Assessment

For comparison, the articles (Pourafkham, 2024) and (Parisa Farmanifard, 2015) were reviewed. But in open databases (Casia Iris v4 Interval, n.d.) there is only ground truth, which does not take into account eyelashes, so there are no metrics. To evaluate the segmentation quality, 53 images were manually segmented. The results of the comparison are summarized in the Table 1.

Metric	Score
Accuracy	0.928
Precision	0.932
Recall	0.932
F1 Score	0.932
Dice Coefficient	0.851
Matthews Correlation Coefficient (MCC)	0.818

Table 1. Comparison results.

A comparison of the segmentation results with the manually marked segmentation is shown in Fig. 18.

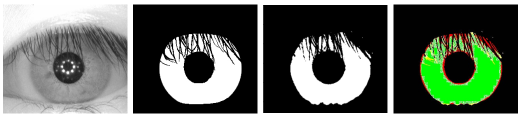

(a) (b) (c) (d)

Figure 18. The results of the proposed iris segmentation method: (a) original image; (b) manually marked image; (c) segmentation result; (d) the comparison of the proposed and manually marked iris segmentation.

8. Conclusion

A neural network method for unsupervised iris segmentation based on W-Net architecture has been developed. The best results were achieved by introducing the Daugman's integro-differential operator into W-Net as a loss function and subsequent fine-tuning for the eyelash class. The method is tested on CASIA-IrisV4-Interval database images.

References

Amiri, M., Brooks, R., Rivaz, H., 2020. Fine-tuning U-Net for ultrasound image segmentation: different layers, different outcomes. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(12), 2510–2518.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12), 2481–2495.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-2-W5-2024-115-2024 | © Author(s) 2024. CC BY 4.0 License.

Baur, C., Albarqouni, S., Navab, N., 2017. Semi-supervised deep learning for fully convolutional networks. *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20*, Springer, 311–319.

Bhattacharyya, D., Ranjan, R., Alisherov, F., Choi, M. et al., 2009. Biometric authentication: A review. *International Journal of u-and e-Service, Science and Technology*, 2(3), 13–28.

CASIA Iris Image Database, n.d. http://biometrics.idealtest.org/. Accessed: 10.10.2023.

Casia Iris v4 Interval, n.d. http://biometrics.idealtest.org/dbDetailForUser.do?id=4/. Accessed: 2023-10-01.

Daugman, J. G., 1993. High confidence visual recognition of persons by a test of statistical independence. *IEEE transactions on pattern analysis and machine intelligence*, 15(11), 1148–1161.

Illingworth, J., Kittler, J., 1988. A survey of the Hough transform. *Computer vision, graphics, and image processing*, 44(1), 87–116.

Li, Z., Ko, B., Choi, H.-J., 2019. Naive semi-supervised deep learning using pseudo-label. *Peer-to-peer networking and applications*, 12, 1358–1368.

Parisa Farmanifard, A. R., 2015. Near-infrared and visible-light periocular recognition with gabor features using frequency-adaptive automatic eye detection. *IET Biometrics*, 4(2), 74-89.

Pourafkham, B., K. H., 2024. ES-Net: Unet-based model for the semantic segmentation of Iris. *Multimed Tools Appl (2024)*.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8), 888–905.

Xia, X., Kulis, B., 2017. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*.