

Article

Deep Learning-Based Iris Segmentation for Iris Recognition in Visible Light Environment

Muhammad Arsalan, Hyung Gil Hong, Rizwan Ali Naqvi, Min Beom Lee, Min Cheol Kim, Dong Seop Kim, Chan Sik Kim and Kang Ryoung Park *

Division of Electronics and Electrical Engineering, Dongguk University, 30 Pildong-ro 1-gil, Jung-gu, Seoul 100-715, Korea; arsal@dongguk.edu (M.A.); hell@dongguk.edu (H.G.H); rizwanali@dongguk.edu (R.A.N.); mblee@dongguk.edu (M.B.L.); mincheol9166@naver.com (M.C.K.); k_ds1028@naver.com (D.S.K.); kimchsi9004@naver.com (C.S.K.)

* Correspondence: parkgr@dongguk.edu; Tel.: +82-10-3111-7022; Fax: +82-2-2277-8735

Received: 8 October 2017; Accepted: 1 November 2017; Published: 4 November 2017

Abstract: Existing iris recognition systems are heavily dependent on specific conditions, such as the distance of image acquisition and the stop-and-stare environment, which require significant user cooperation. In environments where user cooperation is not guaranteed, prevailing segmentation schemes of the iris region are confronted with many problems, such as heavy occlusion of eyelashes, invalid off-axis rotations, motion blurs, and non-regular reflections in the eye area. In addition, iris recognition based on visible light environment has been investigated to avoid the use of additional near-infrared (NIR) light camera and NIR illuminator, which increased the difficulty of segmenting the iris region accurately owing to the environmental noise of visible light. To address these issues; this study proposes a two-stage iris segmentation scheme based on convolutional neural network (CNN); which is capable of accurate iris segmentation in severely noisy environments of iris recognition by visible light camera sensor. In the experiment; the noisy iris challenge evaluation part-II (NICE-II) training database (selected from the UBIRIS.v2 database) and mobile iris challenge evaluation (MICHE) dataset were used. Experimental results showed that our method outperformed the existing segmentation methods.

Keywords: biometrics; iris recognition; iris segmentation; convolutional neural network (CNN)

1. Introduction

Biometrics has two main categories: physiological and behavioral biometrics. Behavioral biometrics considers voice, signature, keystroke, and gait recognition [1], whereas physiological biometrics considers face [2,3], iris [4,5], fingerprints [6], finger vein patterns [7], and palm prints [8]. Iris recognition has been widely used in security and authentication systems because of its reliability and high-security [9,10].

Most existing iris recognition algorithms are designed for highly controlled cooperative environments, which is the cause of their failure in non-cooperative environments, i.e., those that include noise, off-angles, motion blurs, glasses, hairs, specular reflection (SR), eyelids and eyelashes incorporation, and partially open eyes. Furthermore, the iris is always assumed to be a circular object, and common methods segment it as a circle, but considering intense cases of side view and partially open eyes, the iris boundary deviates from being circular and may include skin, eyelid, and eyelash areas. Iris recognition systems are based on the specific texture of the iris area, which is used as a base entity for recognition and authentication purposes. Therefore, the accurate segmentation of the iris boundary is important even in intense environments.

Algorithms for iris segmentation should be designed to reduce user cooperation to improve the overall iris recognition performance. Many studies have attempted to reduce the error caused by the

lack of user cooperation in the past two decades [11–13], but the detection of the true iris boundary is still a challenge. To address this issue, we proposed a two-stage iris segmentation method based on convolutional neural networks (CNN), which is capable of robustly finding the true iris boundary in the above-mentioned intense cases with limited user cooperation. Our proposed iris segmentation scheme can be used with inferior quality noisy images even in visible light environment.

2. Related Works

The existing schemes for iris segmentation can be broadly divided into four types based on implementation. The first and most common type consists of the boundary-based methods. These systems will first find the base element pupil as an inner boundary of the iris, and then find the other parameters, such as eyelid and limbic areas, to separate them from the iris. The first type of methods include Hough transform (HT) and Daugman's integro-differential operator. HT finds the circularity by edge-map voting within the given range of the radius, which is known as the Wildes approach [14]. HT-based detection methods are also used [15,16]. Daugman's integro-differential operator is another scheme that finds the boundary using an integral derivative, and the advanced method was developed in [17]. An effective technique to reduce the error rate in a non-cooperative environment was proposed by Jeong et al. [18]. They used two circular edge detectors in combination with AdaBoost for pupil and iris boundary detection, and their method approximated the real boundary of eyelashes and eyelid detection. Other methods are also known to reduce noise prior to the detection of the iris boundary to increase the segmentation accuracy [19]. All methods in the first type of segmentation require eye images of good quality and an ideal imaging environment; therefore, these types of methods are less accurate for non-ideal situations or result in higher error rates.

The second type of methods is composed of pixel-based segmentation methods. These methods are based on the identification of the iris boundary using specific color texture and illumination information gradient to discriminate between an iris pixel and another pixel in the neighborhood. Based on the discriminating features, the iris classifier is created for iris and non-iris pixel classification. A novel method for iris and pupil segmentation was proposed by Khan et al. [20]. They used 2-D profile lines between the iris and sclera boundary and calculated the gradient pixel by pixel, where the maximum change represents the iris boundary. Parikh et al. [21] first approximated the iris boundary by color-based clustering. Then, for off-angle eye images, two circular boundaries of the iris were detected, and the intersection area of these two boundaries was defined as the iris boundary. However, these methods are affected by eyelashes and hairs or dark skin. The true boundary of the iris is also not identified if it includes the area of eyelids or skin, which reduces the overall iris authentication performance.

The third type of segmentation methods is composed of active contours and circle fitting-based methods [22,23]. A similar approach is used in the local Chan–Vese algorithm, where a mask is created according to the size of the iris, and then an iterative process determines the true iris boundary with the help of the localized region-based formulation [24]. However, this approach shares the drawback faced by other active contour-based models, i.e., it is usually prevented by the iris texture during iteration, and considers the iris pattern as the boundary, which results in erroneous segmentation. On the other hand, active contour-based methods are more reliable in detecting the pupillary boundary because of the significant difference in visibility.

To eliminate the drawbacks of all current segmentation methods and reduce the complexity of intensive pre- and post-processing, a fourth type of segmentation methods evolved, which consists of learning-based methods [25]. Among all learning-based methods, deep learning via deep CNN is the most ideal and popular in current computer vision applications because of its accuracy and performance. This method has been applied to damaged road marks detection [26], human gender recognition from human body images [27], and human detection in night environments using visible light camera [28]. Considering segmentation, CNN can provide a powerful platform to simplify the intensive task with accuracy and reliability similar to brain tumor segmentation done by CNN [29].

Iris-related applications are sensitive because of the very dense and complex iris texture. Therefore, few researchers focus on CNN related to iris segmentation. Liu et al. [30] used DeepIris to solve an intra-class variation of heterogeneous iris images where CNNs are used to learn relational features to measure the similarity between two candidate irises for verification purposes. Gangwar et al. [31] used DeepIrisNet for iris visual representation and cross-sensor iris recognition. However, these two types of research focus on iris recognition instead of iris segmentation.

Considering iris segmentation using CNNs, Liu et al. [32] identified accurate iris boundaries in non-cooperative environments using fully convolutional networks (FCN). In their study, hierarchical CNNs (HCNNs) and multi-scale FCNs (MFCNs) were used to locate the iris boundary automatically. However, due to the use of full input image without the definition of region of interest (ROI) into the CNN, the eyelids, hairs, eyebrows, and glasses frames, which look similar to iris, can be considered as iris points by the CNN model. This scheme has better performance compared to previous methods. However, the error of the iris segmentation can potentially be further reduced.

To address these issues concerning the existing approaches, we propose a two-stage iris segmentation method based on CNN that is robust, to find the true boundary in less-constrained environments. This study is novel in the following three ways compared to previous works.

- The proposed method accurately identified the true boundary even in intense scenarios, such as glasses, off-angle eyes, rotated eyes, side view, and partially opened eyes.
- The first stage includes bottom-hat filtering, noise removal, Canny edge detector, contrast enhancement, and modified HT to segment the approximate the iris boundary. In the second stage, deep CNN with the image input of 21×21 pixels is used to fit the true iris boundary. By applying the second stage segmentation only within the ROI defined by the approximate iris boundary detected in the first stage, we can reduce the processing time and error of iris segmentation.
- To reduce the effect of bright SR in iris segmentation performance, the SR regions within the image input to CNN are normalized by the average RGB value of the iris region. Furthermore, our trained CNN models have been made publicly available through [33] to achieve fair comparisons by other researchers.

Table 1 shows the comparison between existing methods and the proposed method.

Table 1. Comparisons between the proposed and previous methods on iris segmentation.

Type	Methods	Strength	Weakness
Boundary-based methods	Integro-differential operator [17]	Fast processing speed using a simple technique.	These methods are less accurate for iris segmentation in non-ideal situations or visible light.
	Iris localization by circular HT. Upper and lower eyelids detection by parabolic curves [14].	As the first practical scheme, it can produce a good approximation of the iris region with image smoothness.	
	Using two-circular edge detector along with AdaBoost eye detector. In addition, eyelash and eyelid detection is performed [18].	This method provides satisfactory results in non-ideal situation, and a closed eye can be detected as well.	This method fails in iris segmentation in case the RGB values of the eyelids are similar to the iris or pupil/eyelid detection error.
Pixel-based methods	Drawing 2D profile line on the iris-sclera area and calculating the gradient to find the boundary points [20].	A new way to estimate iris boundary from both side gradient.	The calculated gradient is affected by the eyelashes, and the true boundary of the iris is not found.
	Initially approximating iris boundary by color clustering, and then fitting the boundary by two side curves [21].	To reduce the error of iris segmentation, the upper and lower eyelids and the eyelashes are removed by average intensity analysis.	Some empirical threshold is set for eyelid and eyelash detection, but the limitations in detecting the true boundary of the iris still exist.
Active contour and circle fitting-based methods	Iterative method by starting with mask, checking the gradient, and reaching the boundary [24].	For non-ideal cases, it can segment the iris true boundary more accurately than those by boundary- and pixel-based methods.	These methods are better for detecting pupil boundary because they can be prevented by the iris texture considering the iris pattern as the boundary.

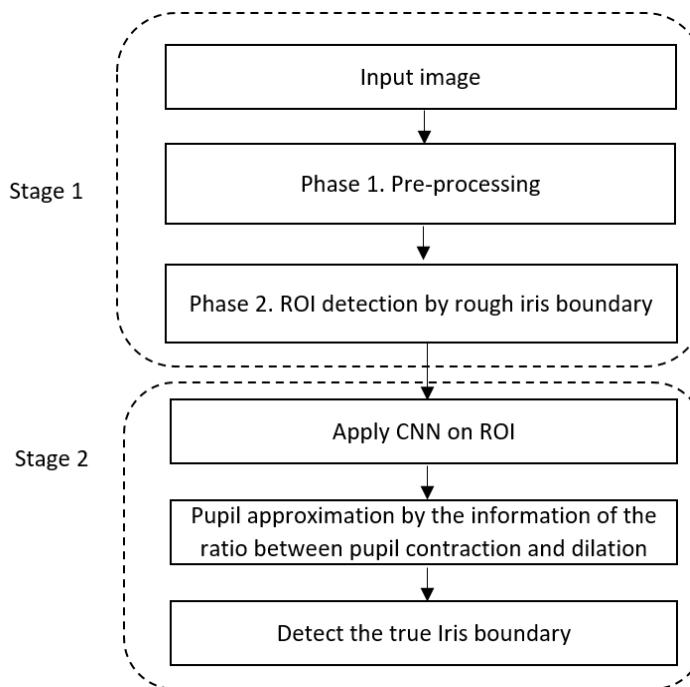
Table 1. *Cont.*

Type	Methods	Strength	Weakness
CNN-based methods	Using HCNNs and MFCNs based deep learning method [32].	This approach has better accuracy relative to existing segmentation methods for non-ideal conditions.	As they use full image into the network, the eyelids, hairs, eyebrows, and glasses frames, which look similar to iris can be considered as iris points by the CNN model.
	Two-stage iris segmentation method (Proposed method)	This approach simply finds the rough boundary of the iris and applies CNN just within the ROI defined based on the rough boundary.	A large amount of data is needed for CNN training.

3. Proposed Method

3.1. Overview of the Proposed System

Figure 1 shows an overall flowchart of the proposed two-stage iris segmentation method. In the first stage, the rough iris boundary is obtained from the input image to define ROI for next stage. The resultant image from Stage 1 includes parts of the upper and lower eyelids and other areas, such as skin, eyelashes, and sclera. Consequently, the true iris boundary needs to be identified. In the second stage, within this ROI (defined by Stage 1), CNN is applied, which can provide the real iris boundary with the help of learned features. Considering the standard information of the ratio between pupil contraction and dilation, pupil approximation is performed, and finally, the actual iris area can be obtained.

**Figure 1.** Flowchart of the proposed method.

3.2. Stage 1. Detection of Rough Iris Boundary by Modified Circular HT

An approximate localization of the iris boundary is the prerequisite of this study, and it is obtained by modified circular HT-based method. As shown in Figure 1, Stage 1 consists of two phases, namely, pre-processing and circular HT-based detection.

3.2.1. Phase 1. Pre-Processing

In Phase 1, as shown in Figures 2 and 3c, the RGB input image is converted into grayscale for further processing, and morphological operation is applied through bottom-hat filtering with symmetrical structuring element disk of size 5 for contrast enhancement. Finally, two images of the gray image and resultant image by bottom-hat filtering are added to obtain an enhanced image as shown in Figure 3d.

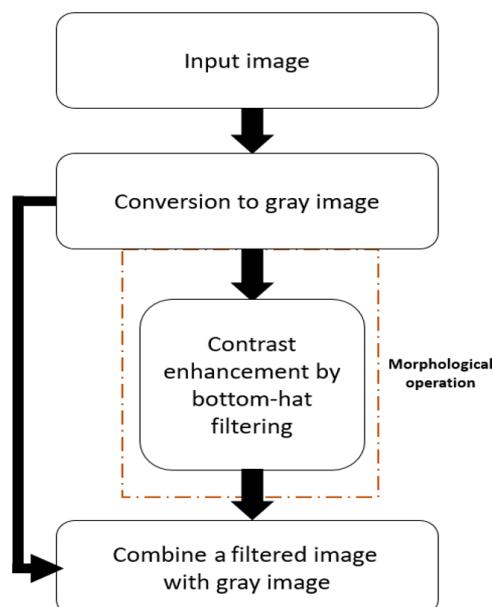


Figure 2. Overall pre-processing procedure of Phase 1.

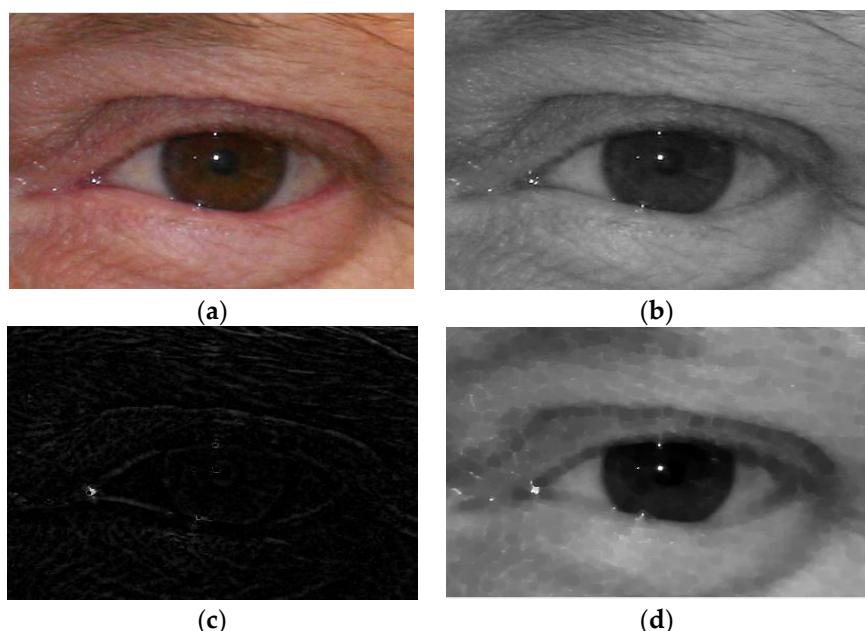


Figure 3. Resultant images by Phase 1 of Figure 1. (a) Original input image of 400×300 pixels; (b) grayscale converted image; (c) resultant image by bottom-hat filtering; (d) resultant image by adding the two images of (b,c).

3.2.2. Phase 2. ROI Detection by Rough Iris Boundary

The overall process of Phase 2 is presented by the flowchart in Figure 4. In Phase 2, the filtered image from Phase 1 is redefined as an image of 280×220 pixels to reduce the effect of eyebrows in detecting the iris boundary. Then, a 17×17 median filter is applied to smooth the image from salt and pepper noises and reduce the skin tone and texture illumination as shown in Figure 5c. 13×13 symmetrical Gaussian smoothing filter with σ of 2 is applied to the filtered image to increase pixel uniformity as shown in Figure 5d. Then, Canny edge detector with same σ value is used to detect the edges of the iris boundary as shown in Figure 5e. However, the edges are not very clear, and gamma adjustment with $\gamma = 1.90$ is applied to enhance the contrast of the image as shown in Figure 5f. With the gamma-enhanced image, the binarized edge image is obtained with eight neighbor connectivity as shown in Figure 5g. In this edge image, there are more circular edges along the iris boundary edges, and circular HT can find all possible circles in the image. However, the incorrect circle-type edges as shown in Figure 5h can be removed by filtering the edges whose radius is out of the range of the minimum and maximum human iris radius. Then, the most-connected edges are selected as iris edges, and the rough iris boundary is detected in Figure 5i. Considering the possibility of detection error of the iris boundary, ROI is defined slightly larger than the detected boundary as shown in Figure 7a.

For fair comparisons, all the optimal parameters for the operation in ROI detection including median filter, Gaussian smoothing filter, Canny edge detector, gamma adjustment, and binarization, etc., were empirically selected only by training data without testing data.

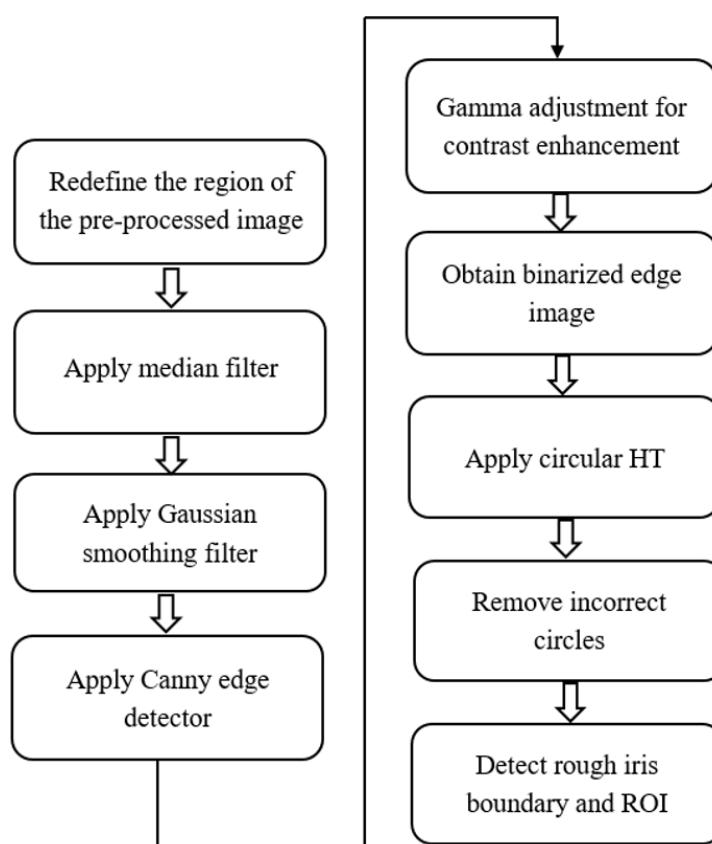


Figure 4. Overall region of interest (ROI) detection procedure of Phase 2.

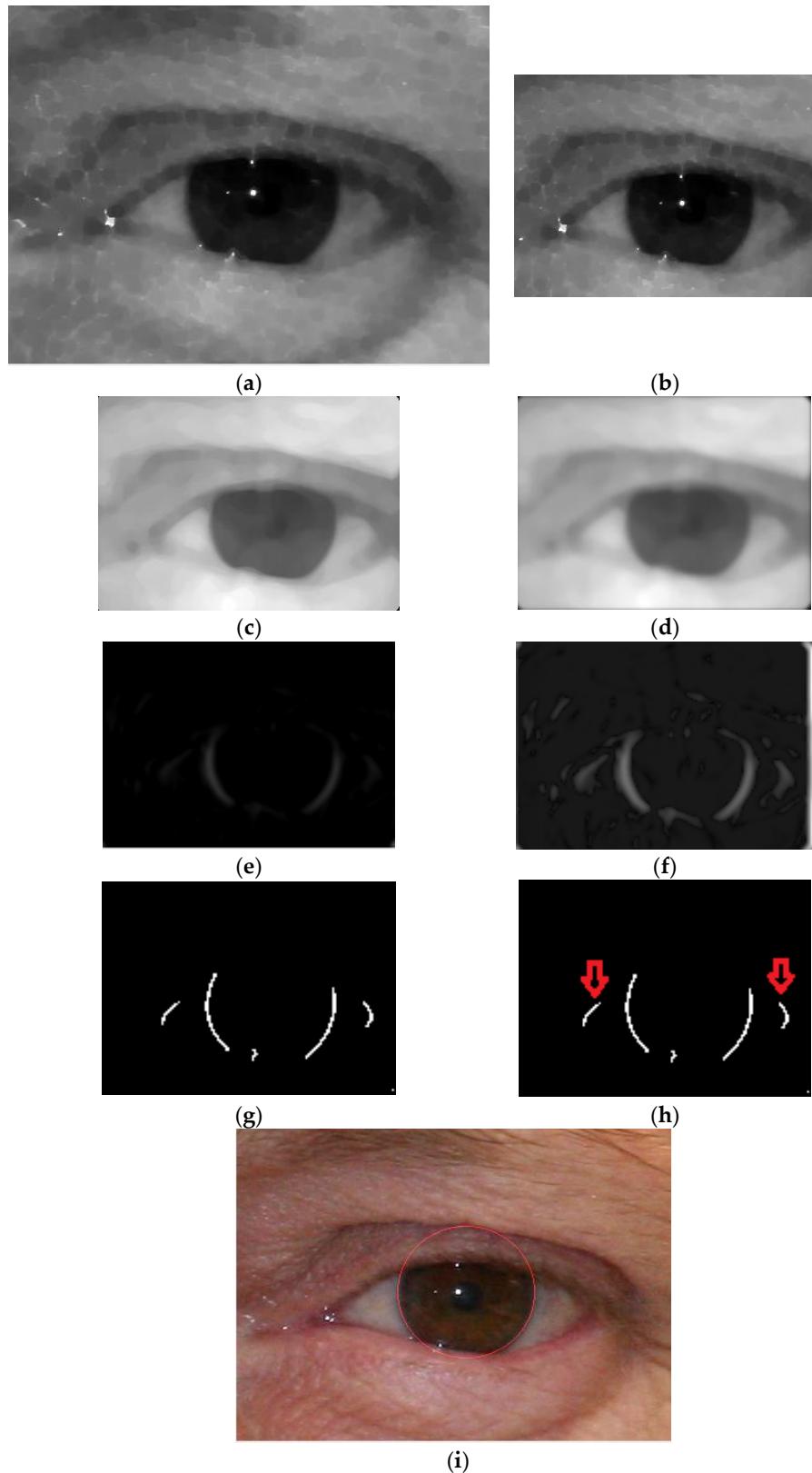


Figure 5. Resultant images of rough iris boundary detected by Phase 2 of Figure 1; (a) pre-processed image of 400×300 pixels from Phase 1; (b) redefined image of 280×220 pixels; (c) median filtered image; (d) image by Gaussian smoothing; (e) image after Canny edge detector; (f) image after gamma contrast adjustment; (g) binarized edge image; (h) resultant image by removing incorrect circles by radius (i) final image with rough iris boundary.

3.3. Iris Rough Boundary Analysis

The rough iris boundary detected in Stage 1 is not the real iris boundary, but an approximation for the next stage. Considering ideal cases in which user cooperation is available, the output of the HT can show the accurate iris boundary. However, for non-ideal cases, such as off-angle eyes, rotated eyes, eyelash occlusions, and semi-open eyes, the HT can sometimes produce inaccurate iris boundaries as shown in Figure 6. Moreover, the detected iris circle includes the eyelid, eyelash, pupil, and SR, which should be discriminated from the true iris area for iris recognition. Therefore, we proposed a CNN-based segmentation method of the iris region based on the ROI defined by the rough iris boundary in Stage 1.



Figure 6. Examples of rough iris boundaries detected by Stage 1 for non-ideal environments.

3.4. CNN-Based Detection of the Iris Region

3.4.1. Extracting the Mask for CNN Input

To detect the iris region accurately, the square mask of 21×21 pixels is extracted from the ROI of Figure 7a and is used as input to CNN. The mask is extracted within the ROI to reduce the number of objects to be classified. Specifically, in many cases, iris color can be similar to the eyebrows and eyelids. Furthermore, in non-ideal cases, the skin can have similar color to iris. Therefore, by extracting the mask only within the ROI, we can reduce the iris segmentation error by CNN. This mask is scanned in both horizontal and vertical directions as shown in Figure 7b. Based on the output of CNN, the center position of the mask is determined as an iris or non-iris pixel. Figure 7c shows the examples of the collected masks of 21×21 pixels for CNN training or testing. As shown in Figure 7c, the mask from the iris region has the characteristics where most pixels of the mask are from the iris texture, whereas that from the non-iris region has the characteristics where most pixels are from the skin, eyelid, eyelash, or sclera.

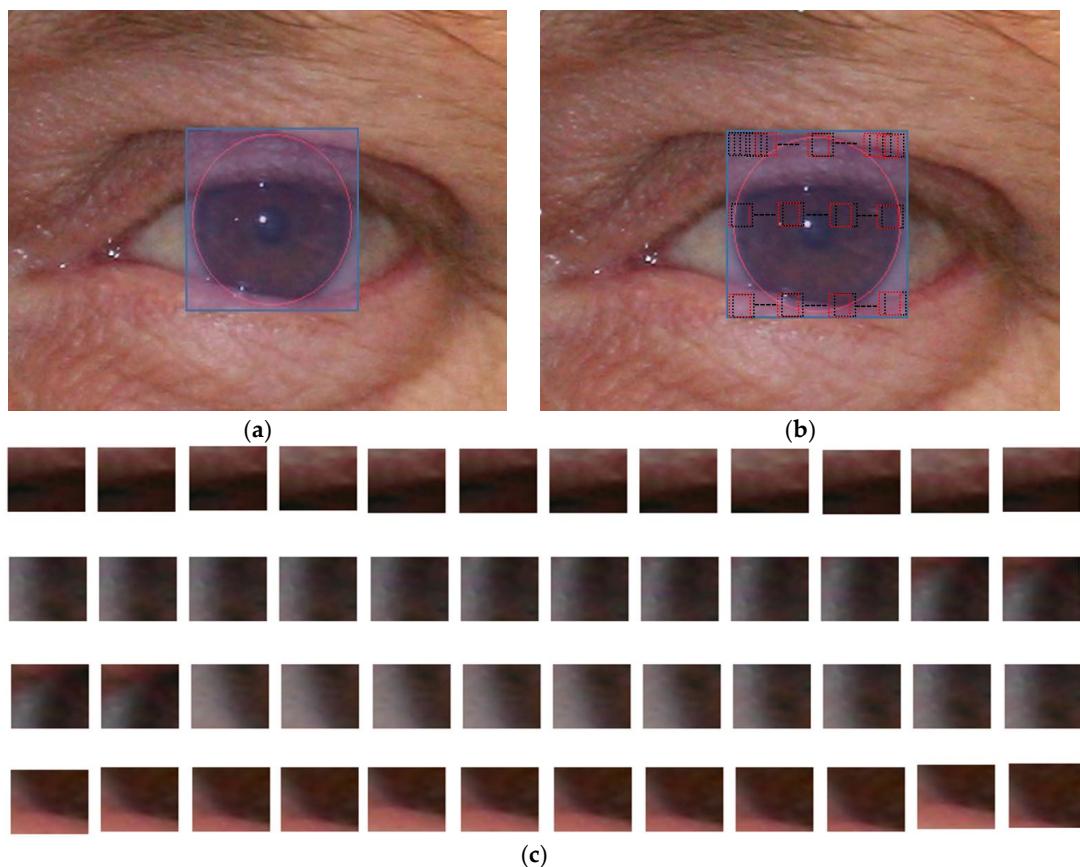


Figure 7. Extracting the mask of 21×21 pixels for training and testing of CNN. (a) ROI defined from Stage 1 of Figure 1; (b) extracting the mask of 21×21 pixels within the ROI; (c) examples of the extracted masks (the 1st (4th) row image shows the masks from the boundary between the upper (lower) eyelid and iris, whereas the others represent those from the left boundary between iris and sclera).

However, if the mask is extracted from the bright SR region, the characteristics of the pixels of the mask can be changed as shown in Figure 8a, which can increase the error of iris segmentation by CNN. To solve this problem, if the bright pixels whose gray level is higher than 245 exist inside the mask, they are replaced by the average RGB value of the iris area (the red dotted box in Figure 8a) as shown in Figure 8b. The threshold of 245 was experimentally found with training data. Here, we can reduce the effect of SR on the iris segmentation by CNN. The position of the red dotted box is determined

based on the rough iris boundary detected in Stage 1. If the dark pixel exists inside the box because the dark pupil can be included in the box, the box position is adaptively moved into the lower direction until it does not include the dark pixel, and the average RGB value is extracted in this box.

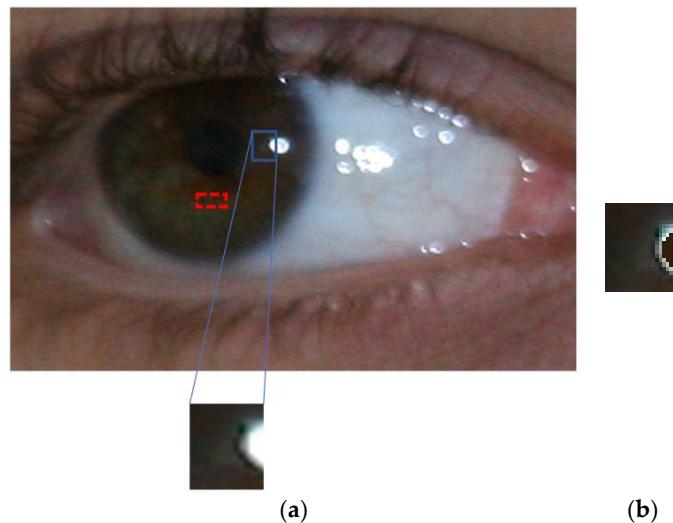


Figure 8. Example of replacing specular reflection (SR) pixels by the average RGB value of the iris region. (a) The mask including the SR region; (b) the mask where SR pixels are replaced.

3.4.2. Iris Segmentation by CNN

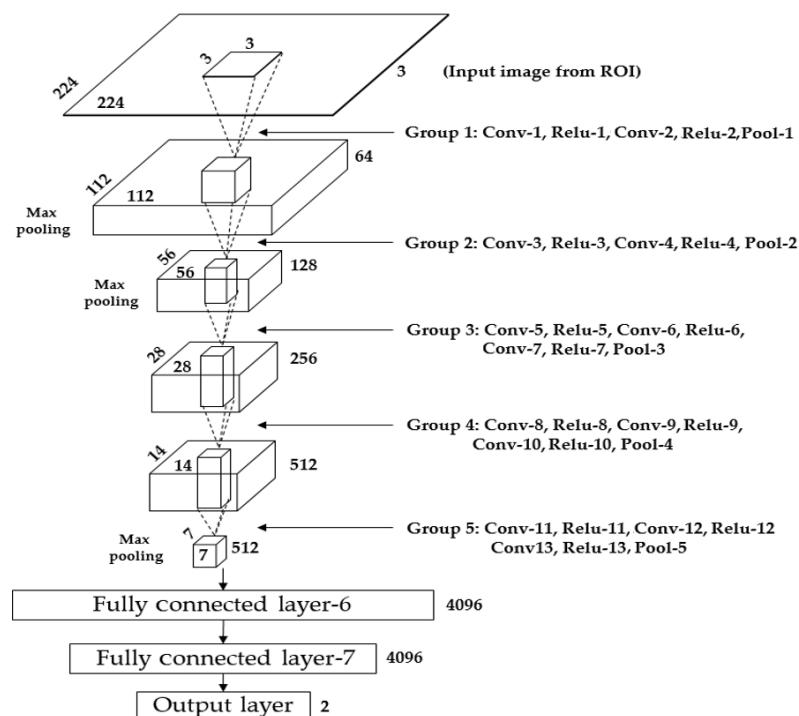
The mask of 21×21 pixels is used as input to CNN, and the output of CNN is either iris or non-iris area. In this study, a pre-trained VGG-face model [34] is used by fine-tuning with the help of our training images. The VGG-face model was pre-trained with about 2.6 million face images of 2,622 different people. Detailed explanations of the configuration of VGG-face are shown in Table 2 and Figure 9. To obtain an accurate boundary and its difference from other objects, the ROI is selected with slightly increased rough iris boundary detected by HT.

Table 2. Configuration of the VGG-face model used in the proposed method.

Type of Layer	No. of Filter	Feature Map Size	Kernel Size	No. of Stride	No. of Padding
Image input layer		$224 \times 224 \times 3$ (height \times width \times channel)			
Group 1	Conv-1 (1st convolutional layer)	64	$224 \times 224 \times 64$	3×3	1×1
	Relu-1		$224 \times 224 \times 64$		
	Conv-2 (2nd convolutional layer)	64	$224 \times 224 \times 64$	3×3	1×1
	Relu-2		$224 \times 224 \times 64$		
Group 2	Pool-1	1	$112 \times 112 \times 64$	2×2	2×2
	Conv-3 (3rd convolutional layer)	128	$112 \times 112 \times 128$	3×3	1×1
	Relu-3		$112 \times 112 \times 128$		
	Conv-4 (4th convolutional layer)	128	$112 \times 112 \times 128$	3×3	1×1
Group 3	Relu-4		$112 \times 112 \times 128$		
	Pool-2	1	$56 \times 56 \times 128$	2×2	2×2
	Conv-5 (5th convolutional layer)	256	$56 \times 56 \times 256$	3×3	1×1
	Relu-5		$56 \times 56 \times 256$		
	Conv-6 (6th convolutional layer)	256	$56 \times 56 \times 256$	3×3	1×1
	Relu-6		$56 \times 56 \times 256$		

Table 2. Cont.

Type of Layer	No. of Filter	Feature Map Size	Kernel Size	No. of Stride	No. of Padding	
Conv-7 (7th convolutional layer)	256	56 × 56 × 256	3 × 3	1 × 1	1 × 1	
Relu-7		56 × 56 × 256				
Pool-3	1	28 × 28 × 256	2 × 2	2 × 2	0 × 0	
Conv-8 (8th convolutional layer)	512	28 × 28 × 512	3 × 3	1 × 1	1 × 1	
Relu-8		28 × 28 × 512				
Group 4	Conv-9 (9th convolutional layer)	512	28 × 28 × 512	3 × 3	1 × 1	1 × 1
	Relu-9		28 × 28 × 512			
	Conv-10 (10th convolutional layer)	512	28 × 28 × 512	3 × 3	1 × 1	1 × 1
	Relu-10		28 × 28 × 512			
	Pool-4	1	14 × 14 × 512	2 × 2	2 × 2	0 × 0
	Conv-11 (11th convolutional layer)	512	14 × 14 × 512	3 × 3	1 × 1	1 × 1
	Relu-11		14 × 14 × 512			
Group 5	Conv-12 (12th convolutional layer)	512	14 × 14 × 512	3 × 3	1 × 1	1 × 1
	Relu-12		14 × 14 × 512			
	Conv-13 (13th convolutional layer)	512	14 × 14 × 512	3 × 3	1 × 1	1 × 1
	Relu-13		14 × 14 × 512			
	Pool-5	1	7 × 7 × 512	2 × 2	2 × 2	0 × 0
Fully connected layer-6 (1st FCL)		4096 × 1				
Relu-6		4096 × 1				
Dropout-6		4096 × 1				
Fully connected layer-7 (2nd FCL)		4096 × 1				
Relu-7		4096 × 1				
Dropout-7		4096 × 1				
Fully connected layer-8 (3rd FCL)		2 × 1				
Softmax layer		2 × 1				
Output layer		2 × 1				

**Figure 9.** CNN architecture used in the proposed method.

The VGG-face model consists of 13 convolutional layers and 5 pooling layers in combination with 3 fully connected layers. The filter size, rectified linear unit (Relu), padding, pooling, and stride are explained in Table 2 and Figure 9. A total of 64 3×3 size filters are adopted in the 1st convolutional layer. Therefore, the size of the feature map is $224 \times 224 \times 64$ in the 1st convolutional layer. Here, 224 and 224 denote the height and width of the feature map, respectively. They are calculated based on $(\text{output height (or width)} = (\text{input height (or width)} - \text{filter height (or width)} + 2 \times \text{the number of padding}) / \text{the number of stride} + 1)$ [35]. For example, in the image input layer and Conv-1 in Table 2, the input height, filter height, number of padding, and number of strides are 224, 3, 1, and 1, respectively. As a result, the height of the output feature map becomes $224 (= (224 - 3 + 2 \times 1) / 1 + 1)$.

There are three common activation functions, such as sigmoid, tanh function, and Relu [35]. The sigmoid-based activation function forces the candidate input value between the range of 0 and 1 as shown in Equation (1), which means that for the negative inputs, the output becomes zero, whereas it becomes 1 for large positive inputs.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

The tanh function is slightly different from the sigmoid activation function because it keeps fitting the input value in the range of -1 and 1 as shown in Equation (2).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

Relu performs faster compared to these two nonlinear activation functions, and it is useful to wipe off the gradient problem in the back propagation at the time of training [36,37]. In [38], they also showed that the speed of training by Relu with the CIFAR-10 dataset based on the four-layered CNN is six times faster than the tanh function with same dataset and network. Therefore, we use Relu for faster training with simplicity and to avoid the gradient issue in CNN. The Relu was initially used in Boltzmann machines, and it is formulated as follows.

$$y_i = \begin{cases} x_i, & x_i \geq 0 \\ 0, & x_i < 0 \end{cases}, \quad (3)$$

where y_i and x_i are the corresponding outputs and inputs of the unit, respectively. The Relu layer exists after each convolutional layer, and it maintains the size of each feature map. Max-pooling layers can provide a kind of subsampling. Considering pool-1, which performs max pooling after the convolutional layer-2 and Relu-2, the feature map of $224 \times 224 \times 64$ is reduced to that of $112 \times 112 \times 64$. In the case that the max-pooling layer (pool-1) is executed, the input feature map size is $224 \times 224 \times 64$, the filter size is 2×2 , and the number of strides is 2×2 . Here, 2×2 for the number of strides denotes the max-pooling filter of 2×2 where the filter moves by two pixels in both the horizontal and vertical directions. Owing to the lack of overlapped area due to filter movement, the feature map size is reduced to $1/4$ ($1/2$ horizontally and $1/2$ vertically). Consequently, the feature map size after passing pool-1 becomes $112 \times 112 \times 64$ pixels. This pooling layer is used after Relu-2, Relu-4, Relu-7, Relu-10, and Relu-13 as shown in Table 2. For all cases, the filter of 2×2 and stride of 2×2 are used, and the feature map size diminishes to $1/4$ ($1/2$ horizontally and $1/2$ vertically).

When an input image of $224 \times 224 \times 3$ is given to the CNN shown in Figure 9, it passes through 13 convolutional layers (Conv-1 to Conv-13), 13 Relu functions, and 5 pooling layers (pool-1 to pool-5). Therefore, after Group 5 (Conv-11, Relu-11, Conv-12, Relu-12, Conv-13, Relu-13, and pool-5), the feature-map of $7 \times 7 \times 512$ is obtained. Then, this feature-map passes through three fully connected layers (fully connected layer-6 to fully connected layer-8), from which two outputs can be obtained. These two final outputs correspond to the two classes of iris and non-iris regions. The number of

output nodes of the 1st, 2nd, and 3rd FCLs are 4096, 4096, and 2, respectively. After the fully connected layer-8, the softmax function is used as follows [39].

$$\sigma(q)_j = \frac{e^{qj}}{\sum_{n=1}^S e^{qn}} \quad (4)$$

In Equation (4), given that the array of the output neurons is q , the probability of neurons of the j th class can be obtained by dividing the value of the j th component by the summation of the values of all the components.

CNN usually has an over-fitting issue, which can reduce the testing accuracy. Therefore, to solve this issue, the dropout method and data augmentation have been considered. Dropout methods are important during training to avoid the same type of neurons representing the same feature repeatedly. Which can cause overfitting, wastage of network capacity, and computational resources. Therefore, the solution is to drop out these cases at random by using the dropout ratio to obtain specific benefits, such as the dropped-out neurons' contribution in forward or back propagation. Consequently, it can reduce co-adoption during the process by decreasing dependencies over other neurons and it can force the network to learn strong features with different neurons [38,40]. In this study, we use the dropout ratio of 0.5. The dropout layer was used twice, that is, after the 1st FCL with Relu-6 and after the 2nd FCL and Relu-7, as shown in Table 2.

Compared to the elements and parameters of the original VGG-face model, three parts of initial learning rate, the momentum value, and the size of the mini-batch were modified, and their optimal values were experimentally found with training data. The detailed explanations of these values are included in Section 4.2. In addition, the last part modified is the number of CNN output as 2 as shown in Figure 9 because the number of classes in our research is 2 as iris and non-iris pixels.

3.5. Pupil Approximation by the Information of the Ratio of Pupil Contraction and Dilation

When applying CNN, false positive errors (non-iris pixel is incorrectly classified into iris pixel) exist in the pupil area. However, the noisy iris challenge evaluation part -II (NICE-II) database used in our experiment includes inferior quality images, where it is very difficult to segment pupillary boundary accurately. Therefore, we used anthropometric information provided by Wyatt [41].

$$d_p = \frac{P_d}{I_d} \quad (5)$$

where P_d represents the pupil diameter and I_d shows the iris diameter. This anthropometric information [41] provides the details of pupil dilation ratio (d_p) in Equation (5) that varies from 12% to 60%, and we use the minimum value in our experiment. Specifically, the iris pixel that is extracted by CNN belongs to the region whose d_p is less than 12% is determined as non-iris pixel (pupil).

4. Experimental Results

4.1. Experimental Data and Environment

In this study, we used the NICE-II training database, which was used for NICE-II competition [42]. The database includes extremely noisy iris data of the UBIRIS.v2. This database contains severely noisy 1000 image of 171 classes. The size of the image is 400×300 pixels. The images of the iris were acquired from people walking 4–8 m away from a high-resolution visible light camera with visible light illumination [43]. Therefore, this database includes the difficulties, such as poorly focused, off-angle, rotated, motion blur, eyelash obstruction, eyelids occlusions, glasses obstructions, irregular SR, non-uniform lighting reflections, and partial captured iris images as shown in Figure 10.



Figure 10. Examples of noisy iris images of NICE-II database.

In this study, among the total 1000 iris images, 500 iris images are used for training, and the other 500 iris images are used for testing purposes. The average value of two accuracies was measured by two-fold cross validation. CNN training and testing are performed on a system using Intel® Core™ i7-3770K CPU @ 3.50 GHz (4 cores) with 28 GB of RAM, and NVIDIA GeForce GTX 1070 (1920 Cuda cores) with graphics memory of 8 GB (NVIDIA, Santa Clara, CA, USA) [44]. The training and testing are done with Windows Caffe (version 1) [45].

4.2. CNN Training

In our proposed method, we fine-tuned the VGG-face model using 500 iris images for classification of two classes (iris and non-iris). For two-fold cross validation, we used the first 500 images and the other 500 images for testing. From the first 500 iris images for training, we have 9.6 million images of 21×21 pixels from the iris ROI for training, and in the second training for cross validation, we have 8.9 million images of 21×21 pixels from the iris ROI. During the CNN training, stochastic gradient descent (SGD) is used to minimize the difference between the calculated and desired outputs with the help of the gradient derivative [46]. The number of the whole training set divided by a mini-batch size is defined as iteration. The total time taken for the complete training including all the iterations is set as 1 epoch, and the training was executed several times as per a pre-determined epoch. In this study, CNN was trained by 10 epochs. For the fine-tuning of VGG-face, the optimum fine-tuning model was experimentally found based on the optimal parameters of initial learning rate of 0.00005, the momentum value of 0.9, and the size of the mini-batch of 20. The detailed explanations of these parameters can be found in the following literature [47]. Figure 11a,b show the curves of average loss and accuracy with training data in case of two-fold cross validation. Epoch count is represented on the X-axis, whereas the right Y-axis represents the training accuracy and the left Y-axis shows the training loss. Depending on the learning rate and mini-batch size, the loss varies. While training, it is important to reach the minimum training loss (maximum training accuracy); therefore, learning rate should be decided carefully. The loss value decreased dramatically with a higher value of the learning rate, which can deteriorate the loss value with reaching to a minimum. In our proposed method, the finest model in which the training loss curve converges to 0% (training accuracy of 100%) is used for testing as shown in Figure 11. To have fair comparisons by other researchers, we have made our trained CNN models publicly available through [33]. Figure 12 shows the examples of trained filters of 1st convolutional layer of Table 2 in case of two-fold cross validation.

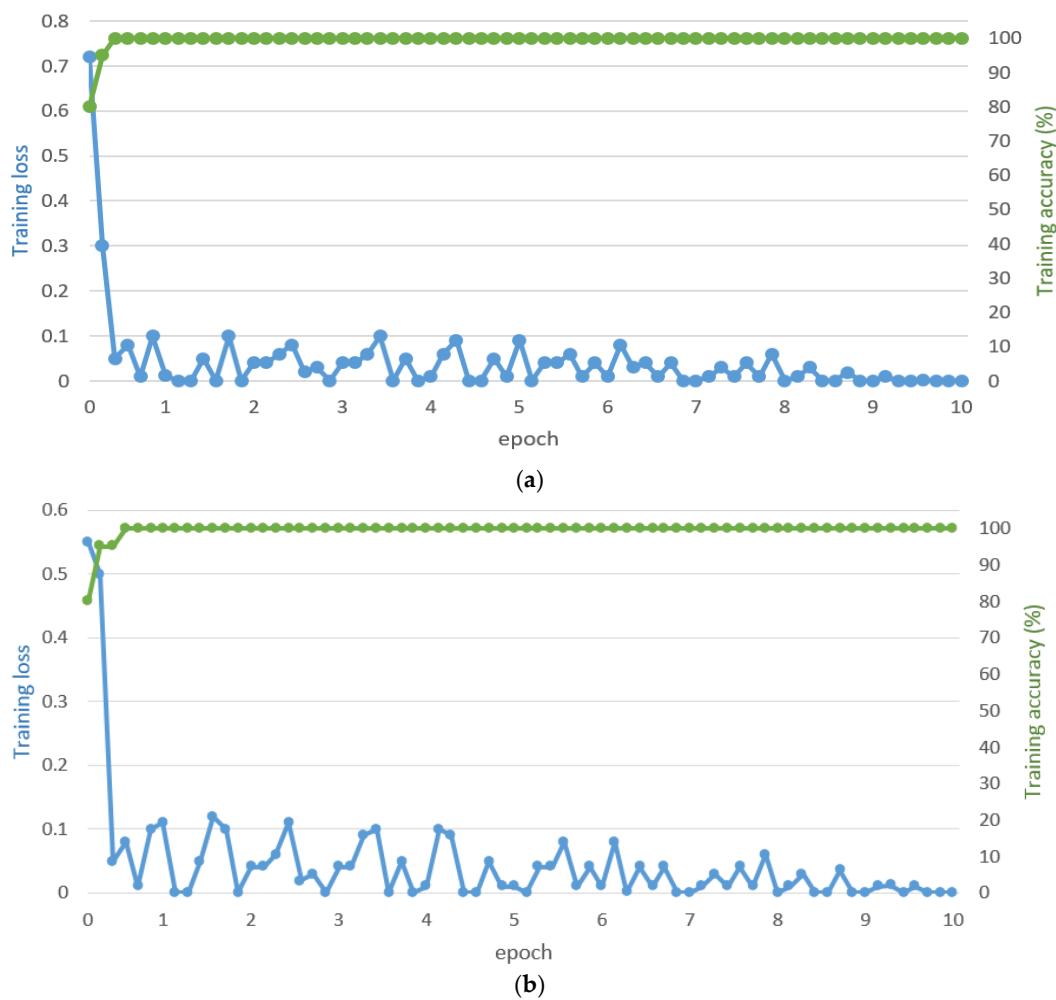


Figure 11. Average loss and accuracy curves for training. Average loss and accuracy curve from (a) 1st fold cross validation; and (b) 2nd fold cross validation.

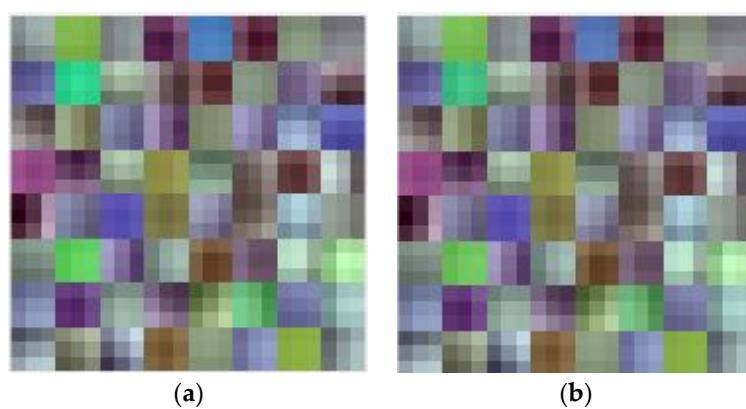


Figure 12. Examples of trained filters of the 1st convolutional layer of Table 2. (a) 1st fold cross validation; (b) 2nd fold cross validation.

4.3. Testing of the Proposed CNN-Based Iris Segmentation

The performance of the proposed method for iris segmentation is evaluated based on the metrics of the NICE-I competition to compare the accuracy with that of the teams participating in NICE-I competition [48]. As shown in Equation (6), the classification error (E_i) is measured by comparing the

resultant image ($I_i(m', n')$) by our proposed method and the ground truth image ($G(m', n')$) with help of the exclusive-OR (XOR) operation.

$$E_i = \frac{1}{m \times n} \sum_{(m', n')} I_i(m', n') \otimes G_i(m', n') \quad (6)$$

where m and n are the width and height of the image, respectively. To evaluate the proposed method, the average segmentation error ($E1$) is calculated by averaging the classification error rate (E_i) of the whole images as shown in Equation (7).

$$E1 = \frac{1}{k} \sum_i E_i \quad (7)$$

where k represents the total number of testing images. $E1$ varies between [0,1], which denotes that “0” represents the least error and “1” represents the largest error.

4.3.1. Iris Segmentation Results by the Proposed Method

Figure 13 shows the examples of good segmentation results by our proposed method. In our experiment, we can consider two types of error, such as false positive and false negative errors. The former denotes that the non-iris pixel is incorrectly classified into an iris, whereas the latter denotes that the iris pixel is incorrectly classified into non-iris. In Figure 13, the false positive and false negative errors are shown in green and red, respectively. The true positive case (iris pixel is correctly classified into iris one) is shown in black. As shown in Figure 13, our proposed method can correctly segment the iris region irrespective of various noises in eye image. Figure 14 shows examples of incorrect segmentation of the iris region by our proposed method. False positive errors occur in the eyelash area whose pixel values are similar to that of the iris region whereas false negative errors happen in case of the reflection noises from glasses surface or severely dark iris area.

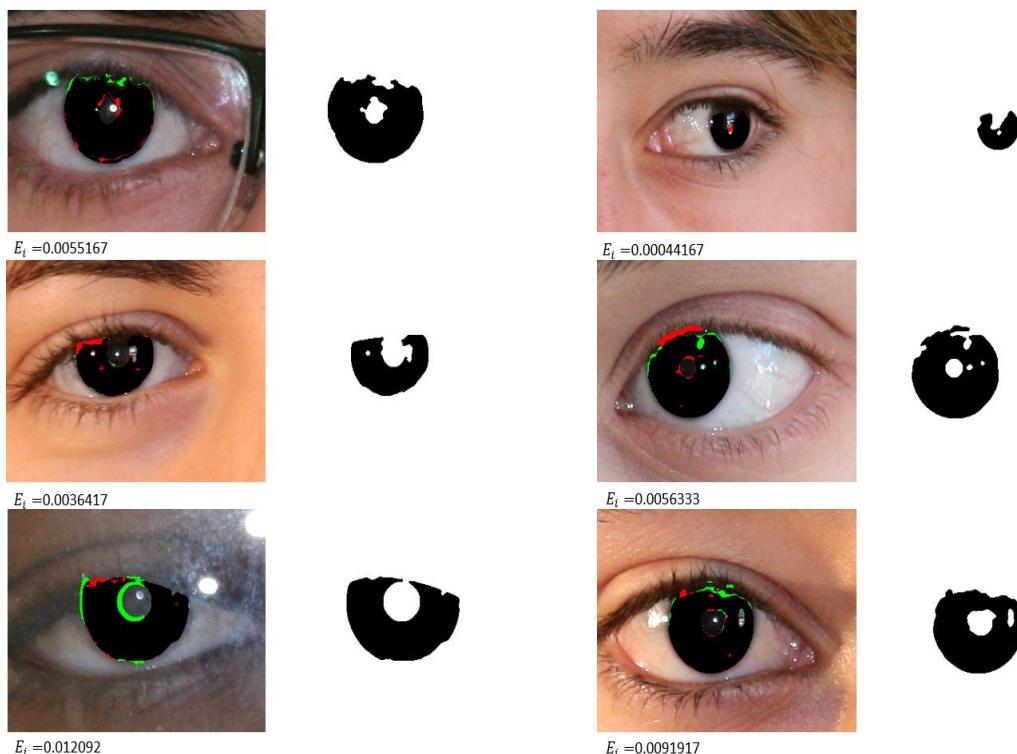


Figure 13. Cont.

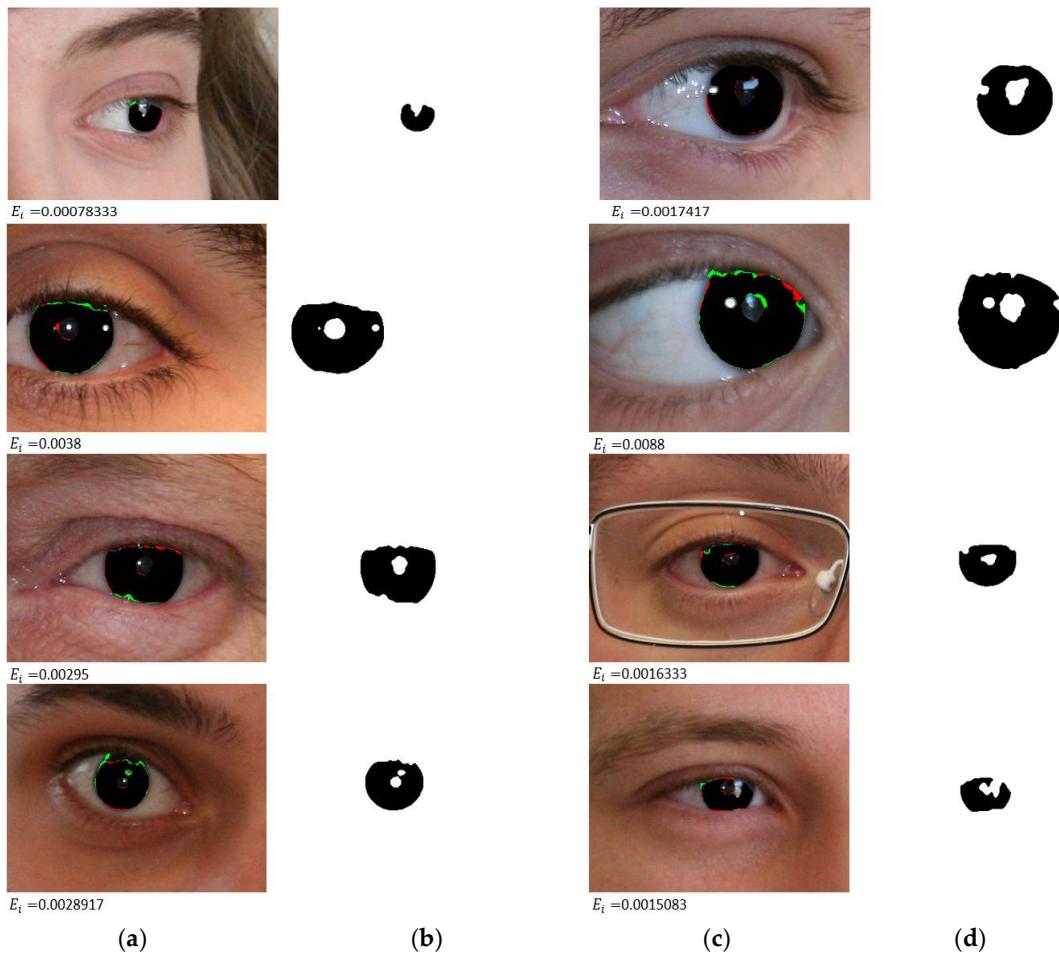


Figure 13. Examples of good segmentation results by our proposed method. (a,c) segmentation results with corresponding E_t ; (b,d) ground truth images (the false positive and false negative errors are shown in green and red, respectively. The true positive case (iris pixel is correctly classified into iris one) is shown in black).

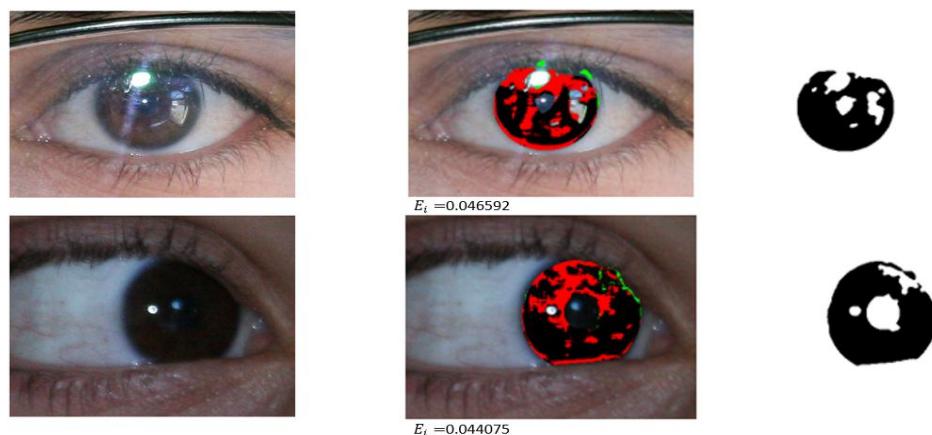


Figure 14. Cont.

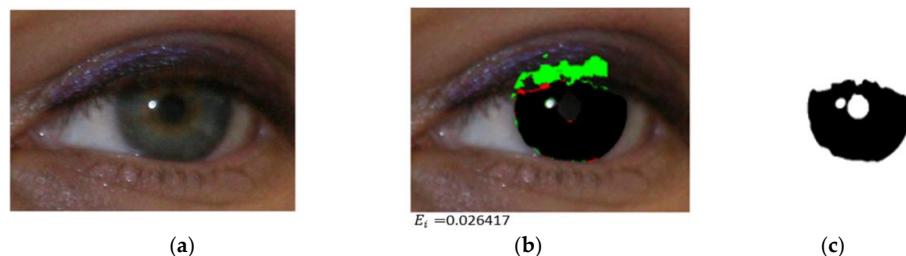


Figure 14. Examples of incorrect segmentation of the iris region by our proposed method. (a) Original input images; (b) segmentation results with corresponding E_1 ; (c) ground truth images (the false positive and false negative errors are shown in green and red, respectively. The true positive case (iris pixel is correctly classified into iris one) is shown in black).

4.3.2. Comparison of the Proposed Method with Previous Methods

In Table 3, we show the comparisons of the proposed method with previous methods based on E_1 of Equation (7). As shown in these results, our method outperforms the previous methods in iris segmentation error.

In addition, we compared the results by support vector machine (SVM) with those by our CNN. For fair comparison, we used the same training and testing data (obtained from the stage 1 of Figure 1) of two-fold cross validation for both SVM and CNN. From the two-fold cross validation, we obtained the average value of E_1 of Equation (7). As shown Table 3, E_1 by proposed method (using CNN for the stage 2 of Figure 1) is lower than that by SVM-based method (using SVM for the stage 2 of Figure 1), from which we can confirm that our proposed CNN-based method is better than traditional machine learning method of SVM. The reason why the error by CNN-based method is lower than that by SVM-based method is that optimal features by optimal filters can be extracted by the 13 convolutional layers (Table 2) of our CNN compared to SVM.

The goal of our research is to detect the accurate positions of iris pixels based on pixel-level labels. For this purpose, we can consider fast R-CNN [49] or faster R-CNN [50], but only the area of square box shape (based on box-level labels) can be detected by these method [49,50]. Therefore, in order to detect the accurate positions of all the iris pixels (not square box-level, but pixel-level), we should consider the different type of CNN such as semantic segmentation network (SSN) [51,52]. By using this SSN, the accurate positions of all the iris pixels can be obtained. In details, for SSN, whole image (instead of the mask of 21×21 pixels extracted from the stage 1 of Figure 1) was used as input without the stage 1 of Figure 1. For fair comparison, we used the same training and testing images (from which the mask of 21×21 pixels were extracted for our method) for the SSN. From the two-fold cross validation, we obtained the average value of E_1 of Equation (7). As shown Table 3, E_1 by proposed method (using CNN for the stage 2 of Figure 1) is lower than that by SSN-based method (using SSN for the stages 1 and 2 of Figure 1), from which we can confirm that our proposed method is better than SSN-based method. The reason why the error by SSN-based method is higher than that by our method is that the classification complexity of iris and non-iris pixels is increased by applying the SSN to whole image instead of the ROI detected by the stage 1 of Figure 1.

Table 3. Comparison of the proposed method with previous methods using NICE-II dataset.

Method	E_1
Luengo-Oroz et al. [53] (8th rank in NICE-I competition)	0.0305
Labati et al. [54] (7th rank in NICE-I competition)	0.0301
Chen et al. [55] (6th rank in NICE-I competition)	0.029
Jeong et al. [18] (5th rank in NICE-I competition)	0.028

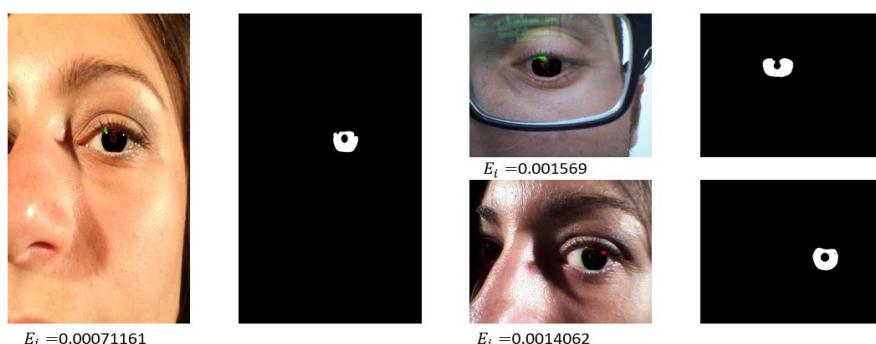
Table 3. *Cont.*

Method	E_1
Peihua et al. [56] (4th rank in NICE-I competition)	0.022
Tan et al. [57]	0.019
Proen��a [58]	0.0187
de Almeida [59] (3rd rank in NICE-I competition)	0.0180
Tan et al. [60]	0.0172
Sankowski et al. [61] (2nd rank in NICE-I competition)	0.016
Tan et al. [19] (1st rank in NICE-I competition)	0.0131
Haindl et al [62]	0.0124
Zhao et al. [16]	0.0121
SSN-based method [51,52]	0.02816
Proposed method (Using SVM for the stage 2 of Figure 1)	0.03852
Proposed method (Using CNN for the stage 2 of Figure 1)	0.0082

4.3.3. Iris Segmentation Error with Another Open Database

We performed additional experiments with other open database of the mobile iris challenge evaluation (MICHE) data [63,64]. There are various databases of the CASIA datasets and iris challenge evaluation (ICE) datasets. The goal of this study is to obtain correct iris segmentation with the iris image captured by the visible light environment. However, very few open iris databases by visible light environment exist, and we used the MICHE datasets for this reason. They were collected by three mobile phones, such as iPhone 5, Galaxy Samsung IV, and Galaxy Tablet II in both indoor and outdoor environments. The ground truth images are not provided; therefore, among the whole images, we used the images where the ground truth iris regions can be obtained by their provided algorithm [56,58] according to the instruction of MICHE. Same procedure of two-fold cross validation was also used for the MICHE data as that for NICE-II dataset.

Figures 15 and 16 show the examples of good segmentation and incorrect segmentations by our proposed method, respectively. Like Figures 13 and 14, the false positive and false negative errors are shown in green and red, respectively. The true positive case (iris pixel is correctly classified into iris one) is shown in black. As shown in Figure 15, our method can correctly segment the iris region with the images captured by various cameras and environment. As shown in Figure 16, the false positive errors are caused by the eyelid region whose pixel values are similar to those of iris region. On the other hand, false negative errors happen in case of the reflection noises from glasses surface or environmental sunlight.

**Figure 15.** *Cont.*

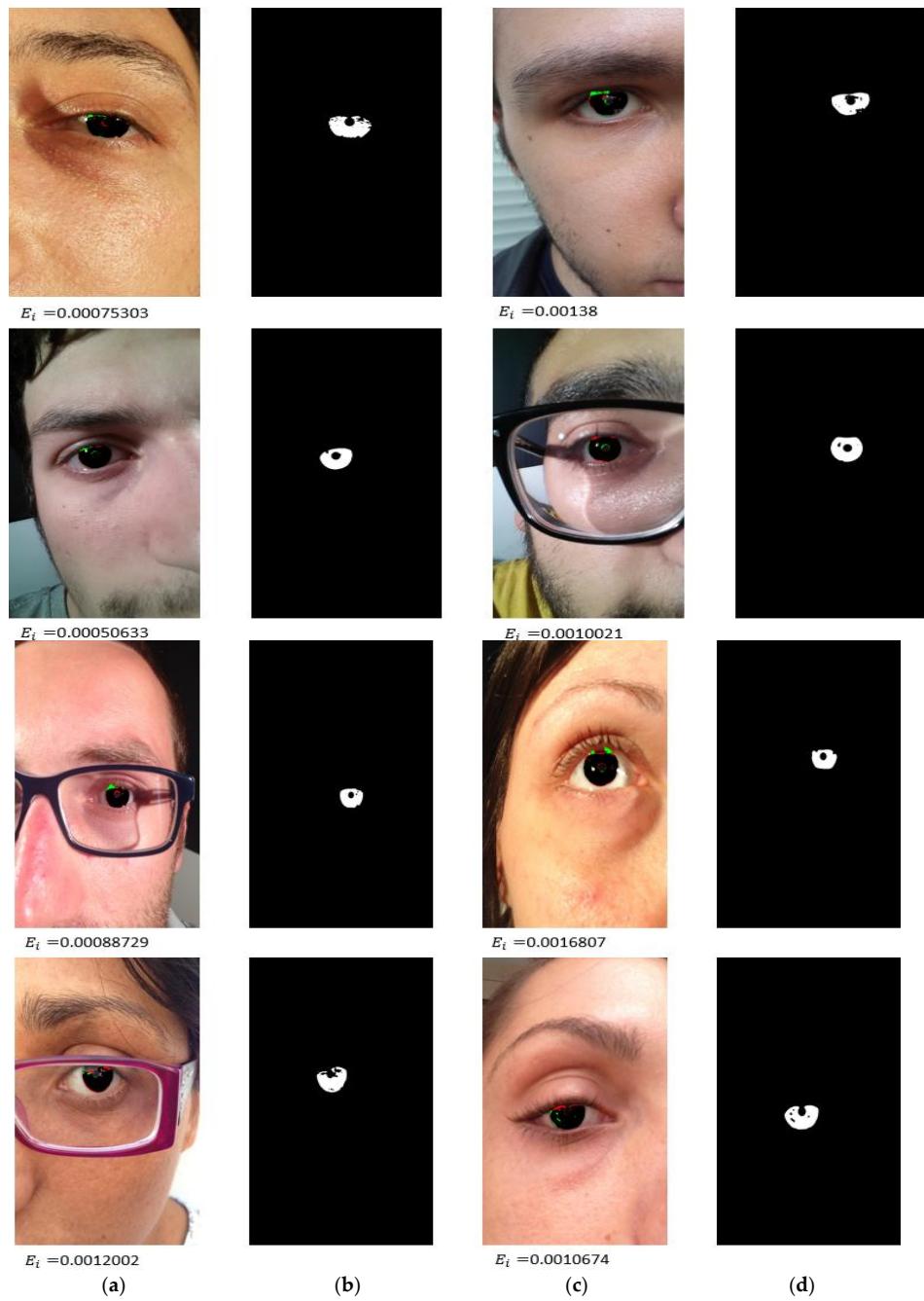


Figure 15. Examples of good segmentation results by our method. (a) And (c) segmentation results with corresponding E_i . (b) and (d) ground truth images (the false positive and false negative errors are shown in green and red, respectively. The true positive case (iris pixel is correctly classified into iris one) is shown in black).

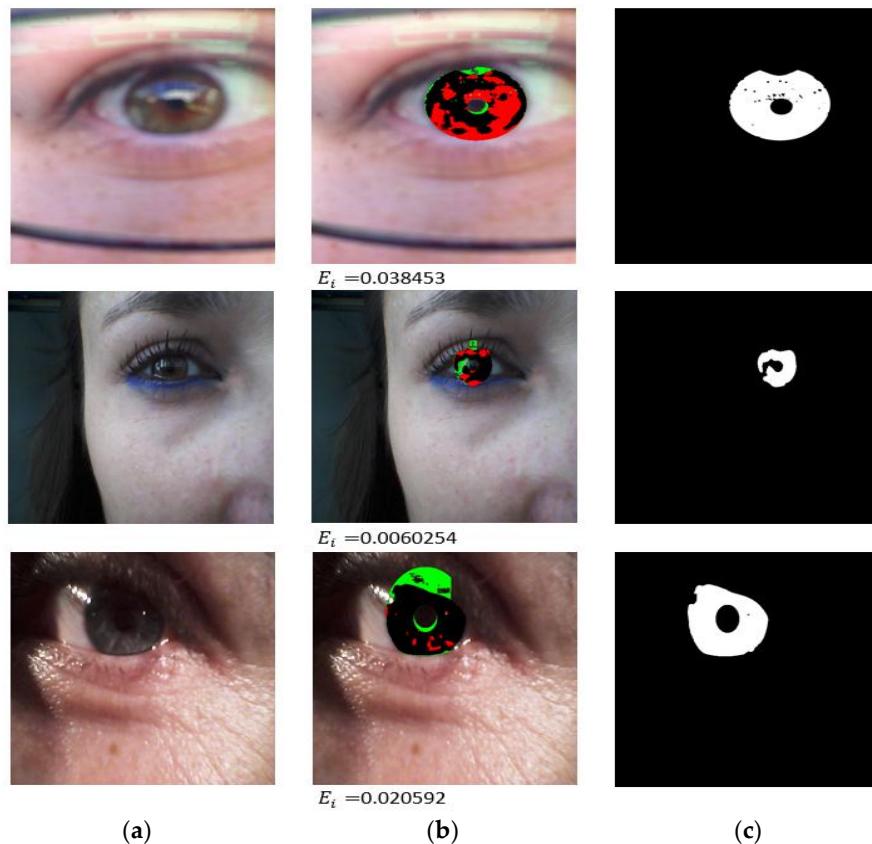


Figure 16. Examples of incorrect segmentation of the iris region by our method. (a) Original input images; (b) segmentation results with corresponding E_i ; (c) ground truth images (The false positive and false negative errors are shown in green and red, respectively. The true positive case (iris pixel is correctly classified into iris one) is shown in black).

In the next experiments, we compared the error of the iris segmentation by our proposed method with that by previous methods in terms of $E1$ of Equation (7). As shown in these results of Table 4, our proposed method outperforms the previous methods in iris segmentation error.

Table 4. Comparison of the proposed method with previous methods using MICHE dataset.

Method	$E1$	
Hu et al. [65]	Sub-dataset by iPhone 5	0.0193
	Sub-dataset by Galaxy Samsung IV	0.0192
Proposed method	Sub-dataset by iPhone 5	0.00368
	Sub-dataset by Galaxy Samsung IV	0.00297
	Sub-dataset by Galaxy Tablet II	0.00352

5. Conclusions

In this study, we proposed a robust two-stage CNN-based method, which can find the true iris boundary within noisy iris images in a non-cooperative environment. In the first stage, a rough iris boundary is found via modified circular HT, which defines the ROI by the slightly increased radius of the iris. In the second stage, CNN is applied by VGG-face fine-tuning to the data obtained from the ROI. The CNN output layer provides two output features. Therefore, based on these features, the iris and non-iris points are classified to find the true iris boundary. Experiments with NICE-II and MICHE

databases show that the proposed method achieved higher accuracies of iris segmentation compared to the state-of-the-art methods.

Although our method shows the high accuracy of iris segmentation, the traditional image processing algorithms should be used in the stage 1 of Figure 1. In addition, it is necessary to reduce the processing time for CNN-based classification with the window masks extracted from the stage 1. To solve these problems, we can consider the other type of CNN such as semantic segmentation network (SSN) which can use whole image as input (without the stages 1 and 2 of Figure 1). However, as shown in Table 3, its performance is lower than proposed method. As future work, we would research the method of using this SSN with appropriate post-processing so as to get high accuracy and fast processing speed. In addition, we would apply our proposed method to various iris datasets in NIR light environments, or other biometrics, such as vein segmentation for finger-vein recognition or human body segmentation in thermal image.

Acknowledgments: This research was supported by the Bio & Medical Technology Development Program of the NRF funded by the Korean government, MSIP (NRF-2016M3A9E1915855), by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03028417), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP; Ministry of Science, ICT & Future Planning) (NRF-2017R1C1B5074062).

Author Contributions: Muhammad Arsalan and Kang Ryoung Park designed the overall system for iris segmentation. In addition, they wrote and revised the paper. Hyung Gil Hong, Rizwan Ali Naqvi, Min Beom Lee, Min Cheol Kim, Dong Seop Kim, and Chan Sik Kim helped to design the CNN and comparative experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, P.; Zhang, R. The evolution of biometrics. In Proceedings of the IEEE International Conference on Anti-Counterfeiting Security and Identification in Communication, Chengdu, China, 18–20 July 2010; pp. 253–256.
2. Larrain, T.; Bernhard, J.S.; Mery, D.; Bowyer, K.W. Face recognition using sparse fingerprint classification algorithm. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 1646–1657. [[CrossRef](#)]
3. Bonnen, K.; Klare, B.F.; Jain, A.K. Component-based representation in automated face recognition. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 239–253. [[CrossRef](#)]
4. Daugman, J. How iris recognition works. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 21–30. [[CrossRef](#)]
5. Viriri, S.; Tapamo, J.R. Integrating iris and signature traits for personal authentication using user-specific weighting. *Sensors* **2012**, *12*, 4324–4338. [[CrossRef](#)] [[PubMed](#)]
6. Jain, A.K.; Arora, S.S.; Cao, K.; Best-Rowden, L.; Bhatnagar, A. Fingerprint recognition of young children. *IEEE Trans. Inf. Forensic Secur.* **2017**, *12*, 1501–1514. [[CrossRef](#)]
7. Hong, H.G.; Lee, M.B.; Park, K.R. Convolutional neural network-based finger-vein recognition using NIR image sensors. *Sensors* **2017**, *17*, 1297. [[CrossRef](#)] [[PubMed](#)]
8. Yixin, Z.; Huanhuan, L.; Xuefei, G.; Lili, L. Palmprint recognition based on multi-feature integration. In Proceedings of the IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, Xi'an, China, 3–5 October 2016; pp. 992–995.
9. Daugman, J. Information theory and the IrisCode. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 400–409. [[CrossRef](#)]
10. Hsieh, S.-H.; Li, Y.-H.; Tien, C.-H. Test of the practicality and feasibility of EDoF-empowered image sensors for long-range biometrics. *Sensors* **2016**, *16*, 1994. [[CrossRef](#)] [[PubMed](#)]
11. He, Z.; Tan, T.; Sun, Z.; Qiu, X. Toward accurate and fast iris segmentation for iris biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 1670–1684. [[PubMed](#)]
12. Li, Y.-H.; Savvides, M. An automatic iris occlusion estimation method based on high-dimensional density estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 784–796. [[CrossRef](#)] [[PubMed](#)]
13. Matey, J.R.; Naroditsky, O.; Hanna, K.; Kolczynski, R.; LoIacono, D.J.; Mangru, S.; Tinker, M.; Zappia, T.M.; Zhao, W.Y. Iris on the move: Acquisition of images for iris recognition in less constrained environments. *Proc. IEEE* **2006**, *94*, 1936–1947. [[CrossRef](#)]
14. Wildes, R.P. Iris recognition: An emerging biometric technology. *Proc. IEEE* **1997**, *85*, 1348–1363. [[CrossRef](#)]

15. Uhl, A.; Wild, P. Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation. In Proceedings of the 5th IEEE International Conference on Biometrics, New Delhi, India, 29 March–1 April 2012; pp. 283–290.
16. Zhao, Z.; Kumar, A. An accurate iris segmentation framework under relaxed imaging constraints using total variation model. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3828–3836.
17. Roy, D.A.; Soni, U.S. IRIS segmentation using Daughman’s method. In Proceedings of the IEEE International Conference on Electrical, Electronics, and Optimization Techniques, Chennai, India, 3–5 March 2016; pp. 2668–2676.
18. Jeong, D.S.; Hwang, J.W.; Kang, B.J.; Park, K.R.; Won, C.S.; Park, D.-K.; Kim, J. A new iris segmentation method for non-ideal iris images. *Image Vis. Comput.* **2010**, *28*, 254–260. [\[CrossRef\]](#)
19. Tan, T.; He, Z.; Sun, Z. Efficient and robust segmentation of noisy iris images for non-cooperative iris recognition. *Image Vis. Comput.* **2010**, *28*, 223–230. [\[CrossRef\]](#)
20. Khan, T.M.; Khan, M.A.; Malik, S.A.; Khan, S.A.; Bashir, T.; Dar, A.H. Automatic localization of pupil using eccentricity and iris using gradient based method. *Opt. Lasers Eng.* **2011**, *49*, 177–187. [\[CrossRef\]](#)
21. Parikh, Y.; Chaskar, U.; Khakole, H. Effective approach for iris localization in nonideal imaging conditions. In Proceedings of the IEEE Students’ Technology Symposium, Kharagpur, India, 28 February–2 March 2014; pp. 239–246.
22. Shah, S.; Ross, A. Iris segmentation using geodesic active contours. *IEEE Trans. Inf. Forensic Secur.* **2009**, *4*, 824–836. [\[CrossRef\]](#)
23. Abdullah, M.A.M.; Dlay, S.S.; Woo, W.L.; Chambers, J.A. Robust iris segmentation method based on a new active contour force with a noncircular normalization. *IEEE Trans. Syst. Man Cybern.* **2017**, *1*–14, in press. [\[CrossRef\]](#)
24. Chai, T.-Y.; Goi, B.-M.; Tay, Y.H.; Chin, W.-K.; Lai, Y.-L. Local Chan-Vese segmentation for non-ideal visible wavelength iris images. In Proceedings of the IEEE Conference on Technologies and Applications of Artificial Intelligence, Tainan, Taiwan, 20–22 November 2015; pp. 506–511.
25. Li, H.; Sun, Z.; Tan, T. Robust iris segmentation based on learned boundary detectors. In Proceedings of the International Conference on Biometrics, New Delhi, India, 29 March–1 April 2012; pp. 317–322.
26. Vokhidov, H.; Hong, H.G.; Kang, J.K.; Hoang, T.M.; Park, K.R. Recognition of damaged arrow-road markings by visible light camera sensor based on convolutional neural network. *Sensors* **2016**, *16*, 2160. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Nguyen, D.T.; Kim, K.W.; Hong, H.G.; Koo, J.H.; Kim, M.C.; Park, K.R. Gender recognition from human-body images using visible-light and thermal camera videos based on a convolutional neural network for image feature extraction. *Sensors* **2017**, *17*, 637. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Kim, J.H.; Hong, H.G.; Park, K.R. Convolutional neural network-based human detection in nighttime images using visible light camera sensors. *Sensors* **2017**, *17*, 1–26.
29. Pereira, S.; Pinto, A.; Alves, V.; Silva, C.A. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans. Med. Imaging* **2016**, *35*, 1240–1251. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Liu, N.; Zhang, M.; Li, H.; Sun, Z.; Tan, T. DeepIris: Learning pairwise filter bank for heterogeneous iris verification. *Pattern Recognit. Lett.* **2016**, *82*, 154–161. [\[CrossRef\]](#)
31. Gangwar, A.; Joshi, A. DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition. In Proceedings of the IEEE International Conference on Image Processing, Phoenix, USA, 25–28 September 2016; pp. 2301–2305.
32. Liu, N.; Li, H.; Zhang, M.; Liu, J.; Sun, Z.; Tan, T. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In Proceedings of the IEEE International Conference on Biometrics, Halmstad, Sweden, 13–16 June 2016; pp. 1–8.
33. Dongguk Visible Light Iris Segmentation CNN Model (DVLIS-CNN). Available online: <http://dm.dgu.edu/link.html> (accessed on 27 July 2017).
34. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep face recognition. In Proceedings of the British Machine Vision Conference, Swansea, UK, 7–10 September 2015; pp. 1–12.
35. CS231n Convolutional Neural Networks for Visual Recognition. Available online: <http://cs231n.github.io/neural-networks-1/> (accessed on 26 July 2017).

36. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
37. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 807–814.
38. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*; Curran Associates, Inc.: New York, NY, USA, 2012; pp. 1097–1105.
39. Heaton, J. Artificial Intelligence for Humans. In *Deep Learning and Neural Networks*; Heaton Research, Inc.: St. Louis, USA, 2015; Volume 3.
40. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
41. Wyatt, H.J. A ‘minimum-wear-and-tear’ meshwork for the iris. *Vision Res.* **2000**, *40*, 2167–2176. [CrossRef]
42. NICE.II. Noisy Iris Challenge Evaluation-Part II. Available online: <http://nice2.di.ubi.pt/index.html> (accessed on 26 July 2017).
43. Proen  a, H.; Filipe, S.; Santos, R.; Oliveira, J.; Alexandre, L.A. The UBIRIS.v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1529–1535.
44. Geforce GTX 1070. Available online: <https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1070/> (accessed on 4 August 2017).
45. Caffe. Available online: <http://caffe.berkeleyvision.org> (accessed on 4 August 2017).
46. Stochastic Gradient Descent. Available online: https://en.wikipedia.org/wiki/Stochastic_gradient_descent (accessed on 26 July 2017).
47. Caffe Solver Parameters. Available online: <https://github.com/BVLC/caffe/wiki/Solver-Prototxt> (accessed on 4 August 2017).
48. NICE.I. Noisy Iris Challenge Evaluation-Part I. Available online: <http://nice1.di.ubi.pt/> (accessed on 26 July 2017).
49. Girshick, R. Fast R-CNN. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
50. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv* **2016**, *1*–14. [CrossRef] [PubMed]
51. Semantic Segmentation Using Deep Learning. Available online: <https://kr.mathworks.com/help/vision/examples/semantic-segmentation-using-deep-learning.html> (accessed on 23 October 2017).
52. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv* **2015**, *1*–14. [CrossRef] [PubMed]
53. Luengo-Oroz, M.A.; Faure, E.; Angulo, J. Robust iris segmentation on uncalibrated noisy images using mathematical morphology. *Image Vis. Comput.* **2010**, *28*, 278–284. [CrossRef]
54. Labati, R.D.; Scotti, F. Noisy iris segmentation with boundary regularization and reflections removal. *Image Vis. Comput.* **2010**, *28*, 270–277. [CrossRef]
55. Chen, Y.; Adjouadi, M.; Han, C.; Wang, J.; Barreto, A.; Rishe, N.; Andrian, J. A highly accurate and computationally efficient approach for unconstrained iris segmentation. *Image Vis. Comput.* **2010**, *28*, 261–269. [CrossRef]
56. Li, P.; Liu, X.; Xiao, L.; Song, Q. Robust and accurate iris segmentation in very noisy iris images. *Image Vis. Comput.* **2010**, *28*, 246–253. [CrossRef]
57. Tan, C.-W.; Kumar, A. Unified framework for automated iris segmentation using distantly acquired face images. *IEEE Trans. Image Process.* **2012**, *21*, 4068–4079. [CrossRef] [PubMed]
58. Proen  a, H. Iris recognition: on the segmentation of degraded images acquired in the visible wavelength. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1502–1516. [CrossRef] [PubMed]
59. De Almeida, P. A knowledge-based approach to the iris segmentation problem. *Image Vis. Comput.* **2010**, *28*, 238–245. [CrossRef]
60. Tan, C.-W.; Kumar, A. Towards online iris and periocular recognition under relaxed imaging constraints. *IEEE Trans. Image Process.* **2013**, *22*, 3751–3765. [PubMed]

61. Sankowski, W.; Grabowski, K.; Napieralska, M.; Zubert, M.; Napieralski, A. Reliable algorithm for iris segmentation in eye image. *Image Vis. Comput.* **2010**, *28*, 231–237. [[CrossRef](#)]
62. Haindl, M.; Krupička, M. Unsupervised detection of non-iris occlusions. *Pattern Recognit. Lett.* **2015**, *57*, 60–65. [[CrossRef](#)]
63. MICHE Dataset. Available online: http://biplab.unisa.it/MICHE/index_miche.htm (accessed on 4 August 2017).
64. Marsico, M.; Nappi, M.; Ricco, D.; Wechsler, H. Mobile iris challenge evaluation (MICHE)-I, biometric iris dataset and protocols. *Pattern Recognit. Lett.* **2015**, *57*, 17–23. [[CrossRef](#)]
65. Hu, Y.; Sirlantzis, K.; Howells, G. Improving colour iris segmentation using a model selection technique. *Pattern Recognit. Lett.* **2015**, *57*, 24–32. [[CrossRef](#)]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).