Chapter 6
Iris Segmentation Using Fully Convolutional Encoder–Decoder Networks

Ehsaneddin Jalilian and Andreas Uhl

Abstract As a considerable breakthrough in artificial intelligence, deep learning has gained great success in resolving key computer vision challenges. Accurate segmentation of the iris region in the eye image plays a vital role in efficient performance of iris recognition systems, as one of the most reliable systems used for biometric identification. In this chapter, as the first contribution, we consider the application of Fully Convolutional Encoder–Decoder Networks (FCEDNs) for iris segmentation. To this extent, we utilize three types of FCEDN architectures for segmentation of the iris in the images, obtained from five different datasets, acquired under different scenarios. Subsequently, we conduct performance analysis, evaluation, and comparison of these three networks for iris segmentation. Furthermore, and as the second contribution, in order to subsidize the true evaluation of the proposed networks, we apply a selection of conventional (non-CNN) iris segmentation algorithms on the same datasets, and similarly evaluate their performances. The results then get compared against those obtained from the FCEDNs. Based on the results, the proposed networks achieve superior performance over all other algorithms, on all datasets.

6.1 Introduction

Deep learning techniques and convolutional neural networks (CNNs), in specific, are driving advances in artificial intelligence, as powerful visual recognition, classification and segmentation tools. Iris recognition is one of the most reliable and accurate biometric technologies used for human identification and authentication. The iris encloses many unique features, which make it a good candidate for distinguishing one person from another. Primary, the trabecular meshwork of the iris tissue is not

E. Jalilian (✉) · A. Uhl
Department of Computer Sciences, University of Salzburg, Jakob Haringer Str. 2,
5020 Salzburg, Austria
e-mail: ejalilian@cosy.sbg.ac.at

A. Uhl
e-mail: uhl@cosy.sbg.ac.at

B. Bhanu and A. Kumar (eds.), *Deep Learning for Biometrics*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-3-319-61657-5_6

genetically influenced during its development, has in excess of "266 degrees of freedom," [6] and it is protected behind the eyelid, cornea and aqueous humour. Like any other biometrics system, the performance of iris recognition systems is highly depended on the accurate segmentation of the target region (iris) from the rest of image [31, 51]. And, to a great extent, the success or failure of the entire iris recognition system is considered to be directly dependent on the precision of this stage [5, 17].

6.1.1 Conventional (Non-CNN) Iris Segmentation Methods

Over the past decades, a lot of conventional (non-CCN) methods are proposed for iris segmentation. A quick review of related literatures unveils a significant number of these methods, which in turn enjoy versatile capabilities in iris segmentation [1, 7, 26, 36, 44, 47, 50]. In general, segmentation methods can be roughly classified into two main categories: contour-based methods and texture-based methods. The most well known contour-based methods are based on integro-differential operators [8], and Hough transforms [49]. The principles of integro-differential algorithms are based on search to find the largest difference of intensity over a parameter space, which normally corresponds to pupil and iris boundaries. Hough transform methods, however, try to find optimal circle parameters by exploring binary edge maps.

Performance of these methods is highly dependent on the images' clear contour and the boundary contrast. Often, in normal conditions, limbic or pupillary boundaries in the images are of low-contrast, or may be of non-circular shape. In addition, the occlusions and specular reflections may introduce further contrast defects to the images. While plenty of improvement such as occlusion detection [18, 21], circle model improvement [29, 45], deformation correction [10], noise reduction [23], boundary fitting [47], and many further methods are introduced to compensate for such defects, yet due to their global approach in segmentation, the performance of these methods can be undermined by these defects, or even in some cases, they may result in total failure of system.

On the other hand, texture-based methods exploit the individual pixel's visual aspects information, such as intensity, color, etc. to classify the iris pixels from the rest of image. The most promising methods in this category use some commonly known pixel-wise image classifiers such as: support vector machines (SVMs) [39], Neural networks [4], and Gabor filters [34] to classify iris pixels from the rest of image. In spite of the efforts to improve the performance of these group of algorithms [26, 43], yet these methods similarly suffer from the same group of defects such as diffusion, reflection and occlusion.

6.1.2 Convolutional Neural Networks (CNNs) for Segmentation

Convolutional neural networks (CNNs) have dramatically improved the state of the art in many image interpretation and analysis tasks, such as image segmentation and classification, since their early introduction [24, 25]. The principle of convolutional neural networks is based on hierarchical explanatory factors, where higher level, more abstract, concepts are learned from the lower level ones, with the help of convolutional operators. Assuming A_{ij} as the data vector at location (i, j) in a particular layer, and B_{ij} for the following layer, the output B_{ij} is computed as:

$$B_{ij} = f_{\alpha\beta}(\{A_{\beta_i+\delta_i,\beta_j+\delta_j}\}\ 0 \leq_{i,j} \leq \alpha)$$

where α is the convolutional kernel size, β is the stride or sub-sampling factor, and $f_{\alpha\beta}$ specifies the layer type. As already mentioned, the core building blocks of CNNs are the convolutional layers (Conv). The convolutional layers' parameters consist of a set of learnable filters. Each filter convolves the input volume and computes the dot product between the entries of the filter and the input at any position, and produces an activation map that gives the responses of that filter at every spatial position. The output of this layer then can be further processed by additional layers such as: nonlinear down-sampling layer (Pool), non-saturating activation function layer (ReLU), and further layers depending on the networks' architectures. All together these layers form the networks' encoder compartment. In addition to the encoding compartment, each network includes a decoding mechanism, in which the main task of labelling is performed, and the network's output is delivered as the classification scores.

The initial versions of convolutional neural networks were developed for classification tasks, where the networks' output was a single class label [22, 41]. In this type of networks, usually a fully connected layer (FC), as an inner-product function, was used to generate the classification scores. However, in many image processing tasks, as in iris segmentation, pixel-wise labelling of the region of interest was required. Prior approaches for pixel-wise segmentation used convents [14, 15]. In more recent CNNs' architectures, including the Fully Convolutional Neural Networks (FCNNs), usually a learnable upsampling layer (Upsample) is used to retrieve the feature maps, and then a softmax layer (Softmax), which normally computes the multinomial logistic loss of the softmax of its inputs, is employed to generate the classification scores. There exist various methods for enabling such upsampling mechanism [30, 46].

In Convolutional Encoder–Decoder Networks (CEDNs) the encoding mechanism, already explained, is repeated in the reverse mode to upsample the low-resolution output maps of the encoder network to full input resolution features of the input volume. Likewise, various methods are proposed for the upsampling mechanism in this type of networks. While some used switch codes to remap the features [32], others, like Ronneberger et al. [40], simply used cropped feature concatenation to generate the upsampled maps. In addition to using a primitive approach for upsampling,

the network proposed by Ronneberger et al. is not fully convolutional, as the input and the output volumes do not match. The proposed mechanism to retrieve the full input volume (Overlap-tile) affects the training process anyway, and introduces further pre and postprocessing work load. The idea of using max-pooling indices from the corresponding encoder feature maps to generate the upsampled feature maps in the decoder network, "without learning," and then convolving with trainable filter banks, is recently proposed in a work by Badrinarayanan et al. [2]. The key leverage of applying such technique is improving the boundary delineation and preserving the edges with more precision.

6.2 CNNs in Iris Recognition

Nevertheless, when it comes to the application of deep learning techniques for iris segmentation, to the best of our knowledge, there exists only one proposal work on application of convolutional neural networks for iris segmentation. In [27] authors proposed two CNN-based iris segmentation models, namely: Hierarchical convolutional neural network (HCNN), and multi-scale fully convolutional network (MFCN). The former network is a simple CNN network composed of three blocks of alternative Conv and Pool layers, whose outputs are fed directly into a FC layer. The latter network includes six blocks of interconnected alternative Conv and Pool layers, whose outputs are simply fused through a single multiplication layer and then fed into a Softmax layer. The proposed networks are used for segmenting noisy-distanced iris images acquired from the Ubiris.v2,[1] and the Casia-distance-v4 databases.[2] Authors used a subset (500 and 400 images respectively) of these two databases. For the evaluation, authors referenced average error rates from other works, without carrying out direct experimental analysis on the same databases. Therefore, their results have to be considered with care, when it comes to the fair evaluation of segmentation accuracy. The ground-truth masks for the second database are manually labelled by the authors.

In [13], authors introduced two convolutional neural networks for iris recognition. But, these are proposed for "iris representation," not for segmentation. The networks are named "DeepIrisNet-A," which is based on the standard convolutional layers, and "DeepIrisNet-B," which uses a stack of so called "inception layers". Based on the results, the networks can model the micro-structures of iris well, and primarily outperform strong baseline based on descriptor and generalize well to new datasets.

In [35], authors used deep sparse filtering technique for iris feature extraction in a smart-phone based visible iris recognition system. In this work two layers of sparse filters are trained with 256 filters, with 16×16 kernels, to generate the feature

[1] Soft Computing and Image Analysis Lab, Univ. of Beira Interior, UBIRIS.v2 Dataset, see http://iris.di.ubi.pt/ubiris1.html.

[2] The Center of Biometrics and Security Research, CASIA Iris Image Database, see http://biometrics.idealtest.org.

maps. The final feature vector is formed by concatenating the histograms of these feature maps. They tested their system on many smart-phones and, depending on the smart-phone type, achieved different levels of accuracies. In [28] authors proposed a CNN, called "DeepIris," for heterogeneous iris verification, which learns relational features to measure the similarity between pairs of iris images. The network's architecture includes a pairwise filter layer, and a stock of alternative Pool, Conv, and Normalization (NR) layers, to generate a similarity map between two input images. The similarity measure is calculated as a scores with help of a FC layer at the end. Their results show that the proposed method achieves promising performance for both cross-resolution and cross-sensor iris verification.

6.3 A Fully Convolutional Encoder–Decoder Network

The architecture of the FCEDN networks used by us is similar to the work by Badrinarayanan et al. [2]. However, here we redesigned the Softmax layer to classify the outputs only into two classes (iris, and non-iris). Basically, this architecture proposes a fully convolutional encoder–decoder network, representing a core segmentation engine for pixel-wise semantic segmentation [2].

The core segmentation engine includes a 44-layered encoder network, and the corresponding decoder network. The encoder network's architecture is organized in five stocks. Each stock is comprised of a set of blocks, whose architectures are formed by a Conv layer, which convolves the input image with a kernel to produce the inputs' feature maps, followed by a batch normalized layer (BN), to normalize the layer input and avoid the "internal covariate shift" [19], and an element-wise rectified-linear nonlinearity layer (ReLU), as an activation function. The blocks then end up in a Pool layer (with a 2×2 window and stride 2), which applies nonlinear down-sampling to the input and achieves translation invariance over small spatial shifts. While applying several Pool layers can help to obtain robust translation invariance, yet applying each layer leads to the loss of spatial resolution, specially in the boundary regions. This issue is resolved by storing max pooling indices, which are latter used for upsampling in the decoding network, in these networks.

The corresponding decoder network, likewise, has a 44-layered structure, which encompasses five stocks. Similarly, each stock is comprised of a set of blocks, whose architectures are formed from an Upsample layer, and trainable banks of decoder filters, including Conv, BN and ReLU layers, to generate the dense feature maps. As already specified, the Upsample layer uses max-pooling indices from the corresponding encoder feature maps to generate the upsampled feature maps, without learning and then the filter banks convolve the maps. Finally, the results are fed into a trainable Softmax layer (SoftmaxWithLoss). The Softmax layer classifies each pixel independently, so that the output of this layer is a N channel image of probabilities, where N is the number of classes. The output segmentation corresponds to the class with highest probability at each pixel. Applying this technique directly results in improvement

of the boundary delineations and preservation of the edges. At the same time, this technique reduces the number of parameters, enabling end-to-end training.

6.3.1 Basic Variant

As we specified in the previous section, the overall structure of the network is organized as set of blocks, which in fact represent the network's learning units. Such a distinct unified structure expedites the modification of the Original network's architecture to best fit the required segmentation tasks. To this extent, and with the primary goal of facilitating the network analysis, an abstract variation of the Original network called "Basic" was introduced [2]. Using such an abstract architecture enables us to evaluate the effect of using less learning, and as the result, less convolutional and down-sampling units in the segmentation process. At the same time, such an architecture can offer faster and less computational expensive segmentation capabilities also. The overall network architecture, similarly, is composed of an encoder and the corresponding decoder networks. The encoder comprises four stocks, whose structures are similar to the Original network's blocks, incorporating Conv, BN, ReLU, and Pool layers. The decoder network's blocks, as well, include Upsample, Conv, and BN layers. The decoder network finally ends up to a SoftmaxWithloss layer.

6.3.2 Bayesian Variant

As another extension to the Original network, and as an attempt to propose a probabilistic pixel-wise segmentation model based on the deep learning technique, another variation of the Original network called "Baysian" was developed [20]. This network enables the probabilistic pixel-wise segmentation using Monte-Carlo sampling and the drop-out technique [12, 42]. The aim here is to find the posterior distribution over the convolutional weights w, given the observed training data x and the labels z.

$$p(w|x, z)$$

In practice, such a posterior distribution can only be approximated, for example, with variational inference techniques, such as minimization of the Kullback–Leibler (kl) divergence between the desired approximated distribution and the full posterior distribution [9].

$$kl(q(w) \,||\, p(w|x, z))$$

Gal et al. [11] have already shown that minimizing the cross entropy loss objective function pretends to minimizing the Kullback–Leibler divergence term. Accordingly, they have proved that training neural networks with the stochastic gradient descent will promote the model to learn the distribution of weights, while avoiding

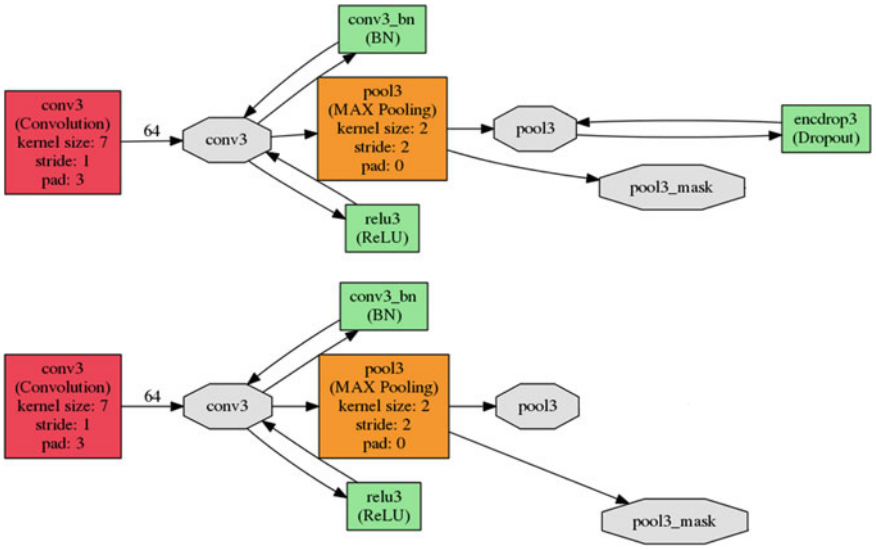

Fig. 6.1 Architectural difference of the third block in the Bayesian-Basic (*up*), and the Basic (*down*) networks (Images are generated using Caffe framework)

over-fitting. Using the same technique, the overall architecture for the "Bayesian-Basic" network would be the same as the Basic network, except for this network includes extra drop-out layers, which are added to the two last blocks of the encoder, and the first two blocks of the decoder networks, as demonstrated in Fig. 6.1. Using this architecture, the posterior distribution over the weights would be sampled at the test time to generate the distribution of softmax class probabilities. The mean of this samples are taken for the segmentation prediction, and the variance is used to output the model uncertainty for each class. Monte-Carlo sampling is used for this purpose as it enables the qualitative understanding of the model uncertainty.

6.4 Experimental Framework

The main objective of this research is the application of FCEDNs for iris segmentation, and subsequently, providing evaluation and analysis of these networks' performance for different scenarios. For these purpose, after design and implementation of networks, each network was run on five different iris datasets, containing images acquired under different scenarios. The detailed specifications of these datasets are discussed in Sect. 4.1. The segmentation capability of each network then was evaluated and analysed with the help of three different segmentation scores, whose details are specified in Sect. 4.2. Next, in order to facilitate true assessment of the performance of networks for iris segmentation, a collection of conventional

iris segmentation algorithms, whose details are specified in Sect. 6.6, got run on the same datasets' testing subsets, and their corresponding performance analysis and evaluation was presented, and compared against those of the FCEDNs.

6.4.1 Datasets

In this work we have selected five well-known iris datasets. The datasets were selected to include versatile iris image acquisition scenarios. The Notredame dataset (subset of ND-Iris-0405 database[3]) includes 835 iris images of 30 different subjects. The image acquisition was done in near-infrared spectrum, in an indoor environment, with the LG 2200 iris biometric system. For our experiments we used 670 images (24 subjects) for the training, and 165 images (six subjects) for the testing of the networks. The Casia-iris-interval-v4 dataset[4] contains a total of 2640 iris images belonging to 249 subjects. Images are acquired under near-infrared illumination, with a close-up iris camera. For our experiments 1307 instances of the right eye images were used, out of which 1045 images (192 subjects) were used for the training, and 262 image (57 subjects) were used for the testing of the networks.

The IITD database[5] consists 1120 iris images corresponding to 224 subjects. All these images are acquired in the indoor environment, with the Jiris, Jpc1000 digital CMOS camera in near-infrared spectrum. For our experiments 900 images (180 subjects) were used for the training, and 220 images (44 subjects) were used for the testing of the networks. The Ubiris dataset (subset of the Ubiris.v2 database) contains 2250 iris images, from 100 different subjects. The images were acquired with a Nikon E5700 camera and split into two parts. The first part includes iris images taken under controlled condition, simulating the enrolment stage. The second part includes iris images which are captured under real-world setup, with natural luminosity corresponding heterogeneity in reflections, contrast and focus. The dataset also contains off-angle iris images captured from various distances with occlusions. For our experiments we used 2055 images of this dataset for the training and 225 images for the testing of the networks.

And finally, the Casia-iris-ageing-v5 dataset[6] contains 120 images per eye and user from video sequences captured in 2009, and 20 images per eye and user from video sequences captured in 2013. For our experiments we used total of 1880 images of both eyes of 94 users from both sessions. Out of that, 1500 images of 75 users were used for the training, and 380 images, corresponding to 19 users, were used for the

[3]Computer Vision Research Lab, Univ. of Notre Dame, Iris Dataset 0405, see https://sites.google.com/a/nd.edu/public-cvrl/data-sets.

[4]The Center of Biometrics and Security Research, CASIA Iris Image Database, see http://biometrics.idealtest.org.

[5]Indian Institute of Technology Delhi, IIT Delhi Iris Database, see http://www4.comp.polyu.edu.hk/~csajaykr/database.php.

[6]See http://www.biometrics.idealtest.org.

testing of the networks. Special attention should be paid to the fact that the selection of the datasets was subject to availability of the ground-truth masks required for the training process. For this work the ground-truth masks were acquired from the Irisseg-ep database provided by WaveLab of the University of Salzburg [16]. The selection of the training and testing subsets followed the Pareto principle, where of total instances, approximately 80% of the data was used for training and 20% for testing, while subjects are not overlapped, and no instances are included in other's set. Also in order to maintain fair spatial input to the networks, images in all datasets was resized to 480×360.

6.4.2 Metrics and Measurements

In order to facilitate holistic statistical analysis and proper assessment of the capabilities of the FCEDNs and the other conventional algorithms on iris segmentation, we have considered a set of evaluation metrics, which cover key segmentation measures such as: true positives (tp), false negatives (fn), and false positive (fp). To this extent, primarily the NICE.I protocol, which is widely accepted for evaluation of iris segmentation accuracy, got adapted. The segmentation error score nice1 calculates the proportion of corresponding disagreeing pixels (by the logical exclusive-or operator) over all the image as follows:

$$nice1 = \frac{1}{c \times r} \sum_{c'} \sum_{r'} O(c', r') \otimes C(c', r') \tag{6.1}$$

where c and r are the columns and rows of the segmentation masks, and $O(c', r')$ and $C(c', r')$ are, respectively, pixels of the output and the ground-truth mask. The second segmentation error score intends to compensate the disproportion between the priori probabilities of iris and non-iris pixels in the images. The type-I and type-II error score nice2 averages between the (fp) and (fn) rates as follow

$$nice2 = \frac{1}{2} (fp + fn) \tag{6.2}$$

The values of nice1 and nice2 are bounded in the [0, 1] interval, and in this context, 1 and 0 are respectively the worst and the optimal values. Additionally, in order to provide comprehensive synopsis of the networks' performances, three more standard measures of segmentation accuracy were considered, namely: precision, recall, and f1 measure, which are well-known measures in the field of information retrieval [38]. Precision gives the percentage of retrieved iris pixels which are correctly segmented as follows:

$$p = \frac{tp}{tp + fp} \tag{6.3}$$

Alternatively, recall provides the same measure using false negatives as follows:

$$r = \frac{tp}{tp + fn} \tag{6.4}$$

Last but not least, the f1 measure is the harmonic mean of p and r, and is calculated as follows:

$$f1 = \frac{2rp}{r + p} \tag{6.5}$$

The values for these three measures are bounded in the [1, 0] interval, and in this context, 0 and 1 are the worst and the optimal values respectively.

6.4.3 Network Implementation

In the first step, we implemented the FCEDNs on the "Caffe" deep learning framework. Caffe is one of the most favourited deep learning frameworks, which at its core, is written in c++. Models and optimizations are defined by configuration without hard-coding, which in turn accelerate the training and testing process. Switching between CPU and GPU can be done just by setting a single flag, and the interface is extensible to python (Pycaffe) and matlab (Matcaffe). Network architecture is defined in separate "prototxt" files, and training and testing parameters are defined in another similar file called "solver". The prototxt files get loaded during the training and testing using caffe commands.

Architectural implementation and the technical specification for the Original, Basic, and Bayesian-Basic networks are presented in Tables 6.1, 6.2, and 6.3 respectively. The convolutional kernel size for the Original network was set to 3×3, and in order to provide a wide context for smooth labelling, this value was set to 7×7 for both of the Basic networks.

We trained our networks by Stochastic Gradient Descent (SGD) back propagation algorithm

$$U_{t+1} = \mu U_t - \alpha \nabla L(W) \tag{6.6}$$

$$W_{t+1} = W_t + U_{t+1} \tag{6.7}$$

Formally, at each iteration $t + 1$ the SGD algorithm computes the update value U_{t+1} and the updated weights W_{t+1}, given the previous weight update U_t and current weights W_t. The algorithm updates the weights W by linear combination of the negative gradient $\nabla L(W)$ and the previous weight update U_t. The learning rate α is the weight of the negative gradient, and the momentum μ is the weight of the previous update.

For our experiments, the momentum value for the SGD algorithm was set to 0.9. In order to investigate the networks' training process, we have considered two

Table 6.1 Architecture and specification of the original encoder (left)–decoder (right) network

Layer	Information	Layer	Layer	Layer	Information	Layer	Layer
Convolution	Output	Batch-Normalization	ReLU	Convolution	Output	Batch-Normalization	ReLU
Conv1-1	64	Conv1-1-b	Rlu1-1	Conv5-3-D	512	Conv5-3-D-b	Rlu5-3-D-b
Conv1-2	64	Conv1-2-b	Rlu1-2	Conv5-2-D	512	Conv5-2-D-b	Rlu5-2-D-b
Conv2-1	128	Conv2-1-b	Rlu2-1	Conv5-1-D	512	Conv5-1-D-b	Rlu5-1-D-b
Conv2-2	128	Conv2-2-b	Rlu2-2	Conv4-3-D	512	Conv4-3-D-b	Rlu4-3-D-b
Conv3-1	256	Conv3-1-b	Rlu3-1	Conv4-2-D	512	Conv4-2-D-b	Rlu4-2-D-b
Conv3-2	256	Conv3-2-b	Rlu3-2	Conv4-1-D	256	Conv4-1-D-b	Rlu4-1-D-b
Conv3-3	256	Conv3-3-b	Rlu3-3	Conv3-3-D	256	Conv3-3-D-b	Rlu3-3-D-b
Conv4-1	512	Conv4-1-b	Rlu4-1	Conv3-2-D	256	Conv3-2-D-b	Rlu3-2-D-b
Conv4-2	512	Conv4-2-b	Rlu4-2	Conv3-1-D	128	Conv3-1-D-b	Rlu3-1-D-b
Conv4-3	512	Conv4-3-b	Rlu4-3	Conv2-2-D	128	Conv2-2-D-b	Rlu2-2-D-b
Conv5-1	512	Conv5-1-b	Rlu5-1	Conv2-1-D	64	Conv2-1-D-b	Rlu2-1-D-b
Conv5-2	512	Conv5-2-b	Rlu5-2	Conv1-2-D	64	Conv1-2-D-b	Rlu1-2-D-b
Conv5-3	512	Conv5-3-b	Rlu5-3	Conv1-1-D	2	Conv1-1-D-b	
Pooling	Stride	Kernel size		Upsample	Width	Height	Scale
Pool1	2	2		Upsamp5	30	23	2
Pool2	2	2		Upsamp4	60	45	2
Pool3	2	2		Upsamp3	–	–	2
Pool4	2	2		Upsamp2	–	–	2
Pool5	2	2		Upsamp1	–	–	2

Table 6.2 Architecture and specification of the basic encoder (left)–decoder (right) network

Layer	Information	Layer	Layer	Layer	Information	Layer
Convolution	Output	Batch-Normalization	ReLU	Convolution	Output	Batch-Normalization
Conv1	64	Conv1-b	Relu1	Conv4-D	64	Conv4-D-b
Conv2	64	Conv2-b	Relu2	Conv3-D	64	Conv3-D-b
Conv3	64	Conv3-b	Relu3	Conv2-D	64	Conv2-D-b
Conv4	64	Conv4-b	Relu4	Conv1-D	64	Conv1-D-b
				ConvC-D	2	
Pooling	Stride	Kernel size		Upsample	Scale	
pool1	2	2		Upsamp4	2	
pool2	2	2		Upsamp3	2	
pool3	2	2		Upsamp2	2	
pool4	2	2		Upsamp1	2	

Table 6.3 Architecture and specification of the Bayesian-Basic encoder (left)–decoder (right) network

Layer	Information	Layer	Layer	Layer	Information	Layer
Convolution	Output	Batch-Normalization	ReLU	Convolution	Output	Batch-Normalization
Conv1	64	Conv1-b	Relu1	Conv4-D	64	Conv4-D-b
Conv2	64	Conv2-b	Relu2	Conv3-D	64	Conv3-D-b
Conv3	64	Conv3-b	Relu3	Conv2-D	64	Conv2-D-b
Conv4	64	Conv4-b	Relu4	Conv1-D	64	Conv1-D-b
				ConvC-D	2	
Pooling	Stride	Kernel size		Upsample		Scale
pool1	2	2		Upsamp4		2
pool2	2	2		Upsamp3		2
pool3	2	2		Upsamp2		2
pool4	2	2		Upsamp1		2
Drop-out		Ratio		Drop-out		Ratio
encdrop3		0.5		D-drop4		0.5
encdrop4		0.5		D-drop3		0.5

learning rates for the different variations of the networks. For the Original network the learning rate was set to 0.001, and for the Basic networks this value was set to 0.1. The learning rates are set optimally based on the work by Badrinarayanan et al. [2]. The direct effect of such a small learning rate is slow but more stable convergence of the network.

Figure 6.2 clearly illustrates this effect on these two network architectures during the training process. Table 6.4 summarizes the training parameters, which were set

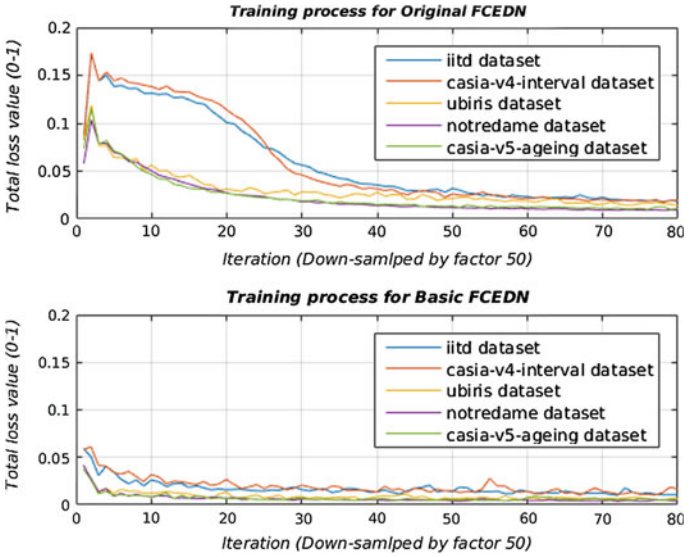

Fig. 6.2 Overall loss value in the first 4000 iterations of the training process for the Original (*up*), and the Basic (*down*) architectures on different datasets

Table 6.4 Solver parameters for the networks

Parameter	Original	Bayesian-Basic	Basic
Iterations	10,000	10,000	10,000
Momentum	0.9	0.9	0.9
Learning Rate	0.001	0.1	0.1

in the solver files. The setting criteria for these parameters is investigated in the work of Leon Bottou in this regard [3].

In our experiments the networks were trained per single image till the convergence point, where the overall loss for both segmentation classes decreased to less than 4%.

6.5 The First Segmentation Experiment Results

We have presented the segmentation results of the FCEDNs in Tables 6.5 and 6.6, as average segmentation scores per iris dataset, and per network respectively. As Table 6.5 demonstrates, the best results are obtained on the Notredame and the Casia5a datasets, and the worst ones on the Ubiris dataset. This is simply due to the difficulty level of these datasets. On the other hand, as it can be seen in Table 6.6, the Bayesian-Basic network outperforms the other two networks, with lower (mean) μnice1, μnice2 and higher μf1 scores (0.0316, 0.0571 and 0.8985 respectively),

Table 6.5 Average FCEDNs' segmentation scores per dataset

FCEDN	Dataset	nice1	nice2	f1
Original	iitd	0.0591	0.0659	0.8661
	notredame	0.0213	0.0424	0.8617
	casia4i	0.0561	0.0588	0.8826
	ubiris	0.0342	0.1249	0.7691
	casia5a	0.0160	0.0420	0.8951
Basic	iitd	0.0539	0.0594	0.8892
	notredame	0.0107	0.0269	0.9351
	casia4i	0.0448	0.0438	0.9072
	ubiris	0.0423	0.1517	0.7700
	casia5a	0.0086	0.0261	0.9510
Bayesian-Basic	iitd	0.0682	0.0701	0.8489
	notredame	0.0095	0.0282	0.9426
	casia4i	0.0391	0.0407	0.9192
	ubiris	0.0306	0.1116	0.8407
	casia5a	0.0105	0.0351	0.9413

Table 6.6 Average FCEDNs segmentation scores per network

FCEDN	μ nice1	μ nice2	μ f1
Bayesian-Basic	0.0316	0.0571	0.8985
Basic	0.0321	0.0616	0.8905
Original	0.0373	0.0668	0.8549

on overall datasets. This is directly due to the probabilistic technique used in this network, and the results clearly endorse the enhanced segmentation capability of the Basic network after applying this technique. The Basic network has comparatively moderate performance on the iris datasets with mean sores of 0.0321, 0.0616, and 0.8905 for μnice1, μnice2, and μf1 respectively. This is principally due to the simple structure of this network, which relays on the appearance information from shallow, fine layers to produce segmentations.

The Original network comes at the end, with mean sores of 0.0373, 0.0668 and 0.8549 for μnice1, μnice2 and μf1 respectively. This is meanly due to the comparatively deep structure of this network, which combines semantic information from deep, coarse layers with appearance information from shallow, fine layers to produce segmentations.

Figure 6.3 demonstrates the networks' best and worst performances samples for different datasets. More detailed performance analysis of the networks for iris segmentation per dataset is presented in Fig. 6.4, which provides further statistical information such as: min, max, median, quantiles and outliers in the form of Box-plots. The outliers are classified based on the following mechanism, where q_3 and q_1 are

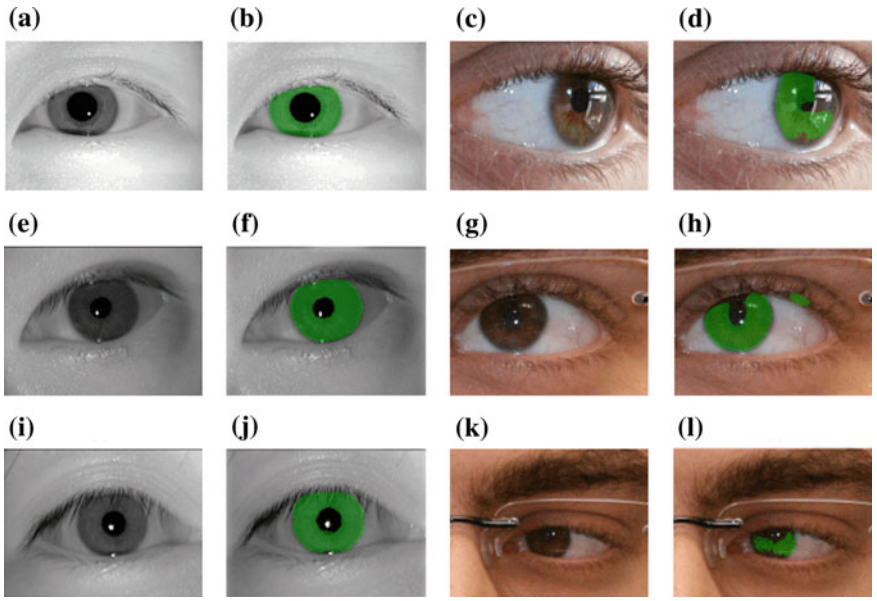

Fig. 6.3 The best (**b**) and the worst (**d**) performance samples of the Bayesian-Basic network on the notredame (**a**) and the ubiris (**c**) datasets' samples respectively. And the best (**f**) and the worst (**h**) performance samples of the Basic network on the casia5a (**e**) and the ubiris (**g**) datasets' samples respectively. And the best (**g**) and the worst (**l**) performance samples of the Original network on the casia5a (**i**) and the ubiris (**k**) datasets' samples respectively

the 25th and 75th percentiles of the scores, and w corresponds to approximately ± 2.7 of the standard deviation (σ): The scores are classified as outlier if they are greater than Z_1 or smaller than Z_2.

$$Z_1 = q_3 + w \times (q_3 - q_1) \tag{6.8}$$

$$Z_2 = q_3 - w \times (q_3 - q_1) \tag{6.9}$$

Furthermore, in order to assess how well the segmentation results generalize to the entire datasets, we trained and subsequently tested the networks on each dataset, applying the K-Fold cross-validation technique. For this purpose, we partitioned each dataset into five complementary subsets, and performed the training with four subsets, and validated the results on the remained subset. Likewise, five rounds of cross-validation were performed on each dataset separately without overlapping.

Figure 6.5 demonstrates the results for the cross-validation per segmentation score on the Notredame dataset. Table 6.7 demonstrates the average cross-validation results for all different datasets. As the results show, the average scores are quiet similar to those of the onefold experiments, and while most scores show around 2% difference, the maximum differences in the scores doe not exceed over 4%.

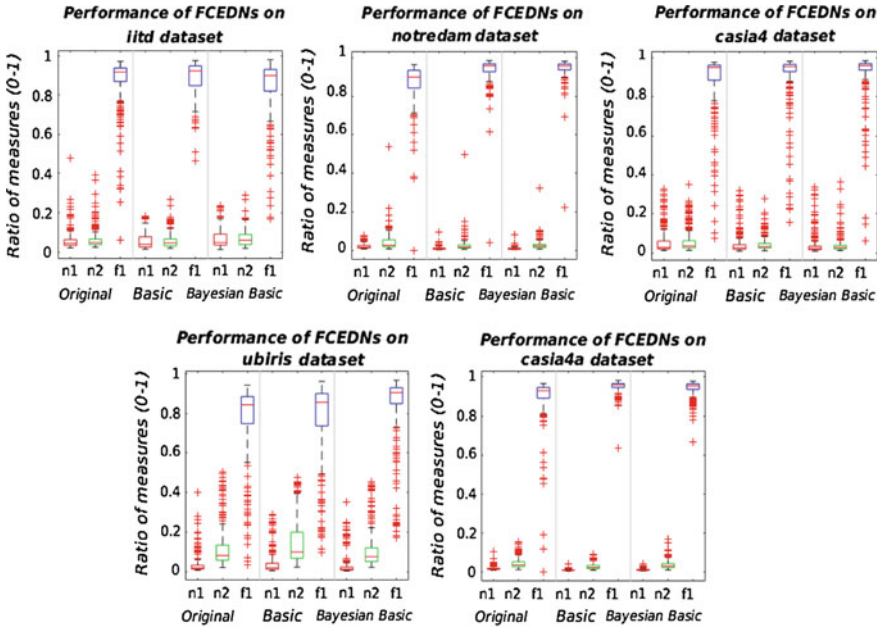

Fig. 6.4 Performance of the FCEDNs per dataset using segmentation errors: nice1 (n1), nice2 (n2) and f1 (f1)

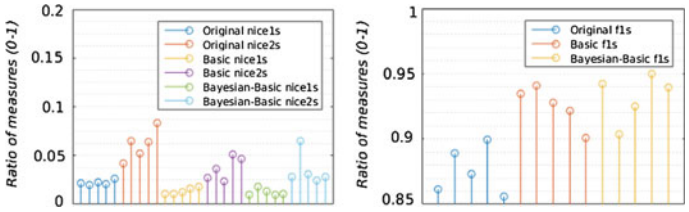

Fig. 6.5 Fivefold cross-validation results on the notredame dataset, demonstrating segmentation scores nic1, nice2 (*left*), and f1 (*right*) per round

Table 6.7 Average cross-validation results for all datasets

Network	Original			Basic			Bayesian-Basic		
Dataset	nice1	nice2	f1	nice1	nice2	f1	nice1	nice2	f1
Casia5a	0.0135	0.0392	0.9000	0.0197	0.0385	0.9250	0.0112	0.0332	0.9400
Casia4i	0.0415	0.0492	0.9175	0.0330	0.0382	0.9375	0.0412	0.0362	0.9250
Iitd	0.0365	0.0353	0.9400	0.0277	0.0322	0.9510	0.0292	0.0337	0.9500
Notredame	0.0220	0.0655	0.8775	0.0135	0.0387	0.9225	0.0132	0.0367	0.9300
Ubiris	0.0305	0.0898	0.7200	0.0262	0.0687	0.7900	0.0187	0.0675	0.8625

Table 6.8 Running time, per segmentation for FCEDNs

FCEDN	Original	Bayesian-Basic	Basic
Running time (s)	18.075	58.661	10.162

Furthermore, we measured the average running time, per segmentation, of different networks for a system with Intel-Xeon E5-1620 3.50 GHz cpu, and 32GiB memory. The results are presented in Table 6.8 respectively. Basically caffe is not optimized for the inlet processors. While based on the developers, using suitable GPUs and cuDNN, Caffe is considered as the fastest convent implementations available.[7]

6.6 The Second Segmentation Experiment Results

Next, in order to streamline the proper assessment of the capabilities of FCEDNs for iris segmentation, and to enable the comparative analysis of these networks' performance, a set of conventional iris segmentation methods (Convs) was considered to be run on the same datasets' testing subsets.

Osiris is an open-source iris recognition software, that includes an iris segmentation algorithm, which uses the Viterbi algorithm on the gradient map of anisotropic smoothed image for iris segmentation [33]. Caht (contrast-adjusted hough transform) [36], Wahet (weighted adaptive Hough and ellipsopolar transform) [47], and Ifpp (iterative Fourier-series push pull) [48] are further open-source iris segmentation algorithms used in this experiment, which are acquired from the Iris-Toolkit package provided by Wavelab at the University of Salzburg [37]. The performance of these algorithms was evaluated using the same segmentation scores used in the first experiment. The performance results of the conventional algorithms are represented in Tables 6.9 and 6.10, as average classification score per iris datasets and per algorithm respectively.

Generally, Osiris tends to underestimate the iris boundaries, as it tries to mask the obstructions out, leading to high precision but lower recall. However, Wahet and Caht lean to overestimate the iris boundaries, resulting in higher recall than precision. The reason for this is that these two algorithms do not utilize eyelid filters. Similarly, in the Ifpp algorithm, less pronounced boundaries are largely affected by noise or eyelids. Therefore, the less expressive boundaries are reconstructed from the more stable ones.

As it can be seem in Table 6.9 the algorithms perform better on less difficult datasets such as the Notredame and the Casia5a, while the worst results are obtained on the Ubiris dataset. In both cases the algorithms' performance results conform with the FCEDNs' results, which endorses the equity of the experimental scheme. As it

[7] Caffe Deep learning framework, see http://caffe.berkeleyvision.org/.

Table 6.9 Average convs' segmentation scores per dataset

Algorithm	Dataset	nice1	nice2	f1
Wahet	iitd	0.1377	0.1762	0.7337
	notredame	0.0248	0.0875	0.8619
	casia4i	0.0608	0.0842	0.8949
	ubiris	0.2743	0.4498	0.1977
	casia5a	0.0248	0.0836	0.8648
Caht	iitd	0.1138	0.1560	0.7767
	notredame	0.0361	0.1408	0.7941
	casia4i	0.1161	0.1470	0.7651
	ubiris	0.1226	0.4809	0.1048
	casia5a	0.0369	0.1514	0.7753
Ifpp	iitd	0.1142	0.1508	0.7965
	notredame	0.0294	0.1113	0.8359
	casia4i	0.1532	0.2372	0.6278
	ubiris	0.2379	0.3970	0.2899
	casia5a	0.0288	0.1123	0.8504
Osiris	iitd	0.0555	0.0757	0.8817
	notredame	0.0131	0.0231	0.9194
	casia4i	0.0565	0.0673	0.8862
	ubiris	0.1827	0.4095	0.2328
	casia5a	0.0181	0.0331	0.8917

Table 6.10 Average convs' segmentation scores per algorithm

Algorithm	μ nice1	μ nice2	μ f1
Osiris	0.0652	0.1217	0.7624
Caht	0.0851	0.2152	0.6432
Wahet	0.1045	0.1763	0.7106
Ifpp	0.1127	0.2017	0.6801

can be seen in Table 6.10 the Osiris algorithm outperforms the other threes, with lower (mean) μnice1, μnice2 and higher μf1 scores (0.0652, 0.1217 and 0.7624 respectively). Figure 6.6 provides further statistical information such as: min, max, median, quantiles, and outliers, about the segmentation performances of these algorithms per dataset in the form of Box-plots. Furthermore, we measured the average running time, per segmentation, of the conventional algorithms for the system specified previously. The results are presented in Table 6.11 respectively.

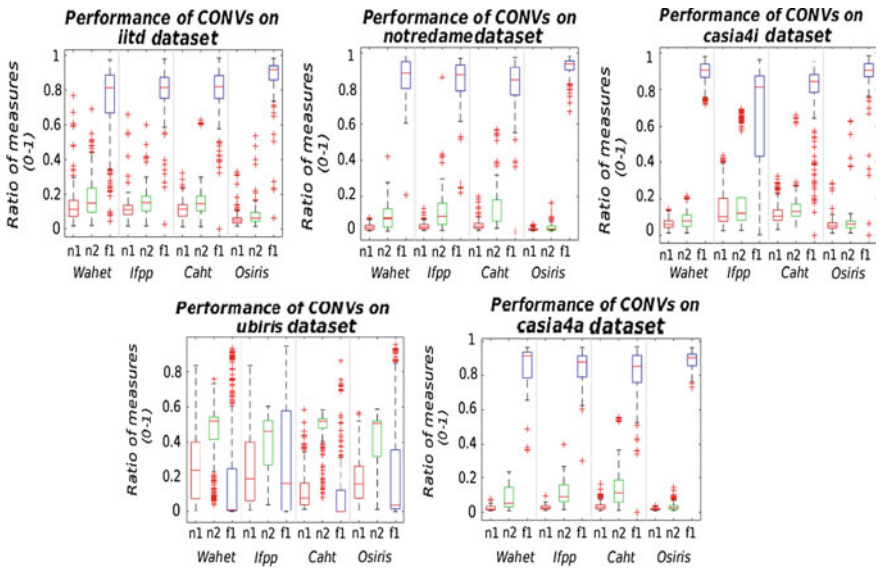

Fig. 6.6 Performance of the conventional algorithms per dataset using segmentation errors: nice1 (n1), nice2 (n2) and f1 (f1)

Table 6.11 Running time, per segmentation for convs

Algorithm	Osiris	Wahet	Caht	Ifpp
Running time (s)	0.583	0.602	6.730	0.470

6.7 Analysis and Discussion

Simple statistical comparison of the segmentation results of the FCEDNs with the conventional algorithms' results demonstrates the superiority of the FCEDNs for iris segmentation over the other conventional algorithms. As it can be seen in Tables 6.6 and 6.10, even the worst FCEDNs' performance result, which is shown by the Original network, scoring: 0.0373, 0.0668 and 0.8549 for nice1, nice2 and f1 respectively, is better then the best conventional algorithms' result, which is obtained by the Osiris algorithm scoring: 0.0652, 0.1217 and 0.7624 for nice1, nice2, and f1 respectively. Yet if we consider the best FCEDNs' result, which is obtained by the Bayesian-Basic network (0.0316, 0.0571, and 0.8985 for nice1, nice2 and f1 respectively), the prominence of the proposed FCEDNs over the conventional algorithms would be consolidated by power of two or three.

The greatest supremacy of the proposed FCEDNs is revealed when analysing the segmentation results per dataset in Tables 6.5 and 6.9. As it can be seen in the tables, the worst segmentation scores for almost all conventional algorithms and FCEDs are obtained on the Ubiris dataset, which deliberately contains samples of off-angle iris

Table 6.12 Average segmentation scores of all methods on the Ubiris dataset

Method	nice1	nice2	f1
Bayesian-Basic	0.0306	0.1116	0.8407
Original	0.0342	0.1249	0.7691
Basic	0.0423	0.1517	0.7700
Osiris	0.1827	0.4095	0.2328
Caht	0.1226	0.4809	0.1048
Wahet	0.2743	0.4498	0.1977
Ifpp	0.2379	0.3970	0.2899

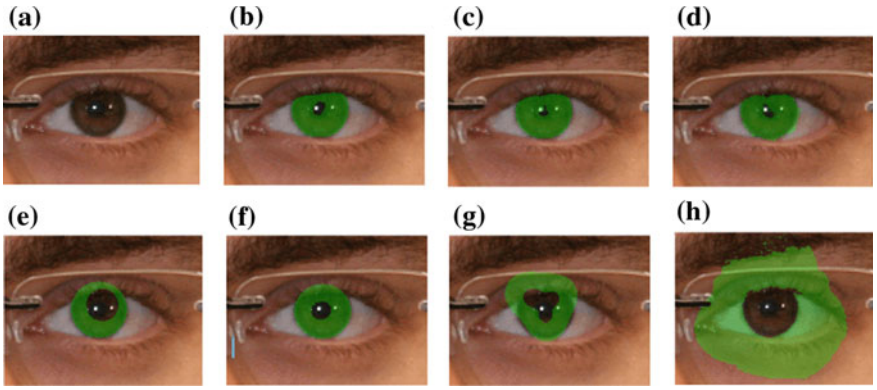

Fig. 6.7 A sample iris image with glasses (**a**) from the ubiris dataset versus the output segmentation masks of: Bayesian-Basic (**b**), Basic (**c**), Original (**d**), Caht (**e**), Wahet (**f**), Ifpp (**g**), and Osiris (**h**)

images recorded from various distances with different types of occlusions, including glasses. While most conventional algorithms such as Wahet, Caht and Ifpp even fail to satisfy the minimum segmentation scores, all FCEDNs demonstrate robust segmentation capabilities on such a difficult and divergent iris dataset.

This can be easily interpreted from Table 6.12, which summarizes the segmentation results of the conventional algorithms along with the FCEDNs on the Ubiris dataset. A simple visual comparison of the Box-plots for the Ubiris dataset in Figs. 6.4 and 6.6 demonstrates this fact clearly also. Figure 6.7 displays a sample iris image with glasses from the Ubiris dataset, along with the corresponding output masks of all segmentation methods (FCEDNs and Convs).

6.8 Conclusion

Accurate segmentation of the iris region from the rest of image plays a vital role in efficient performance of iris recognition systems, and success of the total system is considered to be directly related to the precision of this stage. In this work we have

presented the application of deep learning techniques and FCEDNs for iris segmentation. To this extent, we applied three types of networks for iris segmentation. The performance of the networks was tested and evaluated on five different datasets. The evaluation was carried out using three popular segmentation error scores. Furthermore, in order to streamline proper assessment of the performance of the networks, we presented statistical analysis and the performance evaluation of four well-known conventional iris segmentation algorithms on the same datasets, and compared the results against those obtained from the networks. Results demonstrate the superiority of the networks for iris segmentation over all other algorithms. Yet the greatest supremacy of the proposed networks unveils when dealing with difficult iris images such as off-angle images recorded from various distances with different types of occlusions including glasses. In future work we plan to perform more application specific analysis on these types of networks, and at the same time carry out further research to optimize their design and architecture, and improve their performance for different segmentation tasks.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 700259.

References

1. U. Andreas, W. Peter, Multi-stage visible wavelength and near infrared iris segmentation framework, in *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR'12)*, LNCS (Aveiro, Portugal, 2012), pp. 1–10
2. V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation (2015), arXiv:1511.00561
3. L. Bottou, Stochastic gradient descent tricks, in *Neural Networks: Tricks of the Trade* (Springer, 2012), pp. 421–436
4. R.P. Broussard, L.R. Kennell, D.L. Soldan, R.W. Ives, Using artificial neural networks and feature saliency techniques for improved iris segmentation, in *International Joint Conference on Neural Networks, 2007. IJCNN 2007* (IEEE, 2007), pp. 1283–1288
5. Y. Chen, M. Adjouadi, A. Barreto, N. Rishe, J. Andrian, A computational efficient iris extraction approach in unconstrained environments, in *IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems, 2009. BTAS'09* (IEEE, 2009), pp. 1–7
6. J. Daugman, Recognizing people by their iris patterns. Inf. Secur. Tech. Rep. **3**(1), 33–39 (1998)
7. J. Daugman, How iris recognition works. Int. Conf. Image Process. **1**, I-33–I-36 (2002)
8. J.G. Daugman, High confidence visual recognition of persons by a test of statistical independence. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1148–1161 (1993)
9. J. Denker, Y. Lecun, Transforming neural-net output levels to probability distributions, in *Proceedings of the 3rd International Conference on Neural Information Processing Systems* (Morgan Kaufmann Publishers Inc, 1990), pp. 853–859
10. V. Dorairaj, N.A. Schmid, G. Fahmy, Performance evaluation of non-ideal iris based recognition system implementing global ica encoding, in *IEEE International Conference on Image Processing 2005*, vol. 3 (IEEE, 2005), pp. III–285
11. Y. Gal, Z. Ghahramani, Bayesian convolutional neural networks with bernoulli approximate variational inference (2015), arXiv:1506.02158
12. Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: representing model uncertainty in deep learning (2015), arXiv:1506.02142

13. A. Gangwar, A. Joshi, Deepirisnet: deep iris representation with applications in iris recognition and cross-sensor iris recognition, in *2016 IEEE International Conference on Image Processing (ICIP)* (IEEE, 2016), pp. 2301–2305
14. S. Gupta, R. Girshick, P. Arbeláez, J. Malik, Learning rich features from rgb-d images for object detection and segmentation, in *European Conference on Computer Vision* (Springer, 2014), pp. 345–360
15. B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Simultaneous detection and segmentation. In *European Conference on Computer Vision* (Springer, 2014), pp. 297–312
16. H. Hofbauer, F. Alonso-Fernandez, P. Wild, J. Bigun, A. Uhl, A ground truth for iris segmentation, in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR'14)* (Stockholm, Sweden, 2014), 6pp
17. H. Hofbauer, F. Alonso-Fernandez, J. Bigun, A. Uhl, Experimental analysis regarding the influence of iris segmentation on the recognition rate. IET Biom. **5**(3), 200–211 (2016)
18. J. Huang, Y. Wang, T. Tan, J. Cui, A new iris segmentation method for recognition, in *(ICPR 2004). Proceedings of the 17th International Conference on Pattern Recognition, 2004*, vol. 3 (IEEE, 2004), pp. 554–557
19. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015), arXiv:1502.03167
20. A. Kendall, V. Badrinarayanan, R. Cipolla, Bayesian segnet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding (2015), arXiv:1511.02680
21. W. Kong, D. Zhang, Accurate iris segmentation based on novel reflection and eyelash detection model, in *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2001* (IEEE, 2001), pp. 263–266
22. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105
23. R.D. Labati, F. Scotti, Noisy iris segmentation with boundary regularization and reflections removal. Image Vis. Comput. **28**(2), 270–277 (2010)
24. Y. Le Cun, D. Touresky, G. Hinton, T. Sejnowski, A theoretical framework for backpropagation, in *The Connectionist Models Summer School*, vol. 1 (1988), pp. 21–28
25. Y. Le Cun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
26. Y.-H. Li, M. Savvides, An automatic iris occlusion estimation method based on high-dimensional density estimation. IEEE Trans. Pattern Anal. Mach. Intell. **35**(4), 784–796 (2013)
27. N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, T. Tan, Accurate iris segmentation in non-cooperative environments using fully convolutional networks, in *2016 International Conference on Biometrics (ICB)* (IEEE, 2016), pp. 1–8
28. N. Liu, M. Zhang, H. Li, Z. Sun, T. Tan, Deepiris: learning pairwise filter bank for heterogeneous iris verification. Pattern Recogn. Lett. **82**, 154–161 (2016)
29. X. Liu, K.W. Bowyer, P.J. Flynn, Experiments with an improved iris segmentation algorithm, in *Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AutoID'05)* (IEEE, 2005), pp. 118–123
30. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3431–3440
31. L. Ma, T. Tan, Y. Wang, D. Zhang, Personal identification based on iris texture analysis. IEEE Trans. Pattern Anal. Mach. Intell. **25**(12), 1519–1533 (2003)
32. H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1520–1528
33. D. Petrovska, A. Mayoue, Description and documentation of the biosecure software library, in *Project No IST-2002-507634-BioSecure, Deliverable* (2007)
34. A. Radman, K. Jumari, N. Zainal, Iris segmentation in visible wavelength images using circular gabor filters and optimization. Arab. J. Sci. Eng. **39**(4), 3039–3049 (2014)

35. K.B. Raja, R. Raghavendra, V.K. Vemuri, C. Busch, Smartphone based visible iris recognition using deep sparse filtering. Pattern Recogn. Lett. **57**, 33–42 (2015)
36. C. Rathgeb, A. Uhl, P. Wild, *Iris Recognition: From Segmentation to Template Security*, vol. 59, Advances in Information Security (Springer, Berlin, 2013)
37. C. Rathgeb, A. Uhl, P. Wild, H. Hofbauer, Design decisions for an iris recognition sdk, in *Handbook of Iris Recognition*, 2nd edn., Advances in Computer Vision and Pattern Recognition, ed. by K. Bowyer, M.J. Burge (Springer, Berlin, 2016)
38. C.J.V. Rijsbergen, *Information Retrieval*, 2nd edn. (Butterworth-Heinemann, Newton, 1979)
39. T. Rongnian, W. Shaojie, Improving iris segmentation performance via borders recognition, in *2011 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, vol. 2 (IEEE, 2011), pp. 580–583
40. O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2015), pp. 234–241
41. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2014), arXiv:1409.1556
42. N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
43. C.-W. Tan, A. Kumar, Unified framework for automated iris segmentation using distantly acquired face images. IEEE Trans. Image Process. **21**(9), 4068–4079 (2012)
44. C.-W. Tan, A. Kumar, Towards online iris and periocular recognition under relaxed imaging constraints. IEEE Trans. Image Process. **22**(10), 3751–3765 (2013)
45. C.-L. Tisse, L. Martin, L. Torres, M. Robert et al., Person identification technique using human iris recognition, in *Proceedings of Vision Interface* (2002), pp. 294–299
46. P.V. Tran, A fully convolutional neural network for cardiac segmentation in short-axis mri, in *CoRR* (2016), arXiv:1604.00494
47. A. Uhl, P. Wild, Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation, in *Proceedings of the 5th IAPR/IEEE International Conference on Biometrics (ICB'12)* (New Delhi, India, 2012), pp. 1–8
48. P. Wild, H. Hofbauer, J. Ferryman, A. Uhl, Segmentation-level fusion for iris recognition, in *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'15)* (Darmstadt, Germany, 2015), p. 12
49. R.P. Wildes, Iris recognition: an emerging biometric technology. Proc. IEEE **85**(9), 1348–1363 (1997)
50. Z. Zhao, K. Ajay, An accurate iris segmentation framework under relaxed imaging constraints using total variation model, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 3828–3836
51. J. Zuo, N. D. Kalka, N.A. Schmid, A robust iris segmentation procedure for unconstrained subject presentation, in *2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference* (IEEE, 2006), pp. 1–6