Seg-Edge Bilateral Constraint Network for Iris Segmentation

Junxing Hu[1,2], Hui Zhang[1*], Lihu Xiao[1], Jing Liu[1], Xingguang Li[1], Zhaofeng he[1], Ling Li[1,2]
1 Beijing IrisKing Co., Ltd. 2 University of Chinese Academy of Sciences
*Corresponding author Hui Zhang, zhanghui@irisking.com

Abstract

Iris semantic segmentation in less-constrained scenarios is the basis of iris recognition. We propose an end-to-end trainable model for iris segmentation, namely Seg-Edge bilateral constraint network (SEN). The SEN uses the edge map and the coarse segmentation to constrain and optimize mutually to produce accurate iris segmentation results. The iris edge map generated from low level convolutional layers passes detailed edge information to iris segmentation, and the iris region generated by high level semantic segmentation constrains the edge filtering scope which makes the edge aware focusing on interesting objects. Moreover, we propose pruning filters and corresponding feature maps that are identified as useless by l_1-norm, which results in a lightweight iris segmentation network while keeping the performance almost intact or even better. Experimental results suggest that the proposed method outperforms the state-of-the-art iris segmentation methods.

1. Introduction

Iris recognition is a reliable identification technique due to the stability and uniqueness of the iris in biometrics. It is widely used in banks, mines and other places with high security levels. There are many disadvantages such as long-distance, on-the-move, occlusion and reflection, which seriously impact the quality of acquired iris images. As a crucial part of iris recognition, an excellent iris segmentation simplifies processes and improves the recognition accuracy.

Traditional iris segmentation methods usually consist of several main steps, including preprocessing, denoising, boundary detection, and post-processing. Daugman [7] proposes using an integro-differential operator to detect eyelids and locate iris boundaries. Wildes [26] presents exploiting a circular Hough transform to localize iris boundaries in iris images. Tisse et al. [25] implement an iris segmentation method by combining integro-differential operators and the gradient decomposed Hough transform. Tan et al. [24] propose a clustering based method to identify non-iris regions with an integrodifferential constellation. Tan [23] develop an approach exploiting random walker algorithm with a sequence of post-processing. Zhao [27] propose a total variation formulation using l_1-norm regularization for the iris localization. These methods need manually designed rules or features to solve the occlusion, reflection, blur and other matters. Several steps of pre- and post-processing also increase the complexity of segmentation.

978-1-7281-3640-0/19/$31.00 ©2019 IEEE

Recent years, methods based on neural network are proposed for pixel level iris segmentation. Proença [17] exploits the neural network with one hidden layer for iris pixel-level classification. Tan and Kumar [22] use a typical 3-layer feed forward neural network. These two methods use manually extracted features and pre- and post-processing operations. Liu et al. [14] propose a multi-scale fully convolutional network without pre- and post-processing. They use the fusion layer to sum the output of each layer after up sampling. Although it has been shown that the deep learning based methods perform well, their large number of parameters and massive calculation limit application. There is still a lot of work to do about network structures and model pruning for the iris segmentation.

Motivated by prior studies, we propose an end-to-end model for iris segmentation without pre- and post-processing, namely Seg-Edge bilateral constraint network (SEN). Average segmentation errors obtained by SEN improve over the state-of-the-arts by 2.22% and 22.03% on UBIRIS.v2 [18] and CASIA.v4-distance [1] dataset respectively. The main contributions of this paper include:

1) By using the bilateral constraint domain transform, the SEN not only uses the edge map to improve the segmentation result as [9], but also constrains the edge aware inside the segmented iris region. It makes the network focusing on boundary between iris and non-iris (sclera, pupil, eyelid, eyelash, spot and other occlusion), while ignoring edge of other facial parts (like eyebrow, canthus, and glasses frame).

2) The edge map is generated from all the low level convolutional layers (from conv1 to conv5), which provide rich edge features at different scales or sharpness degrees.

3) We prune the model by l_1-norm method, resulting in

the lightweight network with better performance.

2. Related work

Semantic segmentation With the development of deep learning methods, semantic segmentation has attracted more and more attention. The appearance of Fully Convolutional Networks (FCN) [16] led to a rapid increase in the number of end-to-end semantic segmentation networks. The FCN model transforms all of fully connected layers to convolutional layers which allows the input image of any size. Built upon FCN, U-Net [19] adopts an encoder-decoder structure which consists of a contracting path and a symmetric expanding path. Similarly, SegNet [2] is also built on an encoder-decoder structure, but it use max-pooling indices to enhance location information. The DeepLab model proposed in [5] uses atrous convolution and fully connected Conditional Random Field (CRF) to avoid the reduction of the spatial resolution of feature maps. The atrous convolution effectively enlarges the field-of-view of filters to incorporate larger context without increasing the number of parameters and the amount of computation. And the CRF greatly improves the localization performance. Based on [5], DeepLab v2 [6] uses Atrous Spatial Pyramid Pooling (ASPP) with multiple sampling rates to robustly segment objects at multiple scales. This method gets the excellent result at the PASCAL VOC-2012 semantic image segmentation task [8].

Performance optimization To further enhance the performance of the model like VGG, some methods exploit information from intermediate convolutional layers to improve the segmentation result. In [14], MFCN which consists of 31 convolutional layers fuses 6 convolutional layers from shallow to deep to capture both the coarse and fine details. Moreover, the domain transform [9] is used as an edge-preserving filter and is exploited to improve the semantic segmentation result by aligning segmentation results with object boundaries [4].

Model pruning For model compression, a lot of methods have been proposed. Le et al. [12] propose Optimal Brain Damage to remove unimportant weights from a network. They use second-derivative information to make a tradeoff between network complexity and training set error. Han et al. [10] remove the connection whose weight is lower than a threshold after an initial training phase, and converts a dense, fully-connected layer to a sparse layer. Different from methods above, Li et al. [13] present removing filters together with their connecting feature maps to reduce the computation cost. Under the guidance of l_1-norm, this approach does not produce sparse connectivity patterns and not need the additional regularization.

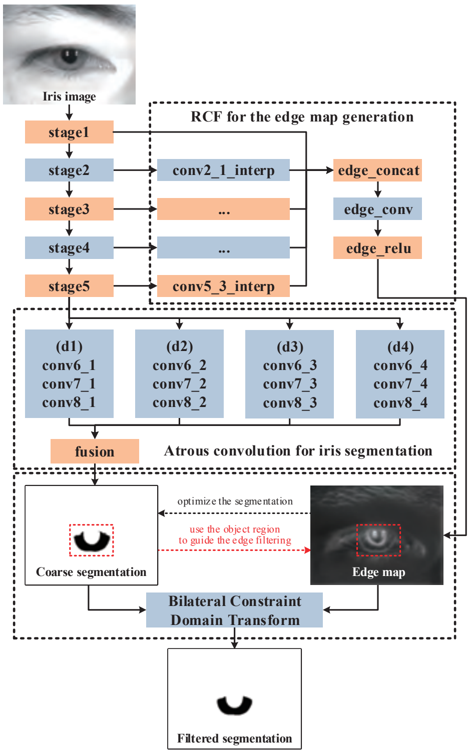

Iris image
RCF for the edge map generation
stage1
stage2
stage3
stage4
stage5
conv2_1_interp
...
...
conv5_3_interp
edge_concat
edge_conv
edge_relu
(d1) conv6_1 conv7_1 conv8_1
(d2) conv6_2 conv7_2 conv8_2
(d3) conv6_3 conv7_3 conv8_3
(d4) conv6_4 conv7_4 conv8_4
fusion
Atrous convolution for iris segmentation
optimize the segmentation
use the object region to guide the edge filtering
Coarse segmentation
Edge map
Bilateral Constraint Domain Transform
Filtered segmentation

Figure 1. The architecture of the proposed SEN.

3. Proposed methods

3.1. Seg-Edge bilateral constraint network

The SEN is mainly composed of three components, as shown in Figure 1. The first is the backbone structure which is an improved version of VGG-16 [20]. The second generates the iris edge map using rich convolutional features (RCF) The third is the bilateral constraint domain transform which generates the final result.

3.1.1 Backbone structure

We exploit the powerful FCN model DeepLab v2 [6] as the backbone structure of the network, using the VGG-16 with full convolutional layers as base model. The output of this component is the coarse iris segmentation result which is one input of the bilateral constraint domain transform.

Atrous convolution for iris segmentation The boundary between iris and non-iris is not with same sharpness. The edge between iris and sclera is not as sharp as the eyelash or eyebrow. The Atrous convolution with different dilation rates proposed in DeepLab v2 is suitable for characterizing iris boundary. It can enlarge the field-of-view of kernels without increasing parameters of the model. It uses dilation d to enlarge the kernel with the size of $f \times f$ to $f_e \times f_e$ as: $f_e = f + (f-1)(d-1)$.

The consecutive filter values are filled with $d-1$ ze-

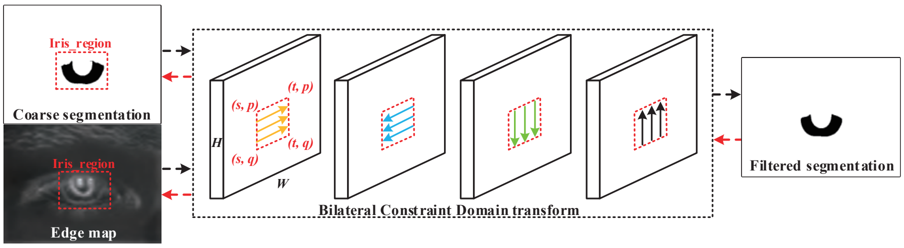


Figure 2. Illustration of one iteration of the bilateral constraint domain transform. The black dashed arrow indicates the forward propagation, while the red dashed arrow indicates the back propagation. The red dashed box indicates the iris region, in which the edge map and the coarse segmentation are recursively filtered across rows and columns.

ros. The Atrous Spatial Pyramid Pooling (ASPP) proposed by DeepLab v2 is used in our task. The ASPP uses 'conv6_1', 'conv6_2', 'conv6_3' and 'conv6_4' as four parallel branches with multiple dilation $d1, d2, d3, d4$ and fuses them to generate the iris segmentation as shown in Figure 1, $d1 = 6, d2 = 12, d3 = 18, d4 = 24$ in our experiments.

The DeepLab v2 conducts the fully connected Conditional Random Fields (CRFs) to improve segmentation results. However, the probability estimate of CRFs show little advances in detailed and complexity edge areas, such as the eyelash and reflected light spots. The CRFs is time consuming which requires more than 0.2 sec for image of 480*360 pixels. Therefore, the CRFs is not used in our models.

3.1.2 Rich features for the edge map generation

Another input of the bilateral constraint domain transform is iris edge map generated from low level convolutional layers, as shown in Figure 1. In the edge detection task, using richer convolutional features is more effective [15]. Low level convolutional layers (conv1 to conv5) contains richer edge features at different scales and sharpness degrees. We exploit all of them to predict the edge map following [4]. For medial convolutional layers, the method rescales their outputs to the original size. A concat layer concatenates them to one output. A convolutional layer with 1×1 kernel is used to produce the iris edge prediction and ReLU makes the prediction in range of zero to infinity.

3.1.3 Bilateral constraint domain transform

To further optimize the accuracy of iris segmentation, we propose the bilateral constraint domain transform (BDT). As shown in Figure 1, the BDT uses edge map and coarse segmentation to constrain each other. One side, the iris edge map generated from intermediate convolutional layers can pass more detailed edge information to segmentation results, which follows the basic idea of the domain transform (DT) [4]. On the other side, iris region produced by coarse segmentation can constrains edge filtering scope, which allows the edge aware focusing on the interesting object parts.

Constraint forward propagation As illustrated in Figure 2, the BDT recursively filters inputs across rows and columns through K iterations. During the forward propagation, the filtering is performed along four directions (left to right, right to left, top to bottom, and bottom to top) in sequence as proposed in DT [4]. We limit these operations within the iris area to further reduce the amount of computation, and modify the recursive formulation of the DT due to its speed and efficiency [9]. For 2-D inputs of height H and width W, the output $y_{i,j}$ is computed as:

$$\begin{aligned} y_{i,j} &= (1 - w_{i,j})x_{i,j} + w_{i,j}y_{i,j-1} \\ i &= p, ..., q \quad (1 \leq p \leq q \leq H), \\ j &= s, ..., t \quad (2 \leq s \leq t \leq W) \end{aligned} \tag{1}$$

where $x_{i,j}$ is the pixel value at $(i, j)$of the coarse iris segmentation, p and q are the lower and upper bound of the iris region in the vertical direction, s and t are the bound in the horizontal direction. The weight $w_{i,j} \in [0, 1]$ is a *feedback coefficient* [21] which is related to the iris edge map.

Constraint back propagation During the backward propagation, the segmentation errors at the output $y_{i,j}$ are back propagated through the DT onto its two inputs. To avoid the interference from the non-iris area and constrain the edge aware inside the object region, we add the same limitation to both the coarse segmentation and the edge map. We calculate the derivative as follows:

$$\begin{aligned} \frac{\partial L}{\partial x_{i,j}} &\leftarrow (1 - w_{i,j})\frac{\partial L}{\partial y_{i,j}} \\ \frac{\partial L}{\partial y_{i,j-1}} &\leftarrow \frac{\partial L}{\partial y_{i,j-1}} + w_{i,j}\frac{\partial L}{\partial y_{i,j}} \\ \frac{\partial L}{\partial w_{i,j}} &\leftarrow \frac{\partial L}{\partial w_{i,j}} + (y_{i,j-1} - x_{i,j})\frac{\partial L}{\partial y_{i,j}} \\ \frac{\partial L}{\partial g_{i,j}} &\leftarrow -\frac{\sqrt{2}}{\sigma_k}\frac{\sigma_s}{\sigma_r}w_{i,j}\frac{\partial L}{\partial w_{i,j}} \\ & i = q, ..., p \quad (1 \leq p \leq q \leq H), \\ & j = t, ..., s \quad (2 \leq s \leq t \leq W) \\ & k = 1, ..., K \end{aligned} \tag{2}$$

where $g_{i,j}$ is the pixel value at $(i,j)$of the iris edge map, σ_s and σ_r are the standard deviation of the filter kernel over the inputs spatial domain and the reference edge maps range. k represents the kth iteration, $\sigma_k = \sqrt{3}\sigma_s 2^{K-k}/\sqrt{4^K - 1}$. Note that the recursions of i, j are opposite to the forward propagation. K is used the same as [4], and the value of σ_s and σ_r are related to the dataset. Since the location of the generated iris region may not be accurate enough, we expand five pixels per side of the region during the filtering.

3.2. Iris segmentation model pruning

Pruning filters in a single layer It is important to determine which filters to prune. So we first prune the single layer to observe its sensitivity to pruning. We exploit the procedure of pruning [13] as: 1) For the ith convolutional layer, let $f_{i,j}$ denotes jth filter, calculate its l_1-norm $v_j = \sum |f_{i,j}|$; 2) Sort filters by v_j; 3) Prune m filters with the smallest v_j and their connecting feature maps; 4) Create a new model with the remaining filter weights and retrain it.

We compare the pruned models and the retrained pruned models. Some layers are sensitive to pruning as we can not recover the accuracy after pruning them.

Pruning filters across multiple layers For layers which are sensitive to pruning, we prune fewer or no filters of them. We adopt the *one-shot* pruning method which prunes filters across multiple layers at once and retrains the model [13]. After retraining, the number of parameters and floating point operations (FLOP) are reduced while keeping the performance almost intact or even better.

4. Experiments and Results

4.1. Datasets

We evaluate algorithms on two datasets. One is a subset of UBIRIS.v2 [18], which contains 500(train) and 445(test) pixel-level labeled images. One is a subset of CASIA.v4-distance [1].400 iris images from the first 40 subjects are manually labeled [14]. The first 300 images are used for training and the other for testing, as the same setting as [14]. Experiments on two datasets are standalone. We resize all images to 480 × 360 pixels for subsequent experiments. We generate images in terms of image resolution, intensity, blur, and add shadow effects in a random direction as [3] to enlarge the database, resulting in 5000 training images for UBIRIS.v2 and 3000 for CASIA.v4-distance.

4.2. Training pipeline

We fine-tune models from DeepLab v2 models [6] which are trained for PASCAL VOC-2012 [8]. We conduct four experiments: 1) Train the DeepLab v2 models for iris images; 2) Prune models from step 1 and retrain them to reduce parameters and FLOP; 3) Add the standard domain transform to models in step 2 and fine-tune (note as SEN_U); 4) Add the bilateral constrain domain transform to models in step 2 and fine-tune (note as SEN_B).

Table 1. The ASE of DeepLab v2 models. 'CRFs' indicates the fully connected CRFs method.

Method	UBIRIS.v2(%)		CASIA.v4(%)	
	no CRFs	CRFs	no CRFs	CRFs
DeepLab v2(VGG-16)	0.926	**0.907**	**0.497**	0.524
DeepLab v2(ResNet-101)	**0.888**	0.940	**0.486**	0.560

Table 2. For three pruning strategies, the best ASE of retraining process are reported. 'Params' and 'FLOP' indicate the reduced percentage of parameters and FLOP for each pruning strategies.

Method	UBIRIS.v2(%)			CASIA.v4(%)		
	ASE	Params	FLOP	ASE	Params	FLOP
Prune_1	**0.926**	2.79	2.75	**0.489**	8.19	9.11
Prune_2	0.979	58.28	46.15	0.494	58.28	46.15
Prune_3	0.928	33.31	26.38	0.494	33.31	26.38

4.3. Experimental results

The accuracies of models are measured by the average segmentation error (ASE):

$$ASE = \frac{1}{N \times H \times W} \sum_{i,j \in (H,W)} G(i,j) \oplus M(i,j) \quad (3)$$

where N is the total number of the test images, H and W are height and width, G and M are the ground truth mask and the generated iris mask respectively. $\oplus$ represents an exclusive OR operation to compute the segmentation error. We first evaluate DeepLab v2 models on two datasets, as shown in Table 1. Results are close or outperform the result of [14]. It demonstrates the effectiveness of the DeepLab v2 and it makes sense to use this architecture. We also use ResNet-101 [11] based model as baseline which shows advantages on the UBIRIS.v2 dataset. It is reasonable to not use CRFs which performs bad in most cases.

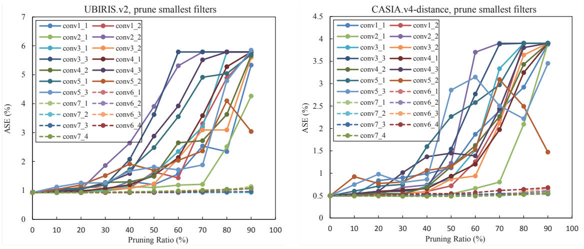
4.3.1 Model pruning results

We use three different pruning strategies, and the corresponding results are shown in the Table 2.

(1) After sorting filters by l_1-norm, we prune the smallest filters of each convolutional layer independently with different pruning ratios in the range of 10% to 90%, and evaluate the ASE of pruned model, seeing Figure 3. Pruning some single layers may even improve the performance. We first prune these filters and retrain. The pruned layers and corresponding pruning ratios of models are:

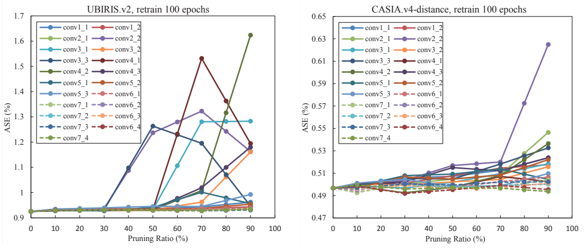
(i) For UBIRIS.v2, we prune 10%, 30%, 70% of conv1_1, conv7_1, conv7_2, respectively.

(ii) For CASIA.v4-distance, we prune 30%, 10%, 40%, 60% of conv2_1, conv3_1, conv6_3, conv7_4, respectively.



(a) UBIRIS.v2 (b) CASIA.v4-distance

Figure 3. The ASE after pruning filters with the smallest absolute weights sum for each single layer of models on datasets.



(a) UBIRIS.v2 (b) CASIA.v4-distance

Figure 4. The ASE after retraining the pruned models on datasets.

Table 3. The ASE of the SEN. 'three' indicates the three layers, 'rich' indicates all of the layers used by models. SEN_U indicates the SEN using the DT, SEN_B using the BDT.

Method	UBIRIS.v2(%)		CASIA.v4(%)	
	three	rich	three	rich
SEN_U	0.923	**0.883**	0.491	0.480
SEN_B	0.925	0.924	0.471	**0.462**

(2) We retrain all pruned models separately, and ASEs are listed in Figure 4. Some layers are sensitive to pruning, as we can not recover their accuracies even using small pruning ratios, such as conv2_2 layer for both datasets. We empirically prune 50% filters of each layer from conv5_1 to the end of the model. Table 2 demonstrates that the ASE of the pruned model for the CASIA.v4-distance can be restored. But we can not restore the ASE for the UBIRIS.v2 because 5th layer are sensitive to pruning for this dataset (seeing Figure 4).

(3) Based on the result of (2), we prune 50% filters of each layer from conv6_1 to conv7_4.

As a trade-off between the accuracy and the number of parameters, we exploit the pruning strategy (1) for the U-BIRIS.v2 and (2) for the CASIA.v4-distance.

Table 4. Comparisons of the ASE with other methods.

Method	UBIRIS.v2 (%)	CASIA.v4 (%)
Proposed SEN	**0.88**	**0.46**
MFCNs [14]	0.90	0.59
RTV-L^1 [27]	1.21	0.68
Tan et al. [24]	1.31	-
Tan and Kumar [23]	1.72	0.81
Proença [17]	1.87	-
Tan and Kumar [22]	1.90	1.13

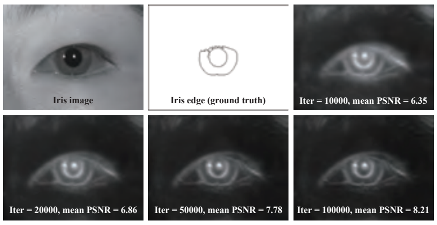

Iris image
Iris edge (ground truth)
Iter = 10000, mean PSNR = 6.35
Iter = 20000, mean PSNR = 6.86
Iter = 50000, mean PSNR = 7.78
Iter = 100000, mean PSNR = 8.21

Figure 5. PSNRs of the iris edge map of different iterations.

4.3.2 Bilateral constraint segmentation results

To verify the effectiveness of the domain transform, we first generate the iris edge map from ground truthR) to measure the , and use the Peak Signal-to-Noise Ratio (PSN generated edge map. As in Figure 5, as the number of iterations increases during training, the iris edge gets clearer.

As shown in Table 3, the ASE of 'rich' (generate edge maps from all five low level layers) are better than 'three' (generate edge maps from conv2_2, conv3_3 and conv4_3) for both datasets which shows the superiority of the rich convolutional features. Results of BDT and the DT are compared. The best result on CASIA.v4-distance is obtained by using the SEN_B, while UBIRIS.v2 by using SEN_U. It is caused by the difference of edge information of two datasets. Because the UBIRIS.v2 benchmark is acquired in VL illumination and the CASIA.v4-distance in NIR illumination, the iris texture and the edge of the former are more weaker than the latter. Therefore, the SEN_B which is designed to further optimize the iris edge map may be worse than the SEN_U for the UBIRIS.v2.

Table 4 compare the proposed method with other methods in recent publications. Results obtained by SEN improve over the former state-of-the-arts by 2.22% and 22.03% on UBIRIS.v2 and CASIA.v4-distance separately. We achieve the state-of-the-art results without any pre- and post-processing. Figure 6 shows some segmentation results.

5. Conclusions

In this paper, we present an end-to-end model, namely Seg-Edge bilateral constraint network. The iris edge map generated from rich convolutional layers optimize the iris segmentation by aligning it with the iris boundary. The iris

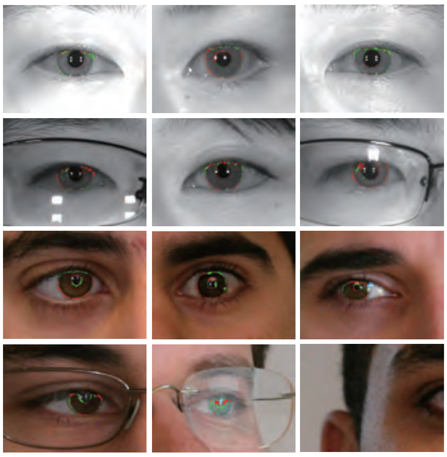


Figure 6. Segmentation results of two datasets by using the SEN. The green pixels indicate non-iris pixels are predicted as iris pixels, and the red pixels are the opposite.

region produced by the coarse segmentation limits the scope. It makes the edge filtering pay more attention to the interesting target. We compress the model while keeping the performance levels almost intact and even better by using l_1-norm. The proposed model advances the state-of-the-art iris segmentation accuracies.

References

[1] Casia.v4 database. `http://www.cbsr.ia.ac.cn/china/Iris%20Databases%20CH.asp`.

[2] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

[3] S. Bazrafkan, S. Thavalengal, and P. Corcoran. An end to end deep neural network for iris segmentation in unconstrained scenarios. *Neural Networks*, 106:79–95, 2018.

[4] L. Chen, J. Barron, G. Papandreou, K. Murphy, and A. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *CVPR*, pages 4545–4554, 2016.

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[6] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2018.

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

[8] M. Everingham, S. Eslami, L. Van G, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *Intl journal of computer vision*, 111(1):98–136, 2015.

[9] E. S. L. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. 30(4):69:1–69:12, 2011. SIGGRAPH.

[10] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[12] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[13] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[14] N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, and T. Tan. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In *ICB 2016*, pages 1–8, 2016.

[15] Y. Liu, M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *CVPR*, pages 5872–5881, 2017.

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[17] H. Proenca. Iris recognition: On the segmentation of degraded images acquired in the visible wavelength. *IEEE TPAMI*, 32(8):1502–1516, 2010.

[18] H. Proenca, S. Filipe, R. Santos, J. Oliveira, and L. Alexandre. The ubiris. v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. *IEEE TPAMI*, 32(8):1529–1535, 2010.

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Intl Conf on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[20] K. S and A. Z. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[21] J. O. Smith. *Mathematics of the discrete Fourier transform (DFT): with audio applications*. Julius Smith, 2007.

[22] C. Tan and A. Kumar. Unified framework for automated iris segmentation using distantly acquired face images. *IEEE TIP*, 21(9):4068–4079, 2012.

[23] C. Tan and A. Kumar. Towards online iris and periocular recognition under relaxed imaging constraints. *IEEE TIP*, 22(10):3751–3765, 2013.

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

[25] C. Tisse, L. Martin, L. Torres, and M. Robert. Person identification technique using human iris recognition. In *Proc. Vision Interface*, pages 294–299, 2002.

[26] R. Wildes. Iris recognition: an emerging biometric technology. *Proc of the IEEE*, 85(9):1348–1363, 1997.

[27] Z. Zhao and K. Ajay. An accurate iris segmentation framework under relaxed imaging constraints using total variation model. In *ICCV*, pages 3828–3836, 2015.