

Towards More Accurate Iris Recognition Using Deeply Learned Spatially Corresponding Features

Supplementary File

This supplementary file aims to provide more details on the experimental configurations which could not be included in the paper due to space limitation.

1. Details on Training/Test Set Composition

The details on the division of the training set and the test set on the four employed databases are provided in Table S1. Both the training set and the test set are formed with the first X ($X = 25, 10$ or 5 , shown in Table S1) or all of the left/right eye images from each of the subjects. If a subject has less than X images in the respective database, then all images from this subject will be included.

Table S1: Summary of the division for training set and test set on the employed databases.

Database	Training Set				Test Set			
	#subjects	#images/subject	side	#images	#subjects	#images/subject	side	#images
ND-IRIS-0405	all 356	first 25	left	9,301	all 356	first 10	right	3,394
CASIA.v4-distance	all 411	first 25	left	6,840	all 411	first 10	right	3,939
IITD	all 224	all	right	1,052	all 224	first 5	left	1,120
WVU Non-ideal	all 231	all	right	1,511	all 231	first 5	left	1,137

During the training phase, the triplet-based architecture introduced in Section 2 requires the input data to be triplet sets (anchor-positive-negative entries) instead of single images. Therefore the training images in each of the databases need to be presented as triplet entries which are generated from the combinations of images. However, enumerating all the possible triplet combinations in the training set will lead to high storage and computational complexity, we therefore selectively generate part of the possible triplet entries for training, as described in the following: For each training set, we firstly enumerate all the possible anchor-positive (genuine) pairs, since the numbers of available genuine pairs are relatively small; for each anchor-positive pair, we randomly select five negative samples that are from different subjects than the anchor subject, and form the anchor-positive-negative triplet. In other words, each genuine pair in the training set will generate five triplet entries for training.

2. Details on Tuning Comparative Methods

We have extensively tuned the benchmarking methods (*i.e.*, OSIRIS [13], 1D log-Gabor [2] and ordinal [4]) to ensure that their best possible performances are used for the fair

comparisons. We iteratively adopted possible combinations of the parameters for these approaches on each of the training sets within the empirically selected ranges, similar to as in many references (e.g., [2], [4] and [9]). The best performing parameters on the training sets were then employed on the respective test sets for the performance evaluation.

- Parameters for *IrisCode* (OSIRIS 2D Gabor filters):

A Gabor filter band containing six filters is provided in the original OSIRIS implementation [13]. In addition to the default one, we generated five Gabor filter bands for tuning this tool to obtain the best performance. Based on [1], a 2D Gabor filter for generating *IrisCode* can be formulated as:

$$g(x, y) = e^{-\left(\frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2}\right)} e^{-i\omega x} \quad (1)$$

Each set of parameters (α, β, ω) can be used to produce two filters which are the real and imaginary parts of the complex filter kernel. We apply three sets of parameters to form a band of six filters. The five additional Gabor filter bands are then generated using the following parameters:

- (i) $(\alpha, \beta, \omega) \in \{(3, 1.5, 0.4\pi), (5, 1.5, 0.2\pi), (7, 1.5, 0.1\pi)\}$
- (ii) $(\alpha, \beta, \omega) \in \{(3, 1.5, 0.4\pi), (5, 1.5, 0.3\pi), (7, 1.5, 0.2\pi)\}$
- (iii) $(\alpha, \beta, \omega) \in \{(5, 2, 0.3\pi), (7, 2, 0.2\pi), (9, 2, 0.1\pi)\}$
- (iv) $(\alpha, \beta, \omega) \in \{(3, 2, 0.3\pi), (6, 2, 0.2\pi), (9, 2, 0.1\pi)\}$
- (v) $(\alpha, \beta, \omega) \in \{(5, 1.5, 0.3\pi), (7, 1.5, 0.2\pi), (9, 1.5, 0.1\pi)\}$

- Parameters for *IrisCode* (1D log-Gabor filter):

Based on the model presented in [2], two parameters were tuned as follows:

σ/f (bandwidth over frequency): ranges from 0.3 to 0.6, with a step of 0.05.

λ (wavelength): ranges from 15 to 40, with a step of 1.

182 combinations in total.

- Parameters for ordinal filter based method:

Based on the model presented in [4], four parameters were tuned as follows:

n (number of lobes): ranges between $\{2, 3\}$.

s (size of each lobe): ranges among $\{5, 7, 9\}$.

d (distance between lobes): ranges among $\{5, 9, 13, 17\}$.

σ (standard deviation of each lobe): ranges among $\{1.5, 1.7, 1.9\}$.

72 combinations in total.

The best parameters automatically selected using the above detailed steps are provided in Table S2. It can be observed that such optimal parameters vary for one dataset to another, which underlines the *need* for selecting parameters for conventional methods according to the imaging environments and the quality of images for different databases. In contrast, our *CrossDB* model is able to deliver stable and satisfactory performance on the four public databases without any tuning, as shown in Figure 7 and Table 2 in the paper.

Table S2: Best performing parameters for *IrisCode* [2] [13] and Ordinal filters [4] on four employed databases.

Method	Parameter	ND-IRIS-0405	CASIA.v4-distance	IITD	WVU Non-ideal
<i>IrisCode</i> (2D Gabor - OSIRIS)	config.	default	(iii)	(ii)	(i)
<i>IrisCode</i> (1D log-Gabor)	σ/f λ	18 0.45	24 0.35	18 0.4	15 0.55
Ordinal filter	n	3	3	2	3
	s	5	9	7	9
	d	9	13	5	5
	σ	1.9	1.7	1.9	1.9

It is worth mentioning that we did *not* use the original or built-in iris segmentation/normalization procedures from OSIRIS and Masek’s 1D log-Gabor implementation. Iris segmentation has been shown to have significant impact on the recognition accuracy. Therefore to ensure the fairness in the evaluation of proposed iris feature representation, we uniformly adopt [10] for iris detection and normalization (as [10] has shown superior results on multiple public databases and also provides implementation codes), and use the output of *MaskNet* as the iris masks for our method and other investigated methods in this paper.

3. Analysis on Complexity

The computational complexity of our model has been evaluated in order to address the potential concerns on the feasibility for the deployment. Since our FCN does not employ fully connected layers, the number of parameters is significantly reduced and therefore it is much spatially simpler than conventional CNN based architectures. Table S3 summaries the computational time for feature extraction and the storage required by our model, as compared with the CNN based approach in [28]. It can be noted that the space and time complexities for our approach are quite small.

Table S3: Summary of number of parameters, model storage size and feature extraction time per image, run with Matlab wrapper and C++ implementation, on Intel i7-4770 CPU, 16G RAM and Nvidia GTX670 GPU.

Approach	#Parameters	Model Size (Byte)	Feature Extraction Time	
			GPU	CPU
Ours	~ 110.7 K	1.5 M	7.6 ms	236 ms
DeepIrisNet [28]	~55,420 K	289.0 M	12.7 ms	335 ms

4. Comparison with VeriEye SDK 9.0

In this paper we have provided reproducible performance comparison with the *IrisCode* [2] [13] and ordinal filter based method [4]. Although these methods are widely cited and have shown to offer competitive performance in the literature, it can be interesting to provide comparison with some commercial solutions for iris recognition, as they are considered to be

more suitable and optimized for real-life deployment. We therefore performed comparative evaluation using a popular commercial product, VeriEye iris recognition SDK from Neurotechnology [14], which released the latest version 9.0 in October 2016 and is available with us. The VeriEye SDK accepts original eye images (without normalization) as input and has its built-in iris segmentation components. Since this software is not open-source for its core functions, we are not able to interference its iris segmentation process. Therefore, the comparison results presented in this section may not be fully representing the effectiveness of iris feature representation, which is the key focus of this paper. Instead, it can be a sample reference for overall performance evaluation. The results for the comparison are shown in Figure S1.

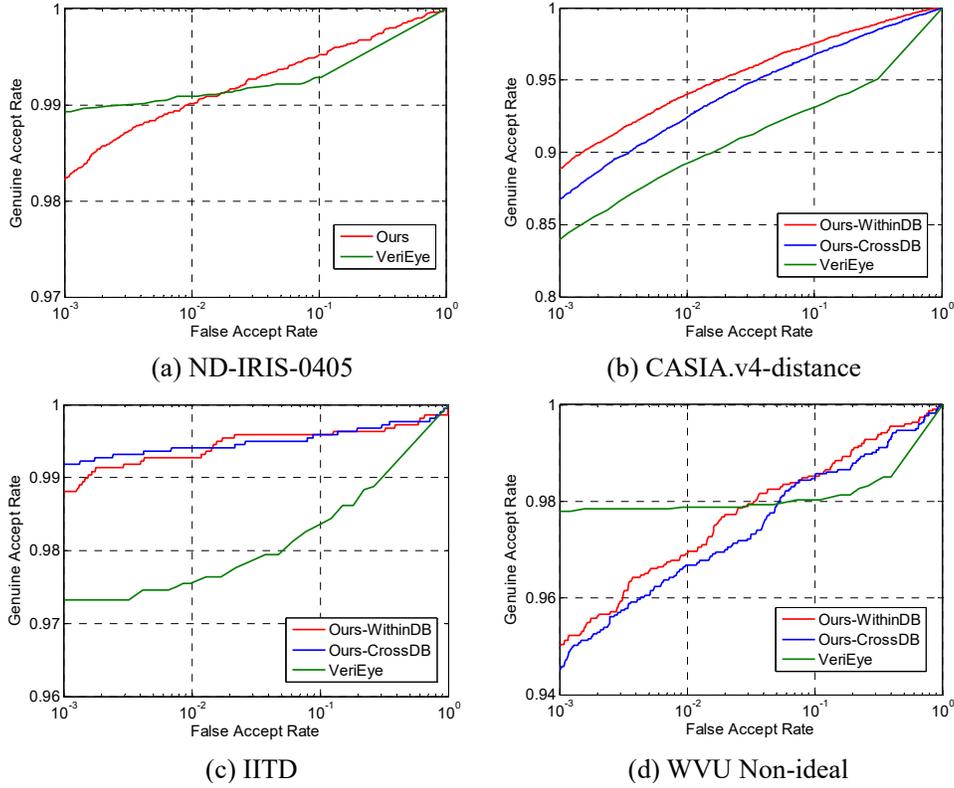


Figure S1: Receiver operating characteristic (ROC) curves from our approaches and the commercial product VeriEye SDK on four databases.

As shown in the figure, on ND-IRIS-0405 and WVU Non-ideal databases, VeriEye has better genuine accept rates (GAR) at lower false accept rates (FAR), while our approach consistently outperforms VeriEye on CASIA.v4-distance and IITD datasets. As discussed earlier, the difference in the segmentation process may have certain impact on the presented recognition results. Besides, VeriEye has a built-in quality assessment function that it does not match images with low quality, which may improve its overall performance to a certain extent, while our approach does not evaluate image quality at the current stage. Considering above factors, it is judicious to believe that our prototype model can already offer highly competitive performance compared with the well optimized commercial system.

5. Sample Evaluation for *MaskNet*

As mentioned in the paper, our key focus is on learning more effective iris feature representation. *MaskNet* is an essential component of *UniNet* for providing immediate and appropriate non-iris masking information to the proposed *Extended Triplet Loss (ETL)* function. In order to assert the adequateness of the masking information during the feature learning process, we have performed a sample evaluation of *MaskNet*. For the evaluation benchmark, we use more recent and advanced iris segmentation approach [10] as this method has already provided comparison with other promising methods in the literature. Similar to as used in [10], the average segmentation error is defined as follows:

$$\bar{e} = \frac{1}{N \times w \times h} \sum_{i=1}^N \sum_{x=1}^w \sum_{y=1}^h \mathbf{M}_i[x, y] \oplus \mathbf{G}_i[x, y] \quad (2)$$

where N is the number of test images, w and h are the weight and height of each image respectively, \mathbf{M}_i is the i -th computed mask and \mathbf{G}_i is the corresponding ground truth mask that should be manually generated. \oplus is the exclusive-OR (XOR) operation for two binary values. This error rate measures the level of disagreement between the computed mask and the ground truth. The difference with [10] is that we measure the segmentation error after iris normalization.

The *MaskNet* employed in our experiments was trained with 500 randomly selected left eye images from ND-IRIS-0405 database, with manually labeled iris masks as the ground truth. The test sets for its evaluation are also generated from the same database, excluding the training samples. We used the following two sets for the testing: (a) 100 randomly selected samples and their ground truth masks manually created by us; (b) 792 samples and their ground truth masks which are available from a public iris segmentation ground truth database, IRISSEG-EP [H. Hofbauer, F. Alonso-Fernandez, P. Wild, J. Bigun and A. Uhl, “A ground truth for iris segmentation”. *ICPR*, 2014]. The average segmentation errors of *MaskNet* and [10] are shown in Table S4.

Table S4: Comparison of average segmentation errors from *MaskNet* and [10].

Approach	Average Segmentation Error	
	Set (a)	Set (b)
<i>MaskNet</i>	5.89%	9.00%
ICCV'15 [10]	6.73%	11.83%

The results shown in Table S4 suggest that for both test sets, the developed *MaskNet* can achieve superior segmentation accuracy compared with state-of-the-art iris segmentation approach. It is therefore reasonable to conclude that *MaskNet* is able to provide appropriate information for identifying valid iris region during the feature learning process via *ETL*.