

Iris Segmentation and Recognition using Dense Fully Convolutional Network and Multiclass Support Vector Machine Classifier

D.Meenakshi¹, Dr.M.Sivajothi²,Dr.M.Mohamed Sathik³

¹Research scholar, Research center for Computer Science, Sadakathullah Appa college,Manonmaniam Sundaranar University, India. E-mail: bcait2017@gmail.com

²Associate Professor, Department of Computer Science, Sri Parasakthi college for Women, Courtallam, Manonmaniam Sundaranar University, India

³Principal, Sadakathulla Appa College,Manonmaniam Sundaranar University, India

Article History: Received: 11 January 2021; Accepted: 27 March 2021; Published online: 4 Jun 2021

Abstract

Iris Recognition has been one of the most robust means of biometric recognition. In recent years, there are so many researches in iris recognition using deep learning. In this paper, deep neural networks are used for segmentation, feature extraction and classification. The Dense Fully Convolutional Network is used for segmentation. The segmented iris is normalized which is then converted to Gabor wavelet for feature extraction. The Gabor features are extracted using multiclass Support Vector Machine classifier. The proposed method is tested on CASIA Iris 1000 and IITD datasets. The performance of the proposed method is evaluated using accuracy, precision and recall measures. It is also compared with state-of-the-art methods.

Keywords: Convolutional Neural Network, iris segmentation, Dense Fully Convolutional Network, Gabor Wavelet

I. Introduction

Biometric recognition has been widely used in every security applications. There are several researches in the field of biometric recognition. One among them is iris recognition. The conventional iris recognition system consists of feature extraction and classification phases. In both the phases, deep learning has its footprints. The iris code was determined in [1] from an optimization perspective. The standard iris code is shown to be the result of an optimization problem that minimizes the distance between feature values and iris codes. Furthermore, by adding terms to the objective function of this optimization problem, more efficient iris codes can be obtained.

Four new network models had been trained and tested on same datasets[2]. The identified best model is Fully Dilated convolution combining U-Net (FD-UNet). It used dilated convolution instead of original convolution to extract more global features. Combining dense network with U-Net, denseU-Net is created, which is narrower and has fewer parameters[3]. It also reduces the training difficulty of deep network.

To extract iris features, a Deep Convolutional Neural Network (DCNN) is trained using a large number of iris samples [4]. The problem of inadequate discrimination caused by the conventional Softmax loss function is solved using an optimized center loss function called Tight Center (T- Center) Loss. The impact of iris segmentation on the output of deep learning models is investigated in [5] using a simple two-stage pipeline consisting of a segmentation and recognition phase. It also demonstrated how segmentation accuracy affected recognition efficiency.

For more precise iris identification, periocular knowledge can be dynamically reinforced by integrating variations in the effective area of usable iris regions. Wang and Kumar discussed a periocular aided dynamic system for more precise less-constrained iris detection [6]. Miyazawa et al. presented a phase-based iris recognition algorithm that is specifically designed for system implementation [7]. The concept of 2D Fourier Phase Code (2D FPC) is implemented for representing iris information in order to minimize the size of iris data and prevent the exposure of iris images. 2D FPC is the quantized version of an iris image's phase spectrum, which is needed for phase-based iris recognition. As compared to using an iris image directly as registration

data, the size of iris data can be reduced to less than a quarter using 2D FPC while maintaining a sufficient level of performance.

To overcome the problem of low contrast and noise interference in the iris image, the Total Variation model is used in [8]. To find the iris area, an integro-differential constellation [9] is introduced, followed by a curvature fitting model. The iris map in [10] is generated using the random walker algorithm. Precaused neural networks as classifiers to detect the sclera and iris regions separately and polynomial fitting is used to estimate the final iris area [11]. Tan and Kumar described a post-classification procedure that includes reflection and shadow removal [12]. To improve the performance on the iris segmentation, several refinements had been done on pupil and eyelid localizations.

In [13], the Histogram of Oriented Gradients is used as feature and Support Vector Machine (SVM) is used to classify the iris features. Umer et al. used a restricted Hough transform algorithm to locate the iris boundaries [14]. Wild et al. [15] combined various iris segmentation algorithms to improve the accuracy. Gangwaret al. [16] used a coarse-to-fine strategy to identify different iris boundaries.

Umer developed a multiscale morphologic features-based iris recognition algorithm in [17]. Kumar [18] used a combination of Haar wavelet, Discrete Cosine Transform, and Fast Fourier Transform features to improve the performance. Farouk designed a scheme that employs elastic graph matching and the Gabor wavelet in [19]. Each iris is interpreted as a labeled graph, and the two graphs are compared using a similarity function. Minaee implemented an algorithm based on textural and scattering transform features [20], which improved the performance. This approach used predefined filters to extract features from a multi-layer representation.

The remaining of the paper is organized as: Section 2 discussed some recent methods that use deep learning for iris recognition. Section 3 elaborates the proposed methodology with its architecture. Section 4 demonstrates the proposed method with its experimental results followed by conclusion in Section 5.

II. Related Work

A Dense-Fully Convolutional Network (DFCN) based architecture [21] with dense blocks for iris segmentation is implemented, and some common optimizer methods, such as batch normalization and dropout, are adopted. The use of deep features derived from VGG-Net for iris recognition was investigated in [22]. For iris recognition, an end-to-end deep learning system based on residual CNN [23] has been proposed, which jointly learned the feature representation and performed recognition.

In [24], a reconstruction loss driven unsupervised pre-training stage is implemented, followed by supervised refinement, to address the lack of labeled iris data. As a result, the network weights are pushed to concentrate on discriminative iris texture patterns. To better exploit iris textures, many textures conscious improvisations are built within a CNN. It was demonstrated that systematic training and architectural choices allowed for the development of an efficient framework with up to 100 fewer parameters than current deep learning baselines while still achieving superior recognition performance for both within and cross dataset evaluations.

Fully Convolutional Encoder Decoder Networks (FCEDNs) are superior to all conventional algorithms, according to Jalilian and Uhl [25]. They proposed three iris segmentation network architectures based on FCNs and called them Fully Convolutional Encoder Decoder Networks (FCEDNs). Arsalan et al. [26] proposed IrisDenseNet, a segmentation network that uses the entire image without preprocessing and can evaluate the true iris boundary even with low-quality images by improving information gradient flow between dense blocks. Iris segmentation models based on the residual skip relation were proposed by Arsalan et al. [27].

III. Proposed Method

In this paper, we propose an algorithm for iris recognition by using DFCN for segmentation. The segmented iris region is normalized to change the size as rectangle shape for making further process easy. Gabor features are

extracted from the normalized segmented iris region which is then classified using multiclass SVM classifier. The flow chart of the proposed system is shown in Fig. 1.

The proposed methodology consists of segmentation, normalization, feature extraction and classification phases. The feature extraction consists of an additional step of wavelet conversion.

DFCN

The DFCN consists of Dense-encoder and decoder. In the Dense-encoder, the input images are first convolved by convolutional layer which is followed by 5 dense blocks. The dense blocks are composed of two 3x3 convolutional layers and feature maps are added through a dense block. The convolutional layers of the dense blocks are followed by the batch normalization layers and ReLU layers. Pooling layers follow after each dense block. The pooling layers perform the average pooling operation. The last pooling layer is followed by two convolutional layers and the score layers. The score layer is composed of n kernels, where n is 2 in this paper, namely, the iris pixels and non-iris pixels.

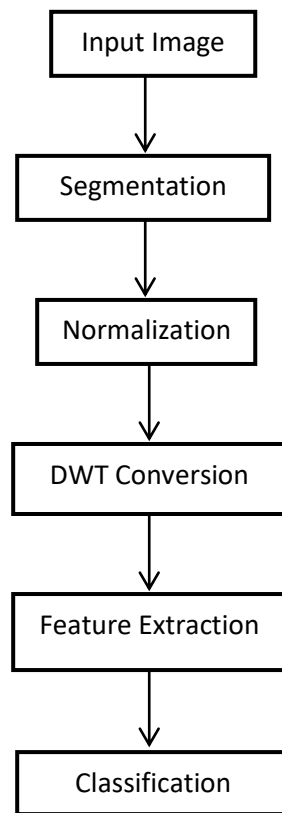
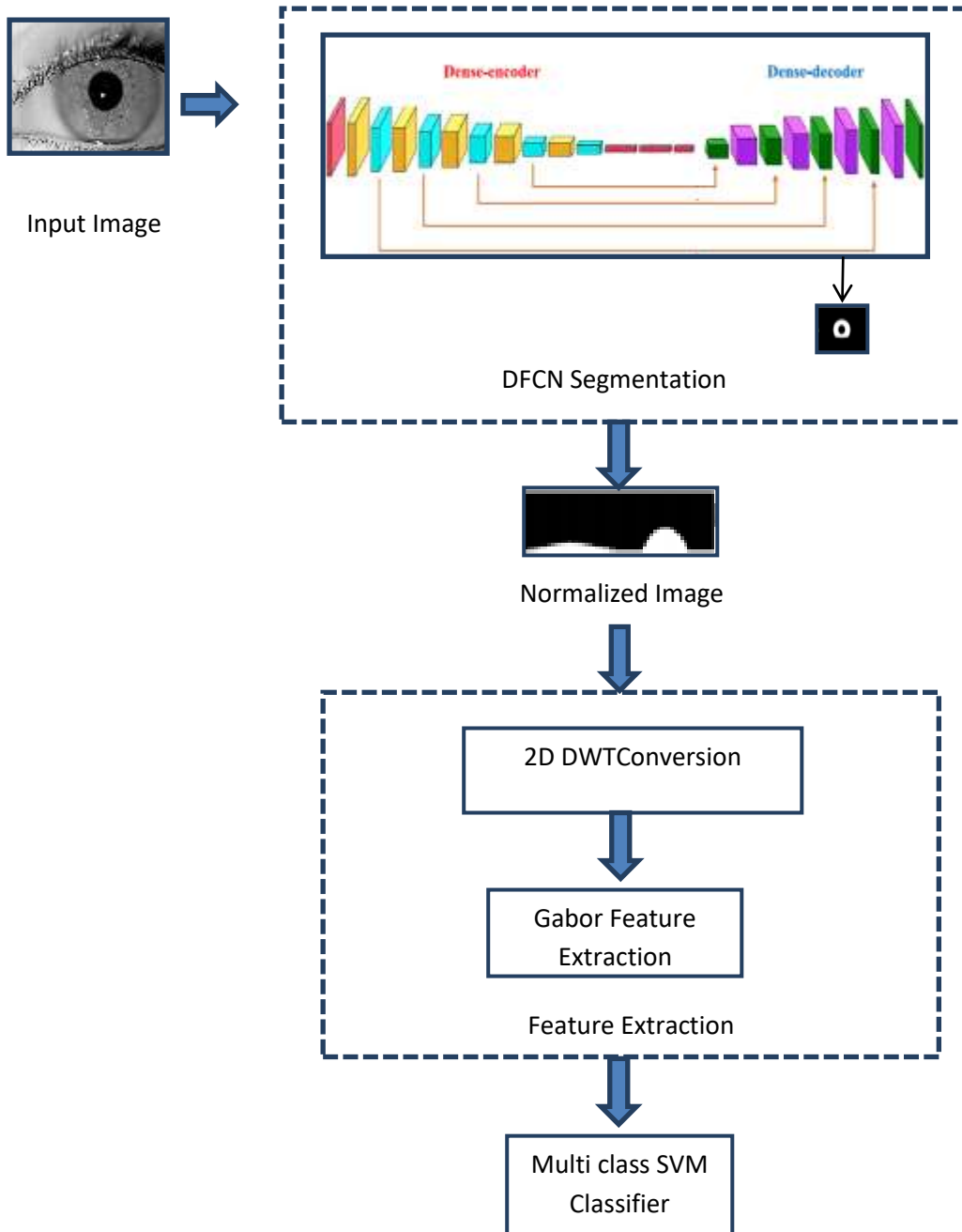


Fig. 1 Flow Chart of the Proposed System

In the Dense-decoder, the n channel feature maps obtained from the Dense-encoder are sampled by 5 transpose layers and the output prediction masks are obtained. The pixel values of the prediction masks are 0 or 1, where 0 denotes the non-iris pixels and 1 represents their iris pixels. Except for the last transpose convolutional layers, all transpose convolutional layers fuse the same size output of the pooling layers in the Dense-encoder by adding the pixel values of the feature maps. Additionally, all fuse layers are followed by processing layers. The process layers are composed of two successive convolutional layers. The first one consists of double numbers of previous layers kernels. The second one consists of half numbers of previous layers kernels. All convolutional kernels are padded with zeros. At the end of the Dense-decoder, the cross entropy function is adopted as the cost function. Fig. 2 shows the architecture of the proposed Iris Recognition system.

In DFCN architecture of Fig. 2, ‘Conv’ represents convolutional layers, ‘DB’ represents dense blocks, ‘PL’ represents pooling layers, ‘Conv_t’ represents transpose convolutional layers and ‘Pro_L’ represents processing layers. The details of DFCN architecture with its encoder and decoder are given in Table 1 and 2.



Conv DB PL Conv_t Pro_L

Fig. 2 Proposed System Architecture of Iris Segmentation

Table 1 Architecture of DFCN – Dense Encoder

| Layer | Stride | No. of Kernel | Output Size |
|----------|--------|---------------|-----------------|
| Input | - | - | 300 x 400 x 3 |
| Conv1 | 3x3 | 64 | 300 x 400 x 64 |
| DB1 | 3x3 | 48 | 300 x 400 x 160 |
| Pool1 | 2 x 2 | - | 150 x 200 x 160 |
| DB2 | 3x3 | 48 | 150x 200 256 |
| Pool2 | 2 x 2 | - | 75 x 100 x 256 |
| DB3 | 3x3 | 48 | 75 x 100 x 352 |
| Pool3 | 2 x 2 | - | 38 x 50 x 352 |
| DB4 | 3x3 | 48 | 38 x 50 x 448 |
| Pool4 | 2 x 2 | - | 19 x 25 x 448 |
| DB5 | 3x3 | 48 | 10 x 13 x 544 |
| Pool5 | 2 x 2 | - | 10 x 13 x 544 |
| Conv6 | 7 x 7 | - | 10 x 13 x 4096 |
| Dropout1 | - | - | - |
| Conv7 | 1 x 1 | - | 10 x 13 x 4096 |
| Dropout2 | - | - | - |

Normalization

To compare two iris images, it is easy if the segmented iris region is set to a particular size i.e. rectangle shape. This normalization is done using Daugman’s model [28]. In this process, the center of the radial circle across the iris region is considered as the reference point. The transformation is done using the following equations:

$$\begin{aligned}
 x(r, \theta) &= (1-r)x_p(\theta) + rx_i(\theta) \\
 y(r, \theta) &= (1-r)y_p(\theta) + ry_i(\theta)
 \end{aligned}
 \tag{1}$$

Where $x_p(\theta)$ denotes the point of the detected boundary of the segmented pupil and θ is the angle of the center of the pupil. $x_i(\theta)$ and $y_i(\theta)$ denotes the coordinates of the points on contour of the iris, obtained by the same principle.

Table 2 Dense Decoder Architecture

| Layer | Stride | No. of Kernel | Output Size |
|-----------------|--------|---------------|----------------|
| Conv_t1 | 3 x 3 | 448 | 19 x 25x 448 |
| Fuse1 | - | - | 19 x 25x 448 |
| Process Layer 1 | 1 x1 | 448 | 19 x 25x 448 |
| Conv_t2 | 3 x 3 | 352 | 38 x 50 x 352 |
| Fuse2 | - | - | 38 x 50 x 352 |
| Process Layer 2 | 1 x1 | 352 | 38 x 50 x 352 |
| Conv_t3 | 3 x 3 | 256 | 75 x 100 x 256 |
| Fuse3 | - | - | 75 x 100 x 256 |
| Process Layer 3 | 1 x1 | 256 | 75 x 100 x 256 |
| Conv_t4 | 3 x 3 | 160 | 15 x 200 x 160 |
| Fuse4 | - | - | 15 x 200 x 160 |

| | | | |
|-----------------|-------|-----|------------------------------|
| Process Layer 4 | 1 x1 | 160 | 15 x 200 x 160 |
| Con_t5 | 3 x 3 | 2 | 300 x 400x Number of classes |
| prediction | - | - | 300 x 400 x 1 |

The normalized image is of size 80×512 . The width of the image is the variation on the angular axis and the height is the variations on the radial axis. The normalized segmented iris region looks like the one shown in Fig. 2.

Feature Extraction and Classification

One of the important phases in the iris segmentation is feature extraction. The features are extracted from the normalized segmented iris region using Gabor wavelet transform. The feature extraction algorithm is in given in Algorithm 1.

From the literature, it is studied that wavelet coefficients gives more features from an image than spatial coefficients. Hence, the input image is transformed into wavelet coefficients using Discrete Wavelet Transform. Since bi-orthogonal wavelet coefficients gives better features for iris images than the other wavelets, it is used in this research work. In this work, the bi-orthogonal wavelet 6.8 is used. Only the low frequency components are used for feature extraction. Let the normalized segmented iris images be Ω .

Algorithm 1: Feature Extraction

Input: Normalized Segmented Iris Images Ω , 2-D Gabor function h

Output: Feature Vector f

Steps:

1. Initialize number of scales $S = 4$
2. Initialize number of orientations $K = 8$
3. For every Image I in Ω
 - 3.1 Calculate Gabor Filtered Image using

$$I_{g_{m,n}}(r, c) = \sum_s \sum_t I_E(r - s, c - t) h_{m,n}^*(s, t)$$
 - 3.2 Calculate mean of the energy of the Gabor filtered image using

$$\mu_{mn} = \frac{\sum_r \sum_c |I_{g_{m,n}}(x, y)|}{H \times W}$$
4. End
5. Concatenate feature Vector as $f = [\mu_1 \mu_2 \mu_3 \dots \mu_b]$
6. End

Gabor features are extracted from the low frequency coefficients obtained from pre-processing step. The 2D-Gabor function is defined as

$$h(x, y) = \exp \left[-\frac{1}{2} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right] \cos(2\pi Ux + \varphi) \tag{2}$$

$$h_{(m,n)}(x, y) = a^{-m} h(\tilde{x}, \tilde{y}), a > 1 \tag{3}$$

Where $\tilde{x} = a^{-m}(x \cos \theta + y \sin \theta)$, $\tilde{y} = a^{-m}(-x \sin \theta + y \cos \theta)$, $\theta = \frac{n\pi}{K}$ (K =total orientation, $n=0,1,\dots, K-1$). Step 3.1 in the Algorithm 1 gives the Gabor filtered image, in which $m = 0,1, \dots, S - 1, n = 0,1, \dots, K - 1, K$ is number of orientations and S is number of scales.

In Gabor feature extraction, there are several post-processing techniques such as complex moment features and Gabor-energy features. In this work, the Gabor-energy features are used. It is designed using K, S , and the upper and lower center frequencies U_h and U_l respectively.

$$a = \left(\frac{U_h}{U_l} \right)^{\frac{1}{S-1}}, \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2 \ln 2}} \tag{4}$$

$$\sigma_v = \tan\left(\frac{\pi}{2K}\right) \sqrt{\frac{U_h^2}{2 \ln 2} - \sigma_u^2}, W = U_h(5)$$

Assuming H, W are height and width of the image respectively, the mean of the energy of the Gabor filtered image is given in Step 3.2 of Algorithm 1. The extracted features of lower coefficients of the input images are given in Step 5, where b is the number of the input images.

The extracted features are classified using SVM classifier.

III. Experimental Results

This section starts with the explanation of datasets used in this paper. Then the hyperparameters that are used for executing DFCN models are discussed. This is followed by the discussion of the experimental results obtained by the proposed method. The performance measures that are used by the proposed method are presented next. Finally, the results are compared with some recent methods.

Details of Datasets

We have tested our algorithm on two iris databases such as CASIA-Iris-1000 [29] and IIT Delhi (IITD) [30]. The CASIA-Iris-1000 images are collected using an IKEMB-100 camera Table 3 displays the details of each dataset.

Table 3 Properties of Datasets

| Property | CASIA 1000 | IITD |
|--------------------------|------------|-----------|
| Number of Classes | 1000 | 224 |
| Number of Images | 20000 | 2240 |
| Image Size | 640 x 480 | 320 x 240 |
| Image Format | JPG | BMP |
| Illumination Environment | NIR | NIR |

The dataset should be divided into training and testing sets to implement the proposed method. Table 4 gives the details after dividing the dataset for training and testing.

Table 4 The details of the dataset after splitting

| Dataset | CASIA 1000 | IITD |
|-------------------------|------------|------|
| Number of training sets | 10000 | 672 |
| Number of test sets | 10000 | 1568 |

Hyperparameters Setup for DFCN Model

The hyper-parameters and the execution environment used in the training process of DFCN are presented in Table 5. In the DFCN model, the learning rate decreases every 10000 steps by a factor of 10. The weights of all convolutional kernels are initialized as a truncated normal distribution with a standard deviation of 0.01, and the bias of all convolutional layers is 0. The values of μ and ν of the mini-batch Adam algorithm are set to 0.9 and 0.99, respectively

Table 5 Experimental Setup of System and Hyper-parameters of Neural Network Models

| Model | Execution Environment | Hyper-Parameters |
|-------|--|---|
| DFCN | Inter Core i9-7900x CPU with 32 GB memory and an NVIDIA 1080ti GPU with 11 GB memory | <ul style="list-style-type: none"> • Learning rate - 0.001. • Iterative step – 30000 • Batch size - 2 • Optimization - Adam optimizer |

IV. Results of the Proposed Method

Input images, groundtruth and the results obtained by the proposed method from IITD and CASIA Iris 1000 datasets are shown in Fig. 3 and 4 respectively.

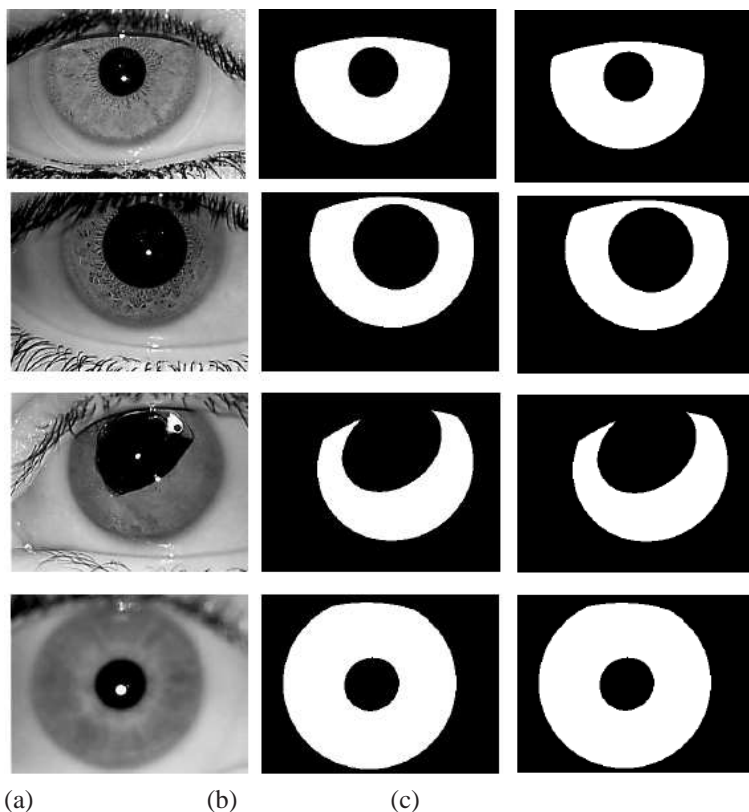


Fig. 3 (a) Input Images (b) Groundtruth (c) Results obtained by the Proposed Method for IITD Dataset

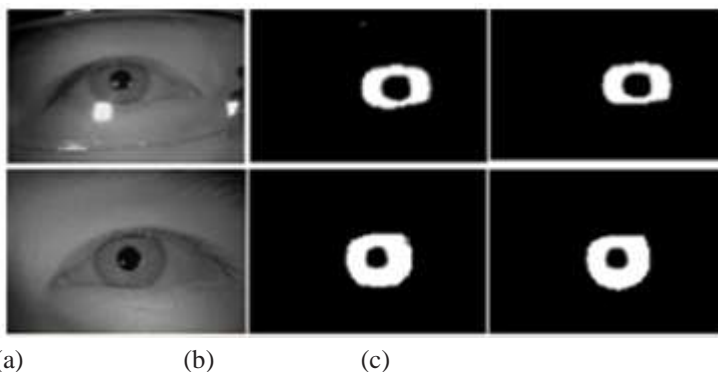


Fig.4 (a) Input Images (b) Groundtruth (c) Results obtained by Proposed Method for Casia Iris 1000 Dataset

Performance Measures

The performance the proposed method is evaluated using accuracy, precision and recall measures. The accuracy gives the correct segmentation pixels and is calculated as follows:

$$acc = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \times 100 \tag{6}$$

The precision gives the average proportion of the correct classification number of iris pixels that are classified as iris pixels. The precision is calculated as follows:

$$P = \frac{T_p}{T_p + F_p} \tag{7}$$

The recall denotes the average proportion of the correct classification number of iris pixels that belong to iris pixels. The recall is calculated as follows:

$$R = \frac{T_p}{T_p + F_n} \tag{8}$$

Comparison of Proposed Method with Recent Methods

The results of the proposed method are compared with recent methods such as DFCN [2], VGGNet [6], DeepIris [8], CNN with Iris Texture [10] and FCEDN [28] which are discussed in Section 2. Table 6 shows the accuracy comparison of the proposed method with recent methods for CASIA Iris 1000 and IITD dataset.

Table 6 Accuracy Comparison of Proposed Method with Recent Methods

| Dataset / Method | CASIA 1000 | IITD |
|----------------------------|------------|-------|
| DFCN[2] | 98.05 | 98.84 |
| VGGNet[6] | 97.6 | 99.4 |
| DeepIris [8] | 92.4 | 95.5 |
| CNN with Iris Texture [10] | 92.61 | - |
| FCEDNs- Original [28] | 94.39 | 94.09 |
| FCEDNs- Basic [28] | 95.52 | 94.61 |
| FCEDNs-Bayesian Basics[28] | 96.09 | 93.28 |
| Proposed Method | 98.85 | 99.8 |

From the Table 6, it is observed that the DFCN method has high accuracy of 98.05% for CASIA 1000 dataset. But our method outperforms the DFCN method by achieving an accuracy of 98.85%. In IITD dataset, the method which uses VGGNet model has the highest accuracy of 99.4%. The accuracy of the proposed method is little higher than that method. Fig. 5 shows the bar chart comparison of the accuracy obtained by the proposed method and recent methods.

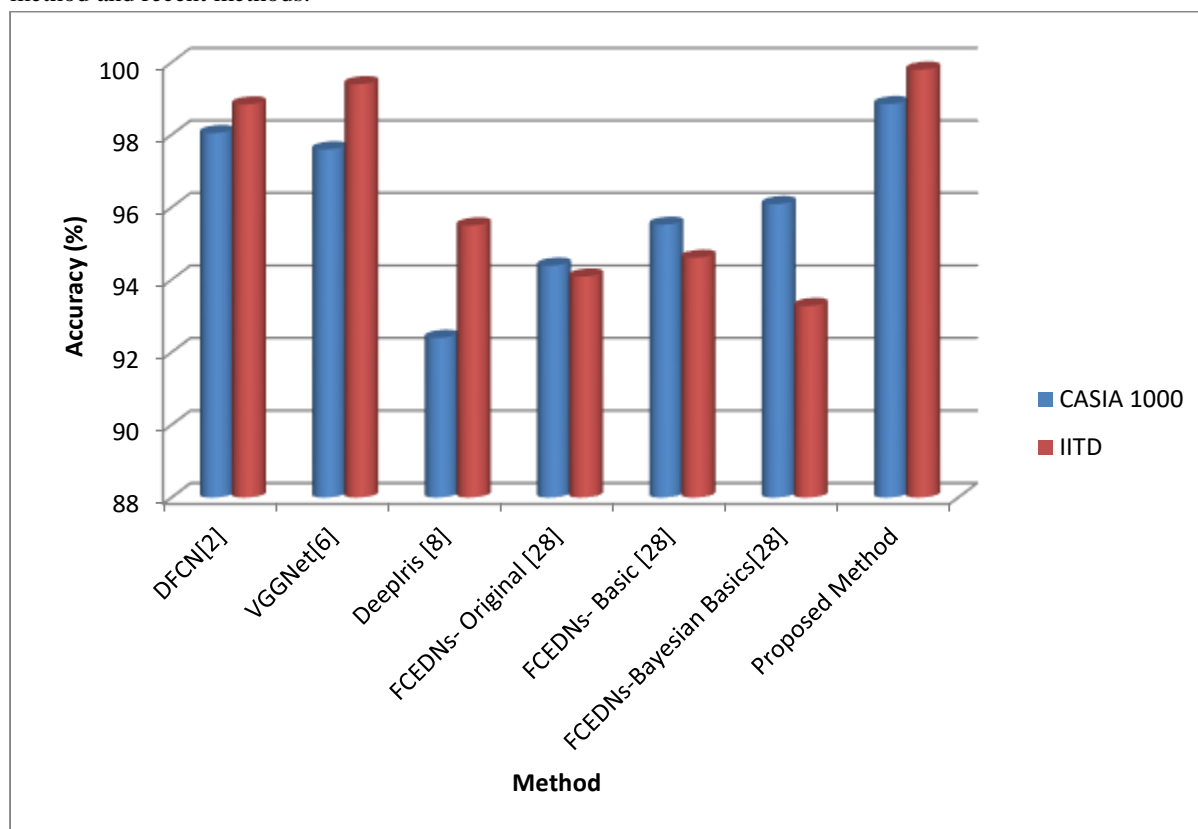


Fig. 5 Bar Chart Comparison of Proposed Method with Recent Methods

Table 7 shows the comparison of precision and recall values of the proposed method with other methods such as IrisDenseNet [29] and FRED Net [30].

Table 7 Precision and Recall Comparison of Proposed Method with Recent Methods

| Measure/ Method | CASIA 1000 | | IITD | |
|------------------|------------|--------|-----------|--------|
| | Precision | Recall | Precision | Recall |
| IrisDenseNet[29] | 0.9810 | 0.9710 | 0.9716 | 0.9800 |
| FRED Net[30] | 0.9659 | 0.9910 | 0.9253 | 0.9968 |
| Proposed Method | 0.9768 | 0.99 | 0.9279 | 0.99 |

From the Table 7, it is studied that the precision and recall measure obtained by the proposed method is higher than other methods.

IV. Conclusion

Biometric recognition has many researches in various applications. One among them is iris recognition. To segment the iris, first we use DFCN and then normalize the segmented region into a rectangle shape using Daugman's rubber sheet method. Then Gabor features are extracted from the normalized iris region which is classified using multiclass SVM classifier. The proposed method is tested on two publicly available datasets such as CASIA Iris 1000 and IITD datasets. The efficiency of the proposed method is evaluated using accuracy, precision and recall measures. The experimental results proved that the proposed method achieves the highest accuracy of 98.85% and 99.4% for CASIA 1000 and IITD datasets respectively which is better than other methods.

References:

- [1] Hu, Y., Sirlantzis, K. and Howells, G., 2016. Optimal generation of iris codes for iris recognition. *IEEE Transactions on Information Forensics and Security*, 12(1), pp.157-171.
- [2] Zhang, W., Lu, X., Gu, Y., Liu, Y., Meng, X. and Li, J., 2019. A robust iris segmentation scheme based on improved U-net. *IEEE Access*, 7, pp.85082-85089.
- [3] Wu, X. and Zhao, L., 2019. Study on iris segmentation algorithm based on dense U-Net. *IEEE Access*, 7, pp.123959-123968.
- [4] Chen, Y., Wu, C. and Wang, Y., 2020. T-center: A novel feature extraction approach towards large-scale iris recognition. *IEEE Access*, 8, pp.32365-32375.
- [5] Lozej, J., Štepec, D., Štruc, V. and Peer, P., 2019, May. Influence of segmentation on deep iris recognition performance. In *2019 7th International Workshop on Biometrics and Forensics (IWBF)* (pp. 1-6). IEEE.
- [6] Wang, K. and Kumar, A., 2019. Segmentation-Aware and Adaptive Iris Recognition. *arXiv preprint arXiv:2001.00989*.
- [7] Miyazawa, K., Ito, K., Aoki, T., Kobayashi, K. and Nakajima, H., 2006, December. An implementation-oriented iris recognition algorithm using phase-based image matching. In *2006 International Symposium on Intelligent Signal Processing and Communications* (pp. 231-234). IEEE.
- [8] Z. Zhao and 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.
- [9] [13] T. Tan, Z. He, and Z. Sun, "Efficient and robust segmentation of noisy iris images for noncooperative iris recognition," *Image Vis. Comput.*, vol. 28, no. 2, pp. 223-230, 2010.
- [10] C.-W. Tan and A. Kumar, "Towards online iris and periocular recognition under relaxed imaging constraints," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3751-3765, 2013.
- [11] H. Proenca, "Iris recognition: On the segmentation of degraded images acquired in the visible wavelength," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1502-1516, 2010.
- [12] C.-W. Tan and A. Kumar, "Unified framework for automated iris segmentation using distantly acquired face images," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4068-4079, 2012.
- [13] A. Radman, N. Zainal, and S. A. Suandi, "Automated segmentation of iris images acquired in an unconstrained environment using HOG-SVM and GrowCut," *Digit. Signal Process.*, vol. 64, pp.60-70, 2017.
- [14] S. Umer, B. C. Dhara, and B. Chanda, "Texture code matrix-based multi-instance iris recognition," *Pattern Anal. Appl.*, vol. 19, no. 1, pp. 283-295, 2016.
- [15] P. Wild, H. Hofbauer, J. Ferryman, and A. Uhl, "Segmentation-level fusion for iris recognition," in *Proc. Int. Conf. Biometrics Special Interest Group*, 2015, pp. 1-6.
- [16] A. Gangwar, A. Joshi, A. Singh, F. Alonso-Fernandez, and J. Bigun, "IrisSeg: A fast and robust iris segmentation framework for non-ideal iris images," in *Proc. Int. Conf. Biometrics*, 2016, pp. 1-8.
- [17] S Umer, BC Dhara, and Bhabatosh Chanda. "Iris recognition using multiscale morphologic features." *Pattern Recognition Letters* 65: 67-74, 2015.

-
- [18] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern Recognition*, vol. 43, no.3, pp. 1016-1026, Mar. 2010.
- [19] RM. Farouk, "Iris recognition based on elastic graph matching and Gabor wavelets," *Computer Vision and Image Understanding*, Elsevier, 115.8: 1239-1244, 2011.
- [20] S Minaee, A Abdolrashidi and Y Wang, "Iris Recognition Using Scattering Transform and Textural Features", *IEEE Signal Processing Workshop*, 2015.
- [21] Chen, Y., Wang, W., Zeng, Z. and Wang, Y., 2019. An adaptive CNNs technology for robust iris segmentation. *IEEE Access*, 7, pp.64517-64532.
- [22] Minaee, S., Abdolrashidi, A. and Wang, Y., 2016, December. An experimental study of deep convolutional features for iris recognition. In *2016 IEEE signal processing in medicine and biology symposium (SPMB)* (pp. 1-6). IEEE.
- [23] Minaee, S. and Abdolrashidi, A., 2019. Deepiris: Iris recognition using a deep learning approach. *arXiv preprint arXiv:1907.09380*.
- [24] Chakraborty, M., Roy, M., Biswas, P.K. and Mitra, P., 2020, October. Unsupervised Pre-Trained, Texture Aware and Lightweight Model for Deep Learning Based Iris Recognition Under Limited Annotated Data. In *2020 IEEE International Conference on Image Processing (ICIP)* (pp. 1351-1355). IEEE.
- [25] E. Jalilian, A. Uhl, "Iris segmentation using fully convolutional encoder-decoder networks," in *Deep Learning for Biometrics*. 2017, pp. 133_155.
- [26] M. Arsalan, R. A. Naqvi, D. S. Kim, P. H. Nguyen, M. Owais, and K. R. Park, "IrisDenseNet: Robust iris segmentation using densely connected fully convolutional networks in the images by visible light and near-infrared light camera sensors," *Sensors*, vol. 18, no. 5, pp. 1501_1531, 2018.
- [27] M. Arsalan, D. S. Kim, M. B. Lee, M. Owais, and K. R. Park, "FRED-Net: Fully residual encoder-decoder network for accurate iris segmentation," *Expert Syst. Appl.*, vol. 122, pp. 217_241, May 2019.
- [28] Daugman, J. The importance of being random: Statistical principles of iris recognition. *Pattern Recognit.* **2003**, 36, 279–291.
- [29] <http://biometrics.idealtest.org/>
- [30] A Kumar and A Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern Recognition*, pp. 1016-1026, 2010.