Deep Learning-Based Iris Segmentation Algorithm for Effective Iris Recognition System

Sruthi Kunkuma Balasubramanian*, Vijayakumar Jeganathan, Thavamani Subramani

Department of Electronics and Instrumentation, Bharathiar University, Coimbatore, India

Received 29 April 2022; received in revised form 09 July 2022; accepted 10 July 2022

DOI: https://doi.org/10.46604/peti.2022.10002

Abstract

In this study, a 19-layer convolutional neural network model is developed for accurate iris segmentation and is trained and validated using five publicly available iris image datasets. An integrodifferential operator is used to create labeled images for CASIA v1.0, CASIA v2.0, and PolyU Iris image datasets. The performance of the proposed model is evaluated based on accuracy, sensitivity, selectivity, precision, and F-score. The accuracy obtained for CASIA v1.0, CASIA v2.0, CASIA Iris Interval, IITD, and PolyU Iris are 0.82, 0.97, 0.9923, 0.9942, and 0.98, respectively. The result shows that the proposed model can accurately predict iris and non-iris regions and thus can be an effective tool for iris segmentation.

Keywords: CNN, deep learning, iris recognition, iris segmentation

1. Introduction

Iris-based biometric recognition is one of the most efficient and extensively used automated individual identification techniques. In recent years, iris texture has played a crucial role for various security and defense purposes because irises are stable and unique structures. Since the iris pattern cannot be reproduced easily, the iris-based recognition technique surpasses most biometric recognition techniques [1]. In addition, with an iris-based biometric recognition system, it is not necessary to update the iris pattern at regular intervals since the iris pattern will not undergo degradation over time. Iris recognition involves the following processes: acquiring eye images, segmenting the iris parts from the eye images, extracting features, comparing the acquired features with the existing database, and matching the features. The effectiveness of the iris segmentation process determines the effectiveness of the overall iris-based biometric recognition system.

As shown in Fig. 1, the iris lies between the pupil and the sclera [2]. If these regions are segmented along the iris, the performance of the recognition system will be affected. For effective identification, iris segmentation plays an essential role. However, by using traditional iris segmentation algorithms, iris portions could be lost. With the development of deep learning, iris segmentation has become a little accessible compared to conventional segmentation algorithms because the deep learning algorithms will be trained for segmenting the iris using labeled data. In standard segmentation algorithms, the detection of elliptical or circular edges has been used to segment the pupil and iris.

The iris can be segmented more precisely when using the localization technique based on the circle equation [4] and Daugman's integrodifferential operator [5], as compared to the canny edge detector and active contour. In addition, all the traditional methods make the segmentation process complicated and lengthy. For these reasons, researchers started to conduct iris segmentation by using deep learning algorithms. Convolutional neural network (CNN) is a widely used deep learning technique for image segmentation. The CNN architecture predicts the iris and non-iris pixels by adopting a down-sampling layer via a fully connected layer. CNN can accurately recognize the features and automatically detect the essential components in an image.

* Corresponding author. E-mail address: Sruthi.ei@buc.edu.in

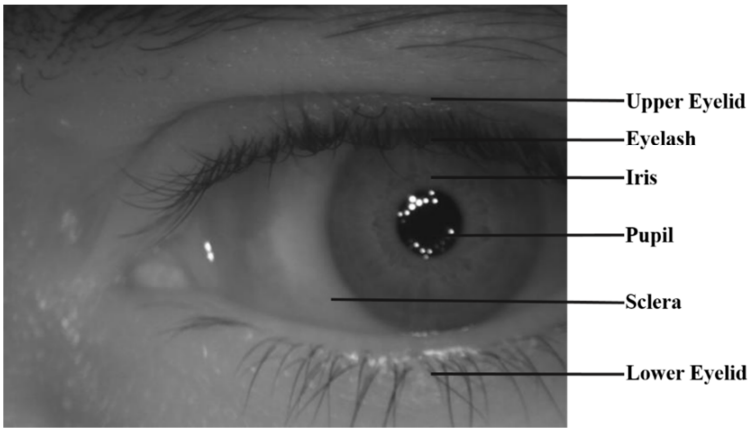

Fig. 1 Human eye [3]

This study aims to develop a CNN-based iris segmentation framework for iris-based biometric recognition systems. The significant contributions of this work are shown as follows.

(1) The proposed framework is not database-specific.

(2) Semi-manual processes are employed for labeling the publicly available iris databases.

The rest of this study is organized as follows. Section 2 reviews the studies related to iris segmentation. Section 3 describes the proposed workflow in detail. Section 4 describes the experimental configuration of this study, such as the employed public iris image datasets, ground truth masks, and hyperparameters. Section 4 also explains the evaluation metrics, and section 5 details the experimental results, comparisons with the existing systems for iris segmentation, and discussion. Finally, the study is concluded in section 6.

2. Related Works

This section briefly explains the state-of-the-art iris segmentation process using deep learning algorithms. UniNet.V2 was developed to detect, segment, and extract iris features from raw eye images [6]. The architecture consists of a mask region-based CNN (MR-CNN), a normalization layer, and a FeatNet layer for feature extraction. Mask R-CNN was introduced to improve stability and accuracy in detecting and segmenting irises. Iris segmentation was performed by identifying the iris and non-iris pixel regions. RoI Align was employed in down-sampled feature maps to recover the accuracy at pixel-level segmentation. The datasets used were ND-IRIS-0405, CASIA v4.0 Distance, IITD, and WVU non-ideal iris databases. A fully CNN (FCNN) with the triplet loss was used to perform the task of iris recognition.

Zhang et al. [7] proposed an improved U-Net which obtained better accuracy than the traditional method [7]. They combined the fully dilated convolution and the U-Net network (FD-UNet). The dilated convolution was combined with U-Net to acquire more information from the segmented image. U-Net consists of two paths, namely the contraction and expansion paths. The contraction path was responsible for pooling and feature extraction, whereas the expansion path was accountable for up-sampling. Every pooling block undergoes two unpadded convolution operations of 3×3 and a pool operation of 2×2. In the contraction path, the image size was reduced to half the original size, and the feature channels were doubled. In the expansion path, the images were resized to their original size, and the feature maps were reduced to half using the rectified linear unit (ReLU) activation function. In up-sampling, the output feature obtained from the contraction path was combined with the corresponding features obtained in the expansion path. The convolution operation with a 1×1 kernel was used to map the components to the appropriate class they belong to. The missing boundary problem was overcome by employing two dilated convolutions instead of unpadded convolution, but the final layer 1×1 convolution was retained. FD-UNet was trained and tested using CASIA Iris Interval (CASIA Interval), ND-IRIS-0405, and UBIRIS v2 datasets.

A dense FCNN, consisting of a CNN architecture combined with iris segmentation dense blocks, was proposed to improve the iris segmentation efficiency [8]. The thick blocks were used in both the encoder and decoder sections, and were trained and tested with CASIA Interval and IITD iris databases. Li et al. [9] developed an edge-based and learning-based algorithm to avoid misclassifying the pupil and limbus boundaries. A six-layer faster region-based CNN (R-CNN) and a Gaussian mixer model (GMM) were used for locating the eye and pupil boundaries, respectively. The limbus boundary was found by searching the maximum intensity pixels along the radius from the center of an eye image and using the edge selection methodology. Faster R-CNN consists of an input layer, four convolution layers with a ReLU, normalization, and a max-pooling layer at the second, third, and fourth layers. Faster R-CNN also consists of fully connected and output layers. A GMM based on the expectation-maximization algorithm was used to reduce the execution time and increase the system's accuracy. The database used for training and testing faster R-CNN was CASIA Iris Thousand.

A fully convolutional deep neural network (FCDNN) was developed to segment the iris from an eye image captured by a mobile camera [10]. Initially, FCDNN was trained using the NIR iris images from CASIA Thousand, Bath 800, UBIRIS v2, and MobBio datasets, later using visible images from the same publicly available databases. The data for training the network was increased to obtain better accuracy using data augmentation. The data augmentation includes changing contrast, applying blur, and shadowing randomly. A deep neural network based on the semi-parallel method (SPDNN) was used to generate the iris map from low-resolution iris images. This work combined SPDNN with FCDNN to obtain a network similar to U-Net without the pooling layer.

ATTention-guided U-Net (ATT-UNet) was proposed to segment the iris more accurately [11]. ATT-UNet was built on the usual U-Net. This work introduced attention mechanisms into the network to study more distinct features present in the image to separate non-iris and iris regions. The contracting path of U-Net was used for extracting the feature map, and finally, the regression module was used for estimating the iris' boundary box to serve as an attention mask. Feature maps were decoded in the expanded path, and an attention mask was added to the model at the final prediction stage. Skip function was employed to concatenate features from the contraction and expansion paths to force the model to obtain global and local information between the contracting and expanding paths. The potential area where the iris was most likely to be present was predicted by attention mask generation. The regression model was obtained by employing a pooling and fully connecting layer at the contracting path. The mean squared error function was used to find the network loss. Using the attention mask obtained, the network was guided to segment the iris at the final stage. Soft attention was integrated with the network to highlight the mistake performed in mask generation to a certain degree. The loss function in segmentation was calculated using the binary cross-entropy function. ATT-UNet was trained and tested using UBIRIS v4 and CASIA v4 Distance.

Hofbauer et al. [12] developed a CNN-based iris segmentation model to further improve the segmentation accuracy [12]. The refinement of the CNN architecture was chosen to obtain better segmentation performance. Four RefineNet were employed and individually connected to one residual net output simultaneously with the former RefineNet block. All four RefineNet implemented were in a cascade arrangement. The pupil boundary was found quickly as they are entirely circular and free from the interference caused due to the presence of eyelids and eyelashes. After detecting the pupil boundary, the limbic boundary was found. This work showed that implementing a CNN-based noise mask in the traditional method improved the accuracy of iris segmentation when the iris images were obtained from a high-quality biometric camera. CNN was trained and tested using CASIA Interval, IITD, ND-IRIS-0405, and prot-1.

An interactive U-Net with squeeze expand module (ISqEUNet) was employed to increase the storage efficiency by decreasing the number of parameters employed in the segmentation process [13]. ISqEUNet contained smaller datasets with few annotated images as well as the ISqEUNet-based segmented output for these iris images. The images that were not segmented correctly were refined manually. The interactive U-Net was trained and tested using CASIA Interval and IITD

datasets. The images segmented incorrectly were refined manually in the interactive mode by users. Fig. 2 shows the databases used in the reviewed articles. From Fig. 2, it can be seen that UBRIS v2, CASIA Interval, and IITD were commonly used databases. This study develops a unified 19-layer CNN architecture for iris segmentation, and trains the CNN model with the labels obtained from WaveLab or created using Daugman's integrodifferential operator [5].

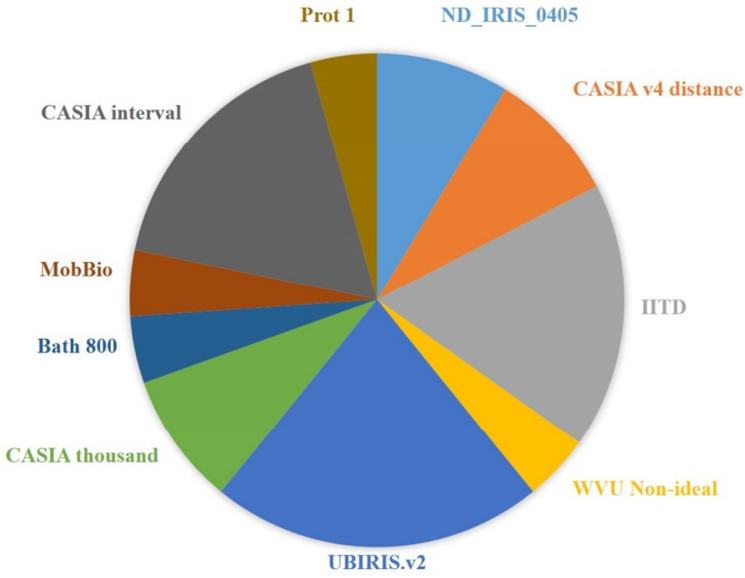

Fig. 2 Databases used in the reviewed articles

3. Methodology

In this section, the proposed methodology for iris segmentation is explained in detail. Fig. 3 shows the overview of the proposed pipeline. As shown in Fig. 3, the images obtained from various publicly available iris databases are used as input images for the proposed CNN model to perform iris segmentation. A detailed description of the process involved in the proposed system is discussed below.

The proposed architecture consists of 19 convolutional layers with variable filter sizes, activation functions, residual layers, skip functions, and batch normalization. The convolutional layers are employed for pixel-based classification to extract the iris region. Each convolution layer consists of a filter kernel for classifying the region of interest (ROI). The grayscale input image is filtered in the convolutional layer with varying kernels. For the n^{th} map of features with a k^{th} layer of convolution, the output x_k^n is given by Eq. (1).

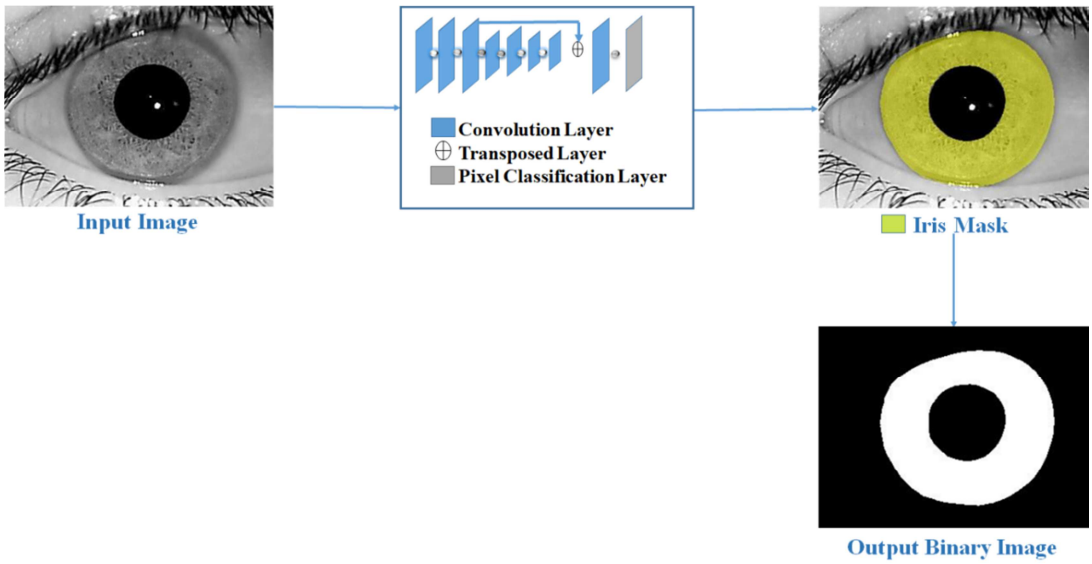

Fig. 3 Overview of the proposed methodology

$$x_k^n = f(W_k^n \times x_{k-1}^n) \tag{1}$$

where $f()$ represents the activation function, $*$ denotes the convolution, and W denotes the kernel filter size. The activation function used in the proposed work is ReLU in each convolutional layer to prevent the saturation of negative and positive inputs. The activation function of the input Y_i is calculated based on Eq. (2).

$$f(Y_i) = \begin{cases} 0, & Y_i < 0 \\ Y_i, & Y_i \geq 0 \end{cases} \tag{2}$$

Eq. (2) shows that the activation function is zero and linear for all negative and positive values, respectively. ReLU is a widely used activation function in CNN because of its minimum computational value since no complicated evaluation method is involved. Due to the minimal computational value, ReLU requires less execution time to train or execute. It has faster convergence because of the linearity property. Unlike sigmoid and tanh, ReLU does not suffer from vanishing gradient problems. ReLU is also sparsely activated, so better predictive power can be obtained. A pooling layer is used to reduce an image's size by applying the pooling window. The pooling window performs down-sampling by moving over the entire image. The pooling layer is of two types, i.e., average and max pooling. In this work, the max-pooling layer is employed. In max-pooling, the highest value pixel from all pixels covered by the pooling window is taken. The stride of a network layer is used for reducing the dimensions of the feature map. The stride is the number of shifts in pixels over an image given as the input. It controls the shift of kernels while carrying out convolution. Zero-padding is employed to fit a filter that does not fit perfectly.

The residual layer is employed to enhance the features by accumulating the previous layer features using skip connection to a learned residue as proposed by He et al. [15]. Residual layers are used for faster gradient propagation to decrease the execution time. The batch normalization technique is employed to normalize the weights of the network. In this way, the variation between the consecutive layers is reduced, resulting in a faster training process. The layers' extracted features are transformed linearly before applying the activation function. After the network weights are updated, the updated weights are applied to the ReLU layer. The proposed network is based on a supervised learning algorithm. The predicted image is verified based on its label image. A soft-max classifier layer is employed to compare every pixel of the predicted image to its ground-truth image. The input of this layer is obtained from the convolution layer. The soft-max function ς receives the input as q. m is the element of q, M is the number of classes to be classified, and p is the node on the output layer. This normalized function is given by Eq. (3).

$$l_m = \varsigma(q)_m = \frac{e^{q_m}}{\sum_{p=1}^{M} e^{q_p}} \; ; m = 1, 2, \ldots, M \tag{3}$$

Adam Optimizer is employed to reduce the loss while updating the network weights, similar to the work of Kingma et al. [16]. The optimizer improves the training rate for scarce data, deteriorates it for the mutual data, and revises rapidly for scarce data and slowly for mutual feature characteristics. The variation between the predicted and label pixels is considered a loss. The cross-entropy process is used to calculate the loss by the soft-max layer. Cross-entropy loss is one of the widely used functions for calculating the loss in deep learning. This loss function is employed to handle the binary classification problem used to compute whether a pixel belongs to a specific class or not [17]. The cross-entropy loss O_n for the proposed network's output is given by Eq. (4).

$$O_n = -\sum_n y_n \log \hat{l}_n \tag{4}$$

In Eq. (4), y_n represents the n^{th} component of the label, and $\hat{l}_n$ represents the n^{th} output of the soft-max classifier layer. Eq. (4) conveys that the cross-entropy gives the same weight to the loss of diverse pixels and does not account for the unbalanced pixel distributions. Fig. 4 shows the detailed architecture of the proposed network.

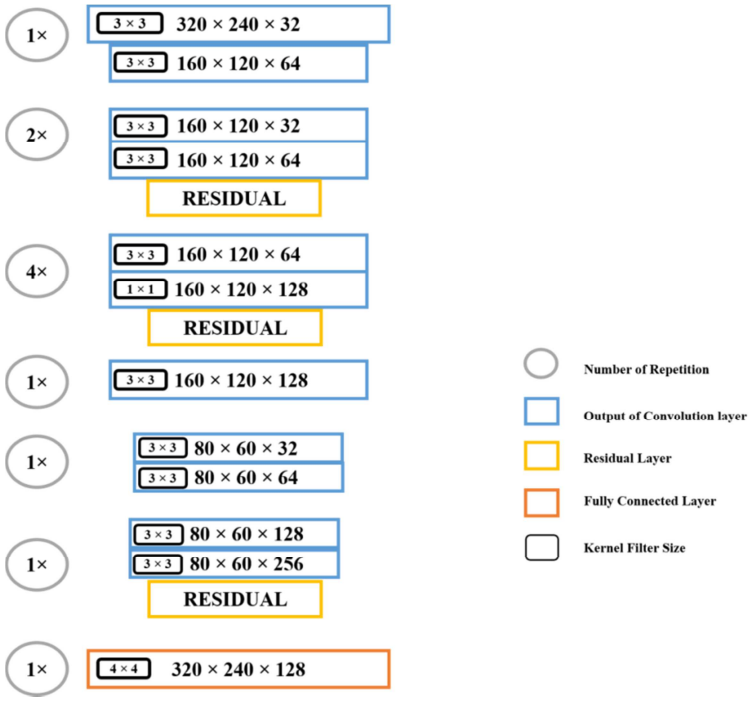

Fig. 4 Network architecture proposed for iris segmentation

4. Experiment

4.1. Dataset

4.1.1. CASIA v1.0

CASIA v1.0 [18] consists of 756 iris images acquired from 108 subjects with 320 × 280 resolution. The iris images are captured using a close-up camera. Eight 850 nm near-infrared (NIR) illuminators are arranged in a circular form around the sensor for adequate and uniform illumination.

4.1.2. CASIA v2.0

CASIA v2.0 [3] comprises 2400 images with 640 × 480 resolution. There are two subsets in CASIA v2.0. Each subset consists of 1200 images captured from sixty individuals using OKI IRISPASS-h and CASIA-IrisCamV2. All the images are captured in an indoor environment.

4.1.3. CASIA Interval

CASIA Interval [19] is a subset of CASIA Iris v3. This dataset comprises 2639 images captured from 249 subjects using a close-up camera with 320 × 280 resolution. The camera consists of NIR LED arrays arranged in a ring format to capture the iris images clearly.

4.1.4. PolyU Iris

PolyU Iris [20] consists of two subsets, namely the first and second sections. The first section comprises 12540 iris images obtained from 209 subjects with 640 × 480 resolution. In the second section, twelve subjects are involved, and their iris images are also available. NIR and visible range images are available in both the subsets.

4.1.5. IITD

The IITD dataset [14] consists of 2240 iris images from 224 subjects aged between fourteen and fifty-five. The images are with 320 × 240 resolution and are acquired using JIRIS, JPC1000, and a digital CMOS camera. Fig. 5 shows the sample images from the employed datasets.

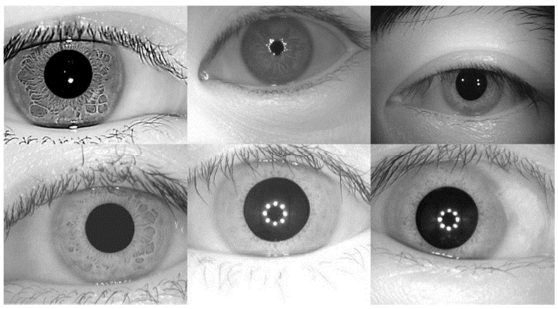

Fig. 5 Iris image samples from the datasets employed [3, 14, 18-20]

4.2. Training

The ground truth image for training the proposed network is obtained from WaveLab for CASIA Interval and IITD datasets. For CASIA v1.0, CASIA v2.0, and PolyU Iris databases, the ground truth images are created using the integrodifferential operator proposed by Daugman [5] and manually checked for proper segmentation. Manual operations for removing the remaining non-iris region are incorporated using the image labeler app in MATLAB. The iris and pupil boundary is located based on Eq. (5).

$$max_{(r,x_0,y_0)}\left|G_{\sigma}(r)\oplus\frac{\partial}{\partial x}\oint_{(r,x_0,y_0)}\frac{I(x,y)}{2\pi r}ds\right| \tag{5}$$

$I(x,y)$ is the input eye image. The operator looks for the maximum blurred partial derivative concerning an increase in the radius r of the normalized contour integral $I(x,y)$ over the image coordinate (x,y). The operator searches along the circular arc ds with the radius r and (x_0,y_0) as the center coordinate. $G_{\sigma}(r)$ is the Gaussian smoothing function with the scale σ. The whole operator acts as a circular edge detector. The path of the integral contour is defined based on searching the maximum derivative $max_{(r,x_0,y_0)}$ of the contour integration through r,x_0,y_0. Fig. 6 shows some sample iris images with their ground truth for the dataset employed in the proposed work.

This study uses K-fold cross-validation for training and validating the proposed framework. K-fold cross-validation is efficient and faster than most other cross-validation techniques as this type of validation repeats the train/test folds K times. Another advantage of the cross-validation technique is that it can reduce bias since the variation of the estimated results decreases as the K increases. In K-fold cross-validation, the datasets are divided into K sub-folds. To avoid the overlap between the training and testing sets, one sub-fold will be reserved as a test-fold while (K-1) folds will be used for training the network. The proposed work uses a five-fold cross-validation process. All datasets used are divided into five subsets containing an equal number of images. One is considered the testing set from the five subsets, while the remaining ones are used as the training data. The exact process is then carried out five times by considering each subset as the testing data exactly once.

The maximum iterative step is kept as 10000, the learning rate is set as 0.001, and the batch size is 4. The weights of every convolutional layer present in the pipeline are initialized with a standard deviation of 0.01. The bias is set as 0 for every convolutional layer. The learning rate and the momentum of the optimizer are kept as 0.01 and 0.9, correspondingly. At last, the saved network in 10000 steps is used for prediction. The proposed network is trained from scratch and does not employ any pre-trained models, and the total number of learnable parameters is 4.68M.

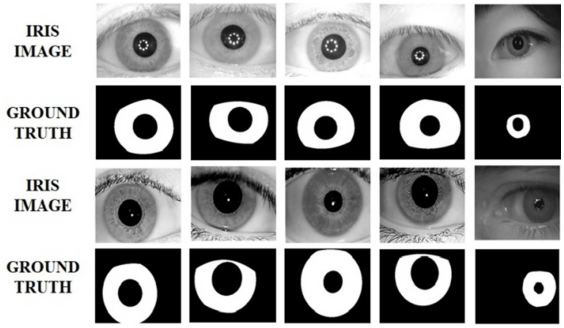

Fig. 6 Iris image samples with their ground truth images for the datasets employed [3, 14, 18-20]

4.3. Evaluation

The evaluation of the proposed work is based on its accuracy, sensitivity, specificity, precision, and F-score. The variables required for evaluation include true positive (t_p), true negative (t_n), false positive (f_p), and false negative (f_n). When an iris pixel is correctly segmented concerning its ground truth, t_p is registered; otherwise, f_n is registered. Similarly, correct segmentation of a non-iris pixel leads to t_n, and incorrect segmentation leads to f_p. The efficiency of the proposed pipeline is evaluated based on the equations given below.

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{6}$$

$$Sensitivity = \frac{t_p}{(t_p + f_n)} \tag{7}$$

$$Specificity = \frac{t_n}{(t_n + f_p)} \tag{8}$$

$$Precision = \frac{t_p}{t_p + f_p} \tag{9}$$

$$F\text{-}score = \frac{2(Precision \times Sensitivity)}{Precision + Sensitivity} \tag{10}$$

5. Experimental Results and Discussion

This section discusses the result obtained using the proposed methodology for iris segmentation with CASIA v1.0, CASIA v2.0, CASIA Interval, IITD, and PolyU Iris databases. The entire iris segmentation, normalization, and enhancement processes are carried out in the MATLAB 2020a platform on a 5 GHz quad-core computer with NVIDIA GeForce GTX 1080ti graphics processing unit. The input of the proposed network is a gray image with 320 × 240 dimensions and a channel of 3. The network gets trained, as explained in section 4.2. The network's output is a grayscale segmented map with values between 0 and 1, with the same number of channels and size as the input. The output binary mask obtained for some of the test images is shown in Fig. 7.

The validation accuracy graph of the proposed model is shown in Fig. 8. The accuracy obtained for CASIA v1.0, CASIA v2.0, CASIA Interval, IITD, and PolyU Iris is 0.82, 0.97, 0.9923, 0.9942, and 0.98, respectively. From Fig. 8, it can be inferred that the accuracy obtained for CASIA v1.0 is the lowest validation accuracy as its number of images available for training and validating the network is fewer than that of other iris databases. In addition, it is also found that except for CASIA v1.0, all other databases converge rapidly.

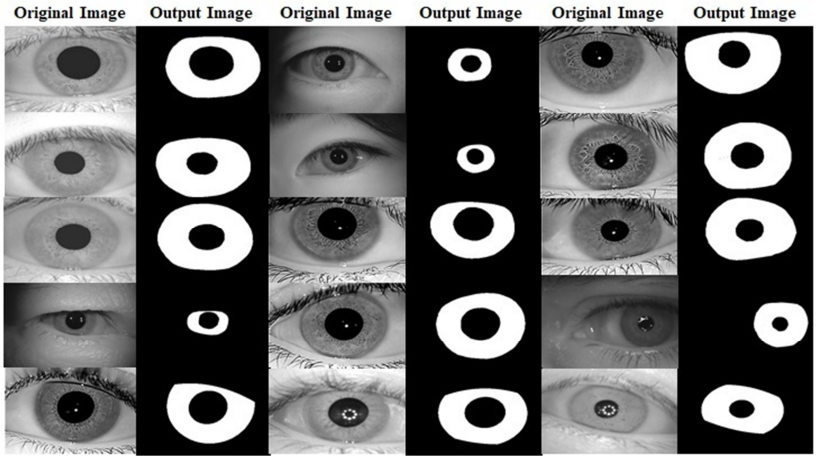

Fig. 7 Result obtained for iris image databases using the proposed architecture [3, 14, 18-20]

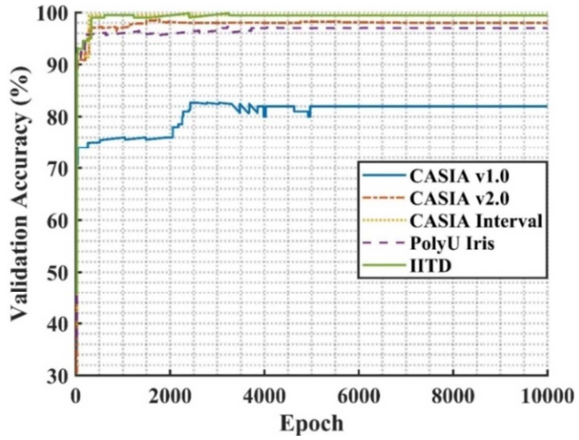

Fig. 8 Validation accuracy obtained for the employed datasets using the proposed architecture

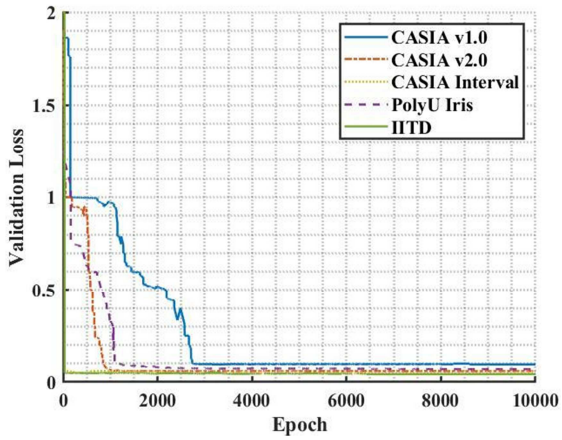

Fig. 9 Validation loss obtained for the employed datasets using the proposed architecture

Table 1 Comparison of the proposed network and other CNN algorithms

Dataset	Reference	Accuracy	Sensitivity	Specificity	Precision	F-score
CASIA v1.0	Proposed	0.82	0.79	0.74	0.63	0.77
CASIA v2.0	Proposed	0.97	0.92	0.99	0.99	0.97
CASIA Interval	Chen et al. [8]	0.9905	0.9828	-	0.9827	0.9828
	Jalilian et al. [21]	0.9609	-	-	-	0.9192
	Arsalan et al. [22]	0.9810	0.9710	-	0.9810	0.9758
	Uhl et al. [23]	0.74	-	-	-	0.8949
	Rathgeb et al. [24]	-	0.9768	-	0.8289	0.8927
	Varkarakis et al. [25] (CASIA Thousand)	0.995	0.9467	0.9986	0.9739	0.9594
	Hao et al. [26]	-	0.9822	-	0.9828	0.9824
	Proposed	0.9923	0.9830	0.9964	0.9840	0.9832
IITD	Chen et al. [8]	0.9884	0.9806	-	0.9818	0.9812
	Jalilian et al. [21]	0.9461	-	-	-	0.8892
	Arsalan et al. [22]	0.9716	0.9800	-	-	0.9756
	Rathgeb et al. [24]	-	0.9660	-	0.7887	0.8628
	Hao et al. [26]	-	0.9779	-	0.9796	0.9787
	Proposed	0.9942	0.9942	0.9804	0.9703	0.9805
PolyU Iris	Proposed	0.98	0.96	0.99	0.97	0.98

Fig. 9 represents the validation loss of the network for the employed dataset. From Fig. 9, it can be seen that the number of data used for training the network affects the validation loss. Since the amount of data for training the network using CASIA v1.0 is less, its validation loss is high compared to other datasets. The validation loss obtained for CASIA v1.0, CASIA v2.0, CASIA Interval, IITD, and PolyU Iris is 0.095, 0.0625, 0.0495, 0.0758, and 0.0498, respectively.

The effectiveness of the proposed CNN architecture is verified by comparing the CNN network with other algorithms, and the results are tabulated in Table 1. From Table 1, it can be observed that the accuracy of the proposed architecture is high. The proposed network shows higher efficiency than traditional iris segmentation algorithms such as adaptive Hough transform and contrast-adjusted Hough transform proposed by Uhl et al. [23] and Rathgeb et al. [24], respectively, regardless of accuracy, sensitivity, precision, and F1-score. Compared to the work of Chen et al. [8], the proposed network shows lower F-score and precision for CASIA Interval and IITD. Similarly, compared to the U-net architecture developed by Hao et al. [26], the proposed system obtains relatively low precision for IITD. Table 1 shows that the proposed CNN algorithm is more efficient than most of the existing iris segmentation algorithms.

6. Conclusions

This study proposed a 19-layer CNN-based iris segmentation pipeline. The predicted output (i.e., the mask) was obtained via a fully connected layer. To analyze the effectiveness of the network in segmentation, five publicly available iris databases with their ground truth mask are employed. For CASIA Interval and IITD databases, the ground truth masks are obtained from WaveLab. The ground truth masks for the rest of the iris databases were created using an integrodifferential operator algorithm. Some manual operations were performed on the received images to remove the non-iris regions. The accuracy, sensitivity, selectivity, precision, and F-score were employed to analyze the proposed network's performance. The obtained results were then compared with the existing traditional methods as well as the deep learning methods. Table 1 shows that the proposed network outperforms most of the existing algorithms.

The possible limitation of this method could be that the proposed CNN architecture is less flexible, i.e., it can only be used to segment the iris for the employed datasets. In the future study, the authors plan to develop a fully automated and end-to-end architecture for iris segmentation as the present work used a semi-manual technique based on the integrodifferential operator for labeling images and was not end-to-end. The accuracy, execution time, and optimization of the proposed network were also to be improved for real-time applications.

Acknowledgments

The authors are thankful to Indian Institute of Technology Delhi, New Delhi, India for providing the IITD iris image database. Also, the authors wish to extend their gratitude to Chinese Academy of Sciences' Institute of Automation (CASIA), Hong Kong Polytechnic University for granting free access to relevant iris databases, as well as WaveLab of the University of Salzburg for providing the Irisseg-ep database.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Statement of Ethical Approval

For this type of study, statement of human rights is not required.

Statement of Informed Consent

For this type of study, informed consent is not required.

References

[1] S. Z. Li, Encyclopedia of Biometrics: I-Z (Vol. 2), New York: Springer, 2009.

[2] R. Jillela, et al., "Methods for Iris Segmentation," Handbook of Iris Recognition, pp. 137-184, October 2013.

[3] "CASIA Version 2.0," http://biometrics.idealtest.org/downloadDB.do?id=2, May 2021.

[4] M. Mahlouji, et al., "Human Iris Segmentation for Iris Recognition in Unconstrained Environments," International Journal of Computer Science Issues, vol. 9, no. 1, pp. 149-155, January 2012.

[5] J. Daugman, "How Iris Recognition Works," The Essential Guide to Image Processing, pp. 715-739, January 2009.

[6] Z. Zhao, et al., "A Deep Learning Based Unified Framework to Detect, Segment and Recognize Irises Using Spatially Corresponding Features," Pattern Recognition, vol. 93, pp. 546-557, September 2019.

[7] W. Zhang, et al., "A Robust Iris Segmentation Scheme Based on Improved U-Net," IEEE Access, vol. 7, pp. 85082-85089, June 2019.

[8] Y. Chen, et al., "An Adaptive CNN's Technology for Robust Iris Segmentation," IEEE Access, vol. 7, pp. 64517-64532, May 2019.

[9] Y. H. Li, et al., "An Efficient and Robust Iris Segmentation Algorithm Using Deep Learning," Mobile Information Systems, vol. 2019, Article no. 4568929, January 2019.

[10] S. Bazrafkan, et al., "An End-to-End Deep Neural Network for Iris Segmentation in Unconstrained Scenarios," Neural Networks, vol. 106, pp. 79-95, October 2018.

[11] S. Lian, et al., "Attention Guided U-Net for Accurate Iris Segmentation," Journal of Visual Communication and Image Representation, vol. 56, pp. 296-304, October 2018.

[12] H. Hofbauer, et al., "Exploiting Superior CNN-Based Iris Segmentation for Better Recognition Accuracy," Pattern Recognition Letters, vol. 120, pp. 17-23, April 2019.

[13] M. Sardar, et al., "Iris Segmentation Using Interactive Deep Learning," IEEE Access, vol. 8, pp. 219322-219330, January 2020.

[14] "IITD Dataset," https://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Iris.htm, July 2020.

[15] K. He, et al., "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, June 2016.

[16] D. P. Kingma, et al., "Adam: A Method for Stochastic Optimization," https://arxiv.org/abs/1412.6980, December 2014.

[17] R. Li, et al., "Connection Sensitive Attention U-Net for Accurate Retinal Vessel Segmentation," https://arxiv.org/abs/1903.05558, March 2019.

[18] "CASIA Iris Dataset Version1," http://biometrics.idealtest.org/downloadDB.do?id=1, May 2021.

[19] "CASIA Interval Dataset," http://biometrics.idealtest.org/downloadDB.do?id=4, May 2021.

[20] "PolyU Iris Databases," https://www4.comp.polyu.edu.hk/~csajaykr/polyuiris.htm, July 2020.

[21] E. Jalilian, et al., "Iris Segmentation Using Fully Convolutional Encoder-Decoder Networks," Deep Learning for Biometrics, pp. 133-155, August 2017.

[22] M. Arsalan, et al., "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, Article no. 1501, May 2018.

[23] A. Uhl, et al., "Weighted Adaptive Hough and Ellipsopolar Transforms for Real-Time Iris Segmentation," 5th IAPR International Conference on Biometrics, pp. 283-290, March 2012.

[24] C. Rathgeb, et al., Iris Biometrics: From Segmentation to Template Security, New York: Springer, 2012.

[25] V. Varkarakis, et al., "Deep Neural Network and Data Augmentation Methodology for Off-Axis Iris Segmentation in Wearable Headsets," Neural Networks, vol. 121, pp. 101-121, January 2020.

[26] K. Hao, et al., "Iris Segmentation Using Feature Channel Optimization for Noisy Environments," Cognitive Computation, vol. 12, no. 6, pp. 1205-1216, 2020.