

Iris Recognition via Deep Learning Using Capsule Networks with Enhanced Routing Algorithm

Farzaneh Kuhifayegh¹, Roozbeh Rajabi^{2*}

¹ Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran

² DITEN Department, University of Genoa, Genoa, Italy

* Corresponding Author: roozbeh.rajabi@unige.it

Article Info

Article type:
Original Article

Article history:

Received 2024-10-08;

Revised 2024-16-11;

Accepted 2024-11-18.

How to cite this article:

Kuhifayegh, F. and Rajabi, R. (2025). Iris Recognition via Deep Learning Using Capsule Networks with Enhanced Routing Algorithm. *Sustainable Energy and Artificial Intelligence*, 1(2), 91-99.

DOI: 10.61186/seai.2410-1012

Abstract

Iris recognition is a widely used biometric technology in various applications. Deep learning methods, particularly convolutional neural networks (CNNs), have been popular in biometric detection due to their ability to generalize well and operate without human intervention. However, CNNs often struggle with image noise and require large datasets for effective training. To address these limitations, Capsule Networks (CapsuleNet) have been introduced, offering improved performance on small and noisy datasets. This paper presents a CapsuleNet architecture with an enhanced routing algorithm tailored for iris recognition. To further refine the learning process, VGG16 and InceptionV3 models are integrated into the CapsuleNet, enabling it to learn effectively from a limited number of samples. The proposed network is structured into a series of subnets, corresponding to its main building blocks. Experiments conducted on the CASIA-V4 Lamp iris dataset demonstrate that the optimized CapsuleNet architecture outperforms traditional networks, offering greater stability and robustness for iris recognitions. Results on the CASIA-V4 dataset show a 6% improvement in accuracy.

Keywords: Iris Recognition; Biometric Identification; Deep Learning; Capsule Network (CapsuleNet); Routing Algorithm.

Copyrights

© 2025 Licensee Hamedan University of Technology, Hamedan, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution –Non-Commercial 4.0 International (CC BY-NC 4.0) License (<http://creativecommons.org/licenses/by-nc/4.0/>).



1. Introduction

The biometric has seen tremendous activity over the past decades. Biometrics includes the ability of an automated program to learn the features, rules, and strategies used by the human cognitive system. Certain studies have concentrated on enhancing the performance of biometric systems. Iris recognition is one of the most important biometric detection technologies that is widely used in various fields. Iris recognition involves localization, normalization, feature extraction, and iris adaptation. Due to the complex visual structure and

high resolution for processing iris recognition, it is difficult for a human.

Iris detection is a new field and advances have been very rapid and effective. Machine learning techniques have been increasingly used in iris detection tasks in recent years [1]. The two main classes of iris recognition methods are neural networks and support vector machines (SVM). Recently, deep learning has become a widely-used technique in biometric cognition because of its automatic learning and strong generalizability [2]. Convolution neural networks (CNN) is a widely-used algorithm for iris recognition [3,4].

However, CNN needs a large number of samples to be trained successfully without the overfitting problem. Some techniques, such as data augmentation, change database size by generating new samples. However, they cannot address limited and noisy datasets. Moreover, the pooling process in CNN hides useful spatial information.

The capsule network [5,6] works with a vector neuron layer as opposed to a scalar neuron layer in CNN. In a capsule network, each neuron constructs a different property of the same feature. The spatial relationship information is learnt and stored on the whole parts and the local of the object. This characteristic enables the network to be trained using a small number of samples and obtaining good results. Zhao et al. [7] employed the capsule network architecture for iris recognition. They provided a modified routing algorithm based on the dynamic routing between two capsule layers. In this method, VGG16, InceptionV3, and ResNet50 were used as the transfer learning algorithms for model pretraining. Experiments conducted on the iris datasets of JluIrisV3.1, JluIrisV4, and CASIA-V4 Lamp showed the performance of the proposed model. The problem with the routing algorithms, such as RBA [5], in the capsule Nets is that the bias term has not been considered, leading to decreasing the probability of negative inputs occurrence.

In this paper, an iris recognition method using a capsule network is presented. Although the term bias has been added to the sum of all weighted inputs in neural networks, the bias has not been mentioned by the capsule network for routing algorithms. It has been proven that routing algorithms decrease the expressivity of capsule networks, preventing them from being universal approximators [8]. To address the limitations of capsule networks, the concept of bias in capsule network activators is introduced to deal with the negative weights created in routing networks and reduce its impact on the performance of the iris recognition systems. We design a capsule network architecture with an improved routing algorithm to enhance the performance of iris recognition. The proposed method achieves an accuracy of over 96% on the CASIA-V4 dataset.

The rest of the paper is organized as follows: Section 2 briefly summarizes related works on deep learning-enabled iris recognition. In Section 3, a capsule network architecture using an improved routing algorithm is developed. Section 4 provides efficiency evaluation results, and then conclusions are deduced in Section 5.

2. Related Works

Several deep networks have been proposed to

improve iris recognition's performance. The DeepIris network [9] consists of 9 layers, each with a bilateral filter layer, one convolution layer, two pooling layers, two normalization layers, and one fully-connected layer. This deep network performed well on the CASIA and Q-FIRE datasets. Two iris recognition networks were proposed in [10]. The first network, called DeepIrisNet-A, consists of eight convolution layers (each followed by a batch normalization layer), four pooling layers, three fully-connected layers, and two output layers. The second network, called DeepIrisNet-B, added two primary layers to enhance modeling capabilities. Two datasets of ND-IRIS-0405 and ND-CrossSensor-Iris-2013 were used to evaluate the networks. Minaei et al. [11] showed that the VGG model, as a transfer learning technique, can be performed well for the iris recognition task.

In [3], a CNN scheme based on a combination of backpropagation algorithm and batch optimization method for improving weight and learning rate is presented. Besides, other training strategies were proposed to evaluate various CNN architectures. The performance of the proposed system was evaluated on three general datasets collected under different conditions: SDUMLA-HMT, CASIA-Iris-V3 Interval, and IITD. A deep learning-based framework for iris detection, segmentation, and recognition was proposed [12]. The proposed framework uses an R-CNN for iris recognition and an optimized convolution network to produce the iris feature, and an extended cost function. According to the results, the recognition rate was 98.8%.

Iris images recorded in non-participatory environments often suffer from undesirable noise, which makes challenges for iris segmentation methods. To solve this problem, an iris segmentation approach based on deep learning, called IrisParseNet, was proposed [13]. Unlike many CNN-based iris segmentation methods that focus only on iris prediction following semantic division frameworks, the proposed method was a complete iris segmentation solution. In addition, a module was included to improve segmentation performance.

In [7], a deep learning method based on capsule network architecture in iris recognition was proposed. A modified routing algorithm based on dynamic routing between capsule layers was proposed to make the method compatible with iris recognition. Learning was well adapted even if the number of samples is limited. Three models of VGG16, InceptionV3, and ResNet50 were used for feature extraction. Instead of a convolution layer in

the capsule network, these networks are used as a convolution to extract the main features. Three iris datasets, JluIrisV3.1, JluIrisV4 and CASIA-V4 Lamp, were used for the experiments.

3. Proposed Methodology

In this section, an iris recognition method using capsule networks is presented. To address the limitations of capsule networks, the concept of bias in capsule network routing algorithm is introduced to deal with the hostile weights created in routing networks and reduce its impact on the performance of the iris recognition system. The proposed framework is shown in Fig. 1 that consists of four main parts: (1) pre-processing of the iris images, (2) deep network based on the capsule network, (3) classification, and (4) image reconstruction.

3-1. Iris Images Preprocessing

The preprocessing phase is divided into three parts: localization, normalization, and enhancement.

A) Localization: First, through quality assessment, the iris tissue part of the main iris image (which has dimensions of 640×480) is located and segmented. The Region of Interest (ROI) is then extracted from the located iris. To do this, the original black and white image is first converted to a color image using the `cvtColor` method from the OpenCV library. Noise is then removed from the images using a two-way filtering technique. The pupil is then calculated in a 120×120 dimension and the edges are identified by the Canny technique.

B) Normalization: At this stage, the ROI image with dimensions of 326×256 are normalized. First, the center of the cropped image is found and then the centers between the inner border and the outer

border are determined. The obtained coordinates are then converted to the normalized value.

C) Enhancement: To standardize the input size, which has a great impact on training as an important parameter in the network, the iris images normalized by the nearest neighbor algorithm are changed to 197×197 as input. The histogram smoothing technique is used to improve the contrast in an image to increase the intensity range.

3-2. Pre-Training Models

The main task of the convolution part is to extract the low-level features from the pixel intensity of the input images and to form the main features used for the primary capsule layer. The convolution part of the capsule networks consists of a convolution layer with 32 channels with dimensions of 9×9 and step 1 (Stride = 1), which makes the input form with dimensions (1, 28, 28) to the output with dimensions (256, 20, 20) change. In the proposed method, two strategies are used to build the convolution part: (a) The transfer learning method is applied and three advanced networks are used, including VGG16, ResNet50 and InceptionV3, which have been trained using the ImageNet dataset. To find the most suitable features, the whole network is divided into sub-networks according to the number of blocks that make them up, then the sub-networks are connected through the vector part and thus a set of different network examples is set up. Therefore, the output of different levels in the network is obtained as a result of convolution to find the most compatible convolution structure for the whole network through experiments. Table 1 shows the learned models used in the convolution part of the proposed model.

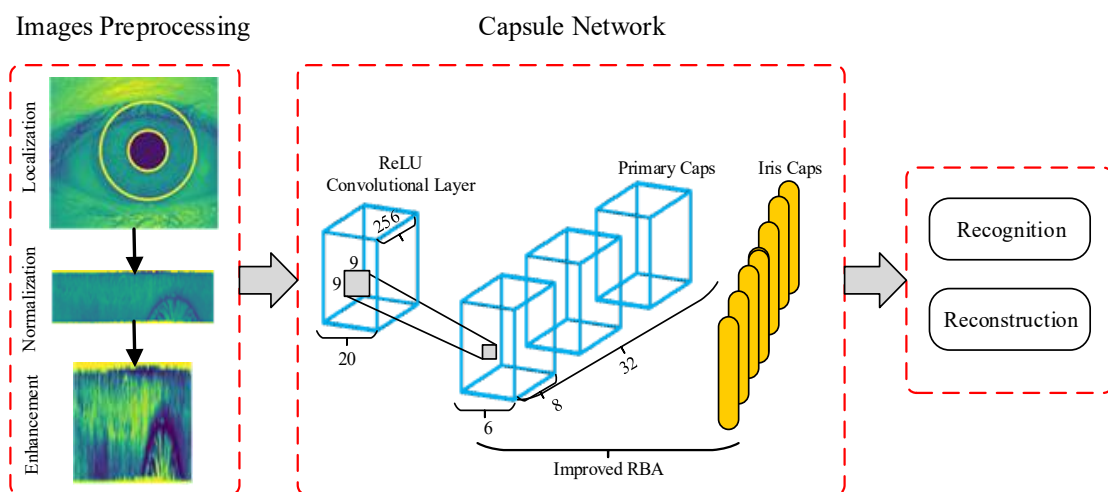


Fig. 1. Proposed Methodology

Table 1. Transfer learning used in convolution part

-Model	-Type	-Blocks	-Depth
-VGG16	-Cannulational block	-5	-23
-ResNet50	-Residual block	-16	-168
-InceptionV3	-Inception blok	-11	-159

We create several shallow convolution networks as part of the convolution without pre-training. These networks include Iris-Dense with two dense blocks, Iris-Inception with one inception block, and Iris-SE with two Squeeze-and-Excitation blocks.

3-3. Routing Algorithm in the Convolution Layer

Capsules in capsule networks are groups of neural cells that represent an object or part of an object in a decomposition tree, introduced by Hinton et al. [14]. A capsule encodes the sampling parameters of an object (such as position, location, direction, etc.) through vector representations. This activation vector is called a capsule and encrypts the possibility that the object is present in the current input. The CapsNet architecture introduced in [5] builds early capsules using initial layers with ReLU convulsions. The initial capsules are then fed to the next layers of the capsule by routing algorithms. The routing-by-agreement (RBA) [5] is one of these techniques that uses low-level capsule predictions to produce high-level capsules. The expected maximum routing algorithm [15], known as EM-routing, is another routing method that uses a status matrix to calculate equal high-level capsules.

Although the two algorithms RBA and EM-routing can connect the capsule in different layers with an iterative routing method, they reduce the ability to represent the capsule networks to symmetric functions. There is a negative value for each input to a capsule layer that cannot be distinguished by capsules, even when this input represents a different class. This restriction has a negative effect on the training of deep capsule networks for datasets because the initial capsules are made with ReLUs, which means that all the components of the primary capsules are positive. Therefore, the negative of a vector and the vector itself can never be the input of the primary capsule layer. After the first capsule layer, routing algorithms are used to generate subsequent capsules, and negative inputs can actually occur. Subsequent layers increase the likelihood of negative inputs, and therefore these routing algorithms negatively affect the training of deep architectures.

In the proposed method, the RBA algorithm is optimized to deal with the above-mentioned constraint. The term $i \in 1, 2, \dots, I$ denotes the number of low-level capsules and $j \in 1, 2, \dots, J$ denotes the number of high-level capsules. The u input for RBA is called the positive input and $-u$ is called the negative input vector. For each variable v in the algorithm, the variable constructed using u is denoted by v^+ and the variable constructed using $-u$ is denoted by v^- and. The term v_{ij} refers to a variable used for low-level capsule i and high-level capsule j . it has been proven that the input activators and the corresponding negative inputs are equal if the RBA routing algorithm is used [8]. The RBA and its parameters are shown in Algorithm 1. The inputs of this routing algorithm are low-level predictions ($\bar{u}_{j|i} = W_{ij} u_i$) where u_i is the activation of the low-level capsule i and W_{ij} is the transfer matrix that is learned by going back.

Algorithm 1. Routing-By-Agreement (RBA)

```

- Input:
- Output:
- 1: Begin
- 2:  $\forall b_{ij}, \mathbf{b}_{ij} \leftarrow 0$ 
- 3: For  $r$  iterations Do
- 4:  $c_{ij} \leftarrow \frac{\exp(b_{ij})}{\sum_l \exp(b_{il})}$ 
- 5:  $s_j \leftarrow \sum_i c_{ij} \bar{u}_{j|i}$ 
- 4:  $\mathbf{v}_j \leftarrow \frac{\|s_j\|^2}{1 + \|s_j\|^2} \times \frac{s_j}{\|s_j\|}$ 
- 4:  $\mathbf{b}_{ij} \leftarrow \mathbf{b}_{ij} + \mathbf{v}_j \cdot \bar{u}_{j|i}$ 
- 6: End For
- 7: End

```

To address this challenge, the term bias is added to the pre-activators of the RBA algorithm. In the modified algorithm, how to add the bias to the preactivator calculations is shown in Equation (1):

$$s_j \leftarrow \left(\sum_i c_{ij} \bar{u}_{j|i} \right) + bias_j \quad (1)$$

Then, a network can learn $bias \neq \emptyset$:

$$\begin{aligned}
\bar{s}_j &= \left(\sum_i \bar{c}_{ij} \bar{u}_{j|i} \right) + bias_j \\
&= \left(\sum_i \bar{c}_{ij} (-\bar{u}_{j|i}^+) \right) + bias_j \\
&= - \left(\sum_i \bar{c}_{ij} u_{j|i}^+ \right) + bias_j \\
&\neq - \left(\left(\sum_i \bar{c}_{ij} u_{j|i}^+ \right) + bias_j \right) = -s_j^+
\end{aligned}$$

This method enables the network to learn non-zero bias parameters so that the calculated pre-activator vector is different for their inputs and

negative inputs. This leads to the differentiation of activation vectors and activation vectors. Thus, the network can learn the bias so that negative inputs and inputs can be detected.

3-4. Network Architecture

The network input is a 197×197 grey image. First, the channel is changed from 1 to 3 using a 1×1 convolution layer. Then, a feature with the size $197 \times 197 \times 3$ is fed into the convolution network, which is the InceptionV3 network. The number of layers in the convolution can vary. In this work, the pre-trained InceptionV3 model with 4 blocks is applied to a feature map with the size of 10×10 and a channel with the size of 768.

The tensor shape is converted to 128×600 by normalizing and transforming the vector. Finally, the feature vector is 32×51 (51 indicates the number of batches and 32 is the size of each capsule). The network output is normalized using L2 normalization, which indicates the probability of each category.

3-5. Computational Complexity

Adding a bias is an element-wise addition operation that requires $O(d)$, for a capsule vector with dimensionality d . Compared to other operations, including matrix multiplications and iterative routing, this addition introduces negligible computational overhead and will not significantly change the overall computational cost of the network.

4. Evaluation Results

In this section, we investigate the validity of the proposed method on the CASIA-V4 Lamp iris dataset. First the characteristics of the iris image dataset are described and then the experimental parameters are explained. In the next step, the performance of several network structures consisting of convolution layers and vectors is examined. Then, the performance of the proposed method with different structures is compared with similar methods presented using CNN algorithms [4] and the base capsule network [7]. Equivalent accuracy and Equal Error Rate (EER) criteria are used to evaluate models.

4-1. Dataset Description

The CASIA-Iris image database has been published for biometric research and updated since

2002 from CASIA-IrisV1 to CASIA-IrisV3. More than 3,000 users from 70 countries have downloaded the CASIA-Iris database. These versions of the dataset have a medium scale of data, while in real applications iris recognition systems have a large scale. The CASIA-IrisV4 dataset was presented to enhance research in the field of large-scale iris detection systems.

The CASIA-Iris-Lamp set was collected using a hand iris sensor. To introduce further changes, a lamp close to the device was turning on and off. Elastic changes in iris tissue are one of the most common and challenging issues in iris recognition due to dilation and contraction of the pupil in different lighting conditions. Therefore, CASIA-Iris-Lamp is suitable for studying real applications. The CASIA-V4 Lamp iris dataset contains 16,215 iris images from 822 subjects. Each person has images from the left and right eye angles with a resolution of 440×640 . Left and right iris images are a subject of different classifications in the experiment.

4-2. Experiments Settings

After the preprocessing, the iris images are divided into two parts, training and testing. In the experiments of this study, random sampling method was used to select images. About 4,500 images were selected for the training phase and 1,500 images for the testing phase. The number of epochs of training is 50 and the learning rate (LR) varies in different experiments. Due to the high volume of processes and the need for high resources, especially main memory, the Colab platform is used to implement the experiments. The Colab environment is a product of Google Research. This environment provides free access to the computing resources, including the GPU. Table 2 shows the implementation environment specifications in Colab.

4-3. Structure Analysis

In this section, we carry out a series of experiments on the CASIA-V4 dataset through various network structures. These networks are configured using VGG16 and InceptionV3 models. Table 3 shows the performance of the capsule network with a different number of blocks and routing algorithms, including DRDL, Dynamic, RBA and IRBA, for the features trained by the VGG16 network. The learning rate in all models is set to 0.0001 and also two criteria of accuracy and square error rate are used to compare the models. The obtained results

show that the use of the improved RBA algorithm (IRBA) in the capsule network increases the accuracy of the recognition model and reduces the amount of error. Better results are obtained for models with four blocks than that of five blocks. The proposed method has the highest performance for the four blocks, with an accuracy value of 0.963 and an error rate value of 1.241.

Table 2. Specifications of the implementation environment

General		Type
Characteristic Platform		Linux
Programming language		Python
Parallel programming RAM	CUDA Version: 10.1	13 GB
GPU		
GPU Model	Tesla P100-PCIE-	16GB
Maximum Threads Per Block		1024
Maximum Blocks		1024*64=65536
Maximum Shared Memory Per Block		49152
		16 GB
CPU		
CPU Model	Intel(R) Xeon(R)	CPU @ 2.20GHz
CPU(s)		2
Thread(s) per core		2
L cache		32K
L2 cache		256K
L3 cache		56320K

Fig. 2a shows the effect of the number of epochs on the performance accuracy of iris recognition models. This experiment was performed using the features obtained from the VGG16 model with 4 and 5 blocks. It can be seen that the deep learning models have reached maximum accuracy before the 40th iteration. The highest training accuracy is

related to the four-block model, which is about 0.99. On the other hand, the lowest accuracy is related to the capsule model with DRDL routing and the block number of 5 (0.947). Fig. 2b shows the capsule network training error using the VGG16 model with 4 and 5 blocks. Like the training accuracy, the training error in the 30th iteration reaches the lowest possible level for the models so that the proposed model with four blocks has the lowest error rate (0.06). In contrast, the 5-block capsule network with the dynamic routing algorithm has the highest error rate, about 0.18.

Table 3. Comparison of different networks for CASIA-V4 dataset with VGG16 network

Architecture	Learning rate	Accuracy (%)	EER (%)
VGG16_4blocks+DRDL	94.32	94.32	1.684
VGG16_4blocks+Dynamic	95.63	95.63	1.874
VGG16_4blocks+RBA	93.84	93.84	1.632
VGG16_4blocks+IRBA (Proposed)	96.3	96.3	1.241
VGG16_5blocks+DRDL	90.2	90.2	2.3415
VGG16_5blocks+Dynamic	91.24	91.24	2.145
VGG16_5blocks+RBA	87.5	87.5	2.547
VGG16_5blocks+IRBA (Proposed)	93.14	93.14	2.136

Table 4 shows the modeling results of the features taught by the InceptionV3 model with four routing algorithms DRDL, Dynamic, RBA and IRBA with a learning rate of 0.0001. The number of blocks examined in this experiment is between 1 and 5. As with the results for the VGG16, the four-block models performed best in terms of accuracy and error rate. The average accuracy for the proposed strategy (IRBA) is 84.5%, which is about 8% better than the original version (RBA), which is 86.8% on average.

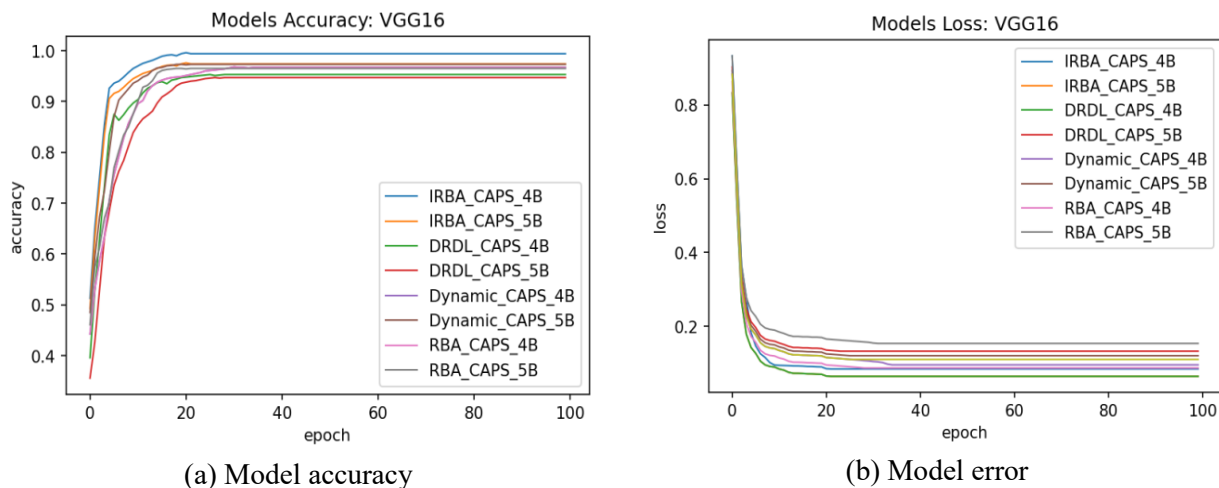


Fig. 2. Training performance for CASIA-V4 dataset using VGG16 feature extraction model

Table 4. Comparison of different networks for CASIA-V4 dataset with InceptionV3 network (Learning rate=0.0001)

Architecture	Accuracy (%)	EER (%)
InceptionV3_1block+DRDL	63.47	4.21
InceptionV3_1block+Dynamic	59.08	5.32
InceptionV3_1blockRBA	57.62	6.87
InceptionV3_1block+IRBA (Proposed)	64.31	4.11
InceptionV3_2blocks+DRDL	84.31	3.42
InceptionV3_2blocks+Dynamic	81.24	4.21
InceptionV3_2blocks+RBA	75.88	5.08
InceptionV3_2blocks+IRBA (Proposed)	86.74	3.12
InceptionV3_3blocks+DRDL	86.58	3.25
InceptionV3_3blocks+Dynamic	84.66	3.66
InceptionV3_3blocks+RBA	78.22	4.54
InceptionV3_3blocks+IRBA (Proposed)	89.14	2.84
InceptionV3_4blocks+DRDL	86.25	1.51
InceptionV3_4blocks+Dynamic	85.75	2.31
InceptionV3_4blocks+RBA	80.41	1.98
InceptionV3_4blocks+IRBA (Proposed)	87.3	1.85
InceptionV3_5blocks+DRDL	91.07	2.24
InceptionV3_5blocks+Dynamic	86.47	2.86
InceptionV3_5blocks+RBA	88.14	3.45
InceptionV3_5blocks+IRBA (Proposed)	93.11	2.33
InceptionV3_6blocks+DRDL	86.41	4.12
InceptionV3_6blocks+Dynamic	85.36	4.63
InceptionV3_6blocks+RBA	81.2	3.14
InceptionV3_6blocks+IRBA (Proposed)	88.34	2.58

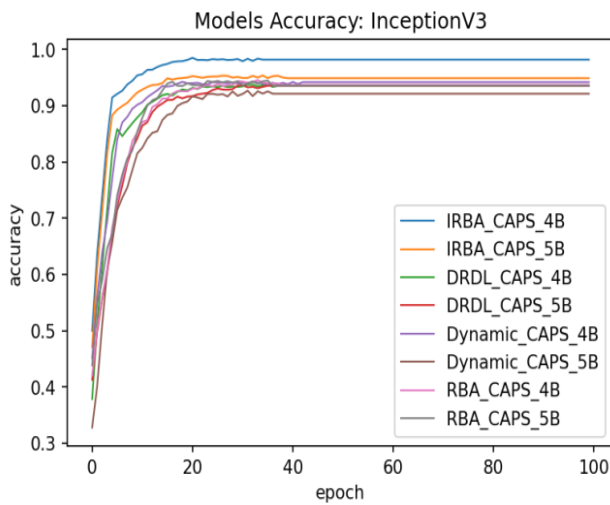
Fig. 3 shows the performance of iris detection models in terms of training accuracy and loss value for 100 epochs. The features of the model were generated using the InceptionV3 model. The results show that the models converge in a higher number of iterations than the VGG16 based models. The four-block capsule network has the best performance, followed by the five-block network. Capsule networks with IRBA routing algorithms perform best in terms of training error and accuracy. Dynamic algorithm-based networks also had the lowest performance and provided the highest training error.

4-4. Performance Comparison with State-of-the-Art Studies

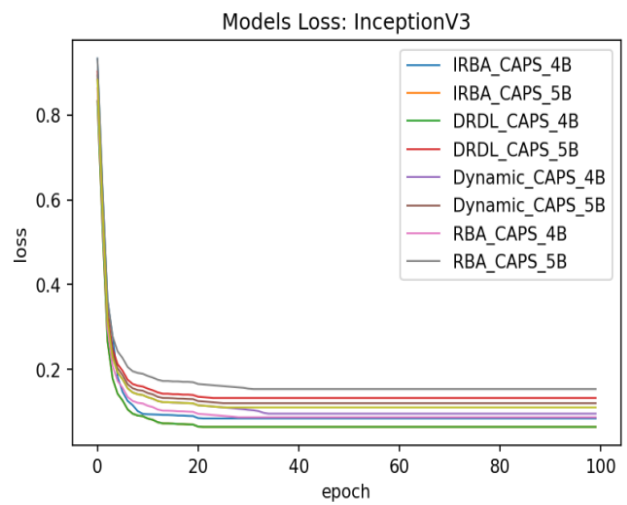
The comparison of our proposed architecture with other well-known deep learning models for iris recognition, including CNN-based [4] and Basic CapsuleNet [7], in terms of accuracy and EER is shown in Table 5. Fig. 4 compares the DL models' training performance in terms of accuracy and loss value in comparison with two other networks. As we can see, our model achieved the highest accuracy and the lowest training loss.

Table 5. Comparison of our proposed model compared with the other models

Architecture	Accuracy (%)	EER (%)
CNN-based [4]	89.35	3.015
Basic CapsuleNet [7]	91.24	2.635
Our Proposed Method	93.14	2.136



(a) Model accuracy



(b) Model error

Fig. 3. Training performance for CASIA-V4 dataset using InceptionV3 feature extraction model

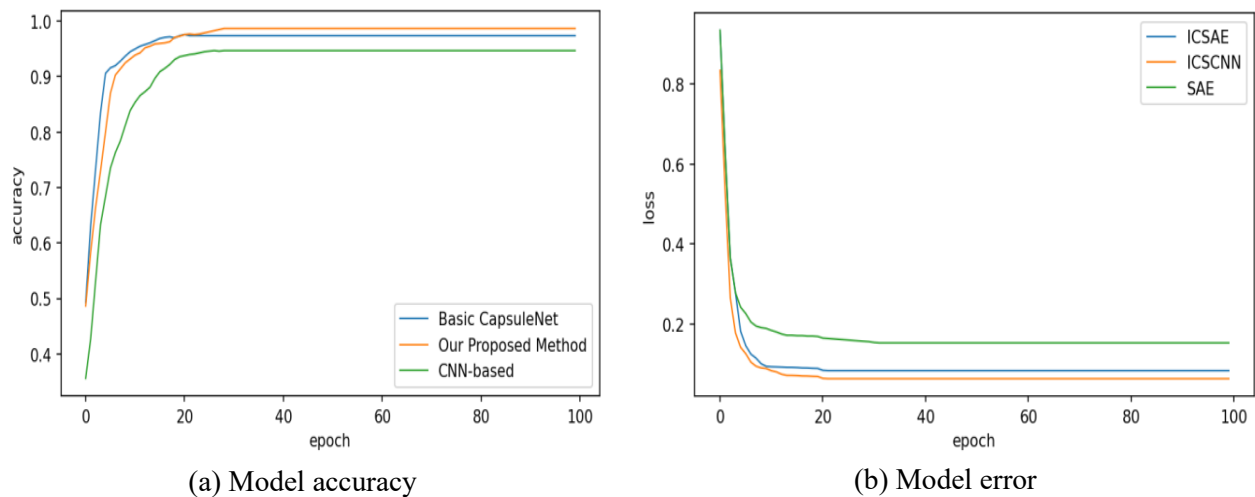


Fig. 4. Training performance for our proposed model compared with the other models

5. Conclusion and Future Works

In this study, a capsule network method using a routing algorithm for iris detection is introduced. From several capsule architectures with different depths according to the outputs of different pre-trained networks. The Routing Algorithm (RBA) was also improved by considering the bias in the routing algorithm. Several networks with different structures and depths were created and investigated using the CASIA-Iris iris dataset. Experiments have shown that a deep network with a capsule architecture can perform well in iris recognition. The designed models were evaluated on the CASIA-Iris dataset and the performance of the models was evaluated in terms of accuracy and detection error. VGG16 and InceptionV3 models were used for feature pre-training. The average accuracy for the proposed model using the VGG16 model was about 94.7% and the mean error rate was 1.241. This value was about 93.2% and 1.136 for the proposed model using the InceptionV3 model. Structures with very high or very shallow depths in different networks sometimes cause the model to learn the appropriate properties. In addition, the performance of the network with the structure of the pre-prepared model is a stable part that uses the complete training structure. For future work, multitask learning can be helpful in iris recognition because data tagging is often the most difficult and time-consuming process in building a classifier. Moreover, to increase the efficiency of the iris recognition systems on the large-scaled datasets, parallel processing, such as Hadoop and GPU, can be used. Furthermore, evolutionary algorithms, such as the genetic and ant colony, can be employed to regulate network parameters as well as model architecture.

References

- [1] Nguyen, K., Proença, H., & Alonso-Fernandez, F. (2024). Deep learning for iris recognition: A survey. *ACM Computing Surveys*, 56(9), 1-35.
- [2] Nguyen, K., Fookes, C., Ross, A., & Sridharan, S. (2017). Iris recognition with off-the-shelf CNN features: A deep learning perspective. *Ieee Access*, 6, 18848-18855.
- [3] Al-Waisy, A. S., Qahwaji, R., Ipson, S., Al-Fahdawi, S., & Nagem, T. A. (2018). A multi-biometric iris recognition system based on a deep learning approach. *Pattern Analysis and Applications*, 21, 783-802.
- [4] Alaslani, M. G. (2018). Convolutional neural network based feature extraction for iris recognition. *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, 10.
- [5] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 3859-3869.
- [6] Choudhary, S., Saurav, S., Saini, R., & Singh, S. (2023). Capsule networks for computer vision applications: a comprehensive review. *Applied Intelligence*, 53(19), 21799-21826.
- [7] Zhao, T., Liu, Y., Huo, G., & Zhu, X. (2019). A deep learning iris recognition method based on capsule network architecture. *IEEE Access*, 7, 49691-49701.
- [8] Peer, D., Stabinger, S., & Rodriguez-Sanchez, A. (2021). Limitation of capsule networks. *Pattern Recognition Letters*, 144, 68-74.
- [9] Liu, N., Zhang, M., Li, H., Sun, Z., & Tan, T. (2016). DeepIris: Learning pairwise filter bank for heterogeneous iris verification. *Pattern Recognition Letters*, 82, 154-161.
- [10] Gangwar, A., & Joshi, A. (2016, September). DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition. In *2016 IEEE international conference on image processing (ICIP)* (pp. 2301-2305). IEEE.

- [11] Minaee, S., Abdolrashidiy, A., & 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.
- [12] Zhao, Z., & Kumar, A. (2019). A deep learning based unified framework to detect, segment and recognize irises using spatially corresponding features. *Pattern Recognition*, 93, 546-557.
- [13] Wang, C., Muhammad, J., Wang, Y., He, Z., & Sun, Z. (2020). Towards complete and accurate iris segmentation using deep multi-task attention network for non-cooperative iris recognition. *IEEE Transactions on information forensics and security*, 15, 2944-2959.
- [14] Hinton, G. E., Krizhevsky, A., & Wang, S. D. (2011). Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21* (pp. 44-51). Springer Berlin Heidelberg.
- [15] Hinton, G. E., Sabour, S., & Frosst, N. (2018, February). Matrix capsules with EM routing. In *International conference on learning representations*.

Biography



Roozbeh Rajabi received his B.Sc. degree in Electrical Engineering from the Iran University of Science and Technology, Tehran, Iran, in 2007, and his M.Sc. degree in Biomedical Engineering (Bioelectric) from Tarbiat Modares University, Tehran, Iran, in 2009. He completed his Ph.D. in Communication Systems Engineering at Tarbiat Modares University in 2014. He is currently an Assistant Professor at the University of Genoa (UniGe), Genoa, Italy. His research interests include hyperspectral data analysis, pattern recognition, and digital signal and image processing. He is a Senior Member of IEEE.
