Received February 20, 2019, accepted March 25, 2019, date of publication April 12, 2019, date of current version April 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2911056

A Deep Learning Iris Recognition Method Based on Capsule Network Architecture

TIANMING ZHAO[1], YUANNING LIU[1], GUANG HUO[2], AND XIAODONG ZHU[1]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]School of Computer Science, Northeast Electric Power University, Jilin 132012, China

Corresponding author: Xiaodong Zhu (zhuxd@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61471181, in part by the Natural Science Foundation of Jilin Province under Grant 20140101194JC and Grant 20150101056JC, in part by the Industrial Technology Research and Development Foundation of Jilin Province under Grant 2019C053-2, and in part by the Science and Technology Project of the Jilin Provincial Education Department under Grant JJKH20180448KJ.

ABSTRACT Iris recognition is one of the most representative identification technologies in biometric recognition, which is widely used in various fields. Recently, many deep learning methods have been used in biometric recognition, owing to their advantages such as automatic learning, high accuracy, and strong generalization ability. The deep convolutional neural network (CNN) is the mainstream method of image processing widely used in many domains, but it has poor anti-noise capacity in image classification and is easily affected by slight disturbances. CNN also needs a large number of samples for training. The recent capsule network not only has high recognition accuracy in classification tasks but can also learn part-whole relationships, increasing the robustness of the model. Furthermore, it can be trained using a small number of samples. In this paper, we propose a deep learning method based on the capsule network architecture in iris recognition. The structure detail of the network is adjusted, and we provide a modified routing algorithm based on the dynamic routing between two capsule layers to make this technique adapt to iris recognition. Migration learning makes the deep learning method available even when the number of samples is limited. Therefore, three state-of-the-art pretrained models, VGG16, InceptionV3, and ResNet50, are introduced. We divide the three networks into a series of subnetwork structures according to the number of their major constituent blocks. They are used as the convolutional part to extract primary features, instead of a single convolutional layer in the capsule network. Our experiments are conducted on three iris datasets, JluIrisV3.1, JluIrisV4, and CASIA-V4 Lamp, to analyze the performance of different network structures. We also test the proposed networks in simulated strong and weak light environments, showing that the networks with capsule architecture are more stable than those without.

INDEX TERMS Iris recognition, deep learning, capsule network, transfer learning.

I. INTRODUCTION

Biometric recognition has played an irreplaceable role in personal identification application in recent years. Given the desirable properties such as uniqueness, stability, and noninvasiveness, iris recognition has better prospects in high-precision recognition compared to other biometric modalities [1]. The first iris recognition system was proposed by Daugman in 1993 [2], using a multiscale 2D-Gabor filter to extract the binary phase encoding features of an iris image from multiorientations and employing Hamming distance matching. After over decades of comparative research, many iris recognition methods have been introduced to enhance the reliability and usability [3]–[6].

In recent years, with the vigorous development of the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [7], deep learning networks, especially CNNs, have shown obvious improvement in the performance of computer vision tasks such as image classification, single object localization, and object detection. Meanwhile, the study of deep learning methods in iris recognition has shown promising prospects [27]–[40].

However, the training of CNNs relies on a large number of samples. The model will be affected by the overfitting with insufficient training data. The data augmentation method usually applied to CNNs cannot solve the problem of small

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du.

samples in essence. In addition, the pooling layers in CNNs (especially in higher layers) lead to a loss of some valuable spatial information, and this loss of information makes the networks particularly sensitive to microinterference.

As a result, we introduce the capsule network to alleviate the shortcomings of using CNNs in an iris dataset of small sample size. The capsule network, which is very different from other deep networks, was proposed by Sabour *et al.* [8] in 2016. The capsule network has a vector neuron layer instead of a scalar neuron layer and a dynamic iterative computing pattern in addition to the back propagation. Most prominently, the network can learn and store the spatial relationship information on the whole parts and the local of the object. Because of this ability, the number of samples needed for training a capsule network is much smaller than that of CNNs, and good results can be obtained without data enhancement, which is proved in [8]. Moreover, the anti-noise ability of the model has been substantially improved. Furthermore, it shows the following characteristics: global features learning, equivariance mapping, robust training, confidence assessment, and overlapping objects detection. Therefore, the capsule network has a potential role in iris recognition, especially under some specific environments, but the research of capsule network in this field is rare.

In this paper, a deep network learning method based on the capsule network is proposed for iris recognition. We first construct the convolutional part with different architectures, including those with shallow convolution networks requiring full training and three state-of-the-art pretrained networks to replace the 9×9 convolution layer in the capsule network. Then, in order to find a more suitable convolution structure, we divide the pretrained models into several subnetworks according to their respective major component blocks. Next, we use the convolution capsule to replace the full connection capsule to reduce the number of parameters. Finally, aiming to make it easier for the network to converge on different datasets, we propose another routing algorithm based on the dynamic routing, which makes more effective use of the information on vector direction and vector length. The experiments are conducted on three iris datasets: the JluIrisV3.1, the JluIrisV4 [9], and the CASIA-V4 Lamp [10]. We compare and explore the impact of networks with a variety of structural combinations on the results. To prove that the network with capsule structure has strong robustness under certain circumstances in keeping recognition accuracy, we design an iris recognition test experiment under simulated strong and weak light environments.

The remainder of this article is organized as follows: The previous related work is presented in Section 2. Section 3 describes the processes of the proposed scheme. Section 4 validates the scheme by a series of experimental results. Section 5 presents the conclusion.

II. RELATED WORK

With the arrival of the era of big data and the improvement of hardware computing capability, the deep learning method has achieved excellent results in many fields, breaking through the limitations of traditional pattern recognition and machine learning methods. Prominently, the deep learning method helps avoid the problems caused by previous empirical or artificially defined feature extraction methods by learning the mapping relationship between input images and category labels. The establishment of the ImageNet project has considerably promoted the development of deep learning as well. This large visual database is used mostly for visual object recognition software research, with over 14 million image URLs being manually annotated to indicate the objects in the pictures. Many excellent deep networks have been developed and become the state-of-the-art methods in many fields, including AlexNet [11], VGG [12], ResNet [13], DenseNet [14], Google LeNet [15], and other Inception structural versions [16], [17]. In addition to the popular deep networks, there are many other interesting networks emerging, such as the SqueezeNet [18], Squeeze-and-Excitation Networks (SENet) [19], and CliqueNet [20].

Owing to the advantages of deep learning technology, the relevant research and application are constantly arising in iris recognition. In 2016, Liu *et al.* [21] proposed DeepIris to solve the problem of iris recognition using a deep learning model for the first time. They used CNN model and constructed a pairwise filter bank (PFB) to learn a pair of input image similarity for the heterogeneous iris problem. Gangwar and Joshi [22] developed a deep learning architecture called DeepIrisNet for iris recognition based on images acquired from different devices. Experiments on the ND-IRIS-0405 and ND-CrossSensor-Iris-2013 [23] Datasets showed that the proposed networks achieved a better performance than the baseline. Nguyen *et al.* [24] applied the five different pretrained CNN models on ImageNet [25] dataset directly to the NIR iris dataset and used a multiclassification SVM method for iris recognition. The experiment was carried out on the LG2200 (NDCrossSensor-Iris-2013) and the CASIA-Iris-Thousand datasets, proving that the five networks outperformed the baseline method [26]. Zanlorensi *et al.* [27] took advantage of the good generalization of two classical networks, VGG and ResNet-50, to apply the pretrained CNNs to study the influence of different iris image input modes on recognition results. A specific data augmentation technique for iris image was proposed. There are also a series of related studies on iris recognition using deep learning network [28]–[34].

In 2016, Hindon's team [8] proposed an implementation of capsule networks. A capsule is defined as a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part. The capsule network performs well in MNIST handwriting image classification test and still maintains a high recognition rate in the case of overlapping numbers. The paper also tries to explain what features the model learns by reconstructing images, which indicates that deep learning is moving towards more complex layer structure and interpretability. As mentioned in [8], there were many possible ways to implement

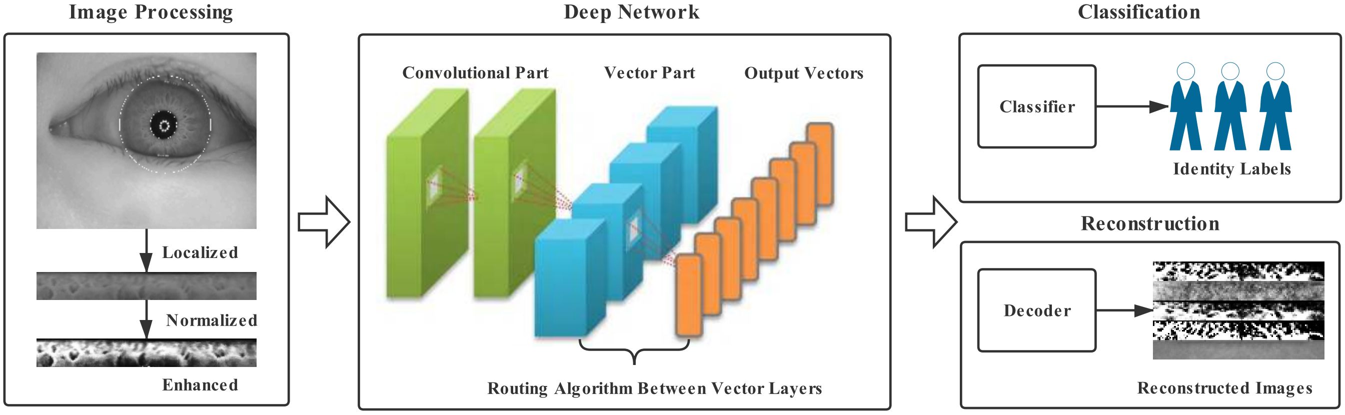


FIGURE 1. Overview of the proposed framework for iris recognition.

the general idea of capsules. A new capsule networks implementation called Matrix capsules [35] was introduced to multiangle object recognition in 2018. The capsule was in the form of a 4 × 4 pose matrix and a logistics unit. Each capsule layer carried out internal linear weighting by sharing weights similar to CNN, and the EM (Expectation-Maximization) algorithm was used to iterate the feature clustering between different capsules.

The appearance of the capsule networks has also suggested new ideas for researchers in some other fields, causing related studies in areas such as text classification [36], visual reconstruction [37], brain tumor classification [38], capsule GAN [39], and deep reinforcement learning [40] to emerge. The capsule network has shown application potential in various fields as described in the related literature. In this work, we will discuss the effect and significance of this technology being used in iris recognition.

III. METHOD

The framework of the proposed method is shown in Figure 1, and consists of four main parts: iris image preprocessing, deep network, classification, and image reconstruction.

A. IRIS IMAGE PREPROCESSING

The iris image preprocessing steps are shown in Figure 2. We first locate and segment the iris texture part from the 640 × 480 original iris image through quality evaluation. Then, we extract the Region of Interest (ROI) from the located iris. Next, we normalize the ROI image to 256 × 32 and enhance it. Finally, in order to unify the input size, which has immense influence on training as an important hyperparameter of the network, the normalized iris image is resized to 197 × 197 by the Nearest Neighbor algorithm as the input.

B. PRETRAINED MODEL FOR CONVOLUTION

The main task of the convolutional part is to extract low-level features from pixel intensities of the input images and form the primary features used for the primary capsule layer. The convolutional part of the capsule networks consists of a convolution layer with 32 channels and 9 × 9 kernels, a stride of 1, making the output shape change to (20, 20, 256) from (28, 28, 1) of the input image. In this study, we use two strategies to construct the convolutional part: (a) as in Table 1, we apply the transfer learning method and introduce three state-of-the-art networks including VGG16, ResNet50, and InceptionV3, which are all pretrained in the ImageNet dataset. To find the most suitable primary features, we divide the entire network into subnetworks according to the number of their component blocks, then we connect the subnetworks with the vector part, and thus, we set up a series of different network instances. In this way, we get the output of different levels in the same network as the result of the convolutional part to find the most compatible options of convolution structure for the entire network through experiments; (b) as in Table 2, we build several shallow convolution networks as the convolutional part without pretraining. These shallow convolution networks include Iris-Dense with

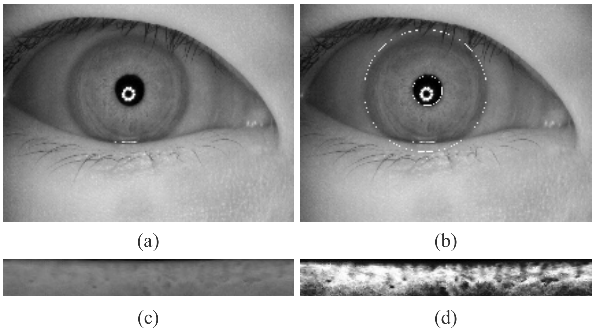


FIGURE 2. Iris image preprocessing: (a) original eye image, (b) iris localization, (c) iris ROI selection and normalization, (d) enhancement.

TABLE 1. Transfer learning used in the convolutional part.

Model	Type	Number of block	depth
VGG16	Convolutional Block	5	23
ResNet50	Residual Block	16	168
InceptionV3	Inception Block	11	159

TABLE 2. Networks with designed shallow convolutional part.

Model	Number of layer
Iris-Dense Net	28
Iris-Inception Net	14
Iris-SE Net	20

two dense blocks, Iris-Inception with one inception block, and Iris-SE with two Squeeze-and-Excitation blocks.

C. ROUTING ALGORITHM IN THE VECTOR LAYER

1) DYNAMIC ROUTING IN THE CAPSULE NETWORK

The dynamic routing algorithm provides the nonlinear mapping for two adjacent capsule layers. The capsule i in layer L is trying to predict the output of capsules j in layer $L+1$.

$$\hat{u}(j|i) = W_{ij} \cdot u_i \tag{1}$$

As given in the equation (1), the predicted feature vector matrix $\hat{u}(j|i)$ is obtained by linear weighting through the output of the capsule u_i in layer L. The weighting matrix W_{ij} is learned during the back propagation procedure.

$$s_j = \sum_i c_{ij} \cdot \hat{u}(j|i) \tag{2}$$

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{3}$$

Then, the output of the capsule j in layer $L+1$ is calculated by equation (2). The coupling coefficient c_{ij} is calculated by the *softmax* function as equation (3), where b_{ij} represents the degree of correlation between capsules in layer L and layer $L+1$, and the initial value of b_{ij} is 0. Update b_{ij} through equation (4) until the iteration requirements are met.

$$b_{ij} = b_{ij} + \hat{u}(j|i) \cdot v_j \tag{4}$$

At each iteration, the output of the capsule j passes through the nonlinear squashing function as equation (5):

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{5}$$

where $\frac{s_j}{\|s_j\|}$ is an unit vector, that is, the length of the original vector is scaled to $\frac{\|s_j\|^2}{1+\|s_j\|^2}$. Thus, the length of the output vector ranges from 0 to 1, which is expressed in a probabilistic manner. By compressing and redistributing the length of the output vectors, the longer vectors become more important, whereas the shorter ones become less important.

The dynamic routing algorithm uses the cosine similarity of two vectors to measure their agreement. However, this manner is not quite good at judging quite good agreement and very good agreement, which usually makes it difficult for the network to converge after training in our experiments.

2) ADJUSTED DYNAMIC ROUTING ALGORITHM

To improve the applicability of the capsule structure and make it easy for the network to converge when dealing with input images of large size, we propose a dynamic routing-based algorithm, which takes direction and length information of vectors into consideration, called the DRDL algorithm. In the iterative calculation of the coupling coefficient c_{ij}, we combine the direction and the length of the capsule as the evaluation indexes of the similarity of two vectors, instead of the original single cosine similarity.

In multidimensional space, we define vector $A = (A_1, A_2, â\quad, A_n)$ and vector $B = (B_1, B_2, â\quad, B_n)$, and the cosine of the two vectors is acquired:

$$\cos\theta = \frac{\sum_1^n (A_i \cdot B_i)}{\sqrt{\sum_1^n A_i^2}\sqrt{\sum_1^n B_i^2}} \tag{6}$$

We calculate the cosine of $\hat{u}(j|i)$ and v_j as the directional similarity to update b_{ij} at each routing iteration as equation (7), where $\|\hat{u}(j|i)|\|$ and $\|v_j|\|$ are the length of $\hat{u}(j|i)$ and v_j, $\|\hat{u}(j|i)|\| = \sqrt{\sum_1^n \hat{u}(j|i)^2}$, $\|v_j|\| = \sqrt{\sum_1^n v_j^2}$.

$$b_{ij} = \frac{\hat{u}(j|i) \cdot v_j}{\|\hat{u}(j|i)|\| \, \|v_j|\|} \tag{7}$$

At the last iteration, the mathematical expression of updating b_{ij} is as given in equation (8). The difference in length of the two vectors is recorded as d, $d = abs(\|\hat{u}(j|i)|\| - \|v_j|\|)$.

$$b_{ij} = b_{ij} \cdot (1 - \frac{d}{1+d}) = \frac{b_{ij}}{1+d} \tag{8}$$

Additionally, the squashing function is used only at the last iteration for the length revising, and we apply $L2$ normalization function at the other routing iterations instead.

Algorithm 1 DRDL

Input: $\hat{u}(j|i)$, the estimation of capsule i in layer L
Output: v_j, the output of capsule j in layer $L+1$

```
Set $b_{ij} = 0$
for $r$ in routing iterations do
    if $r$ is not the last routing iteration then
        $c_{ij} = softmax(b_{ij})$
        $s_i = \sum_i c_{ij}\hat{u}(j|i)$
        $v_j = L2(s_j)$
        $b_{ij} = \hat{u}(j|i) \cdot v_j / \|\hat{u}(j|i)\| \, \|v_j\|$
    else
        $d = abs(\|\hat{u}(j|i)\| - \|v_j\|)$
        $b_{ij} = b_{ij}/1 + d$
        $c_{ij} = softmax(b_{ij})$
        $s_i = \sum_i c_{ij}\hat{u}(j|i)$
        $v_j = Squashing(s_j)$
    end if
end for
return $v_j$
```

As shown in **Algorithm 1**, let $\hat{u}(j|i)$ be the input, which is obtained by the equation (1), and the output v_j is the output of

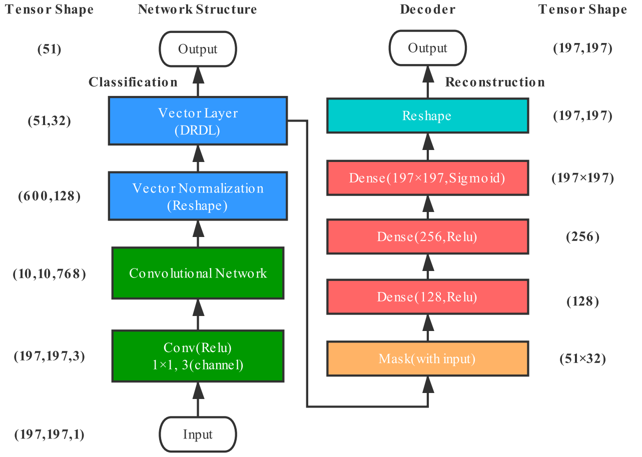


FIGURE 3. Example architecture of proposed method networks.

capsule j in layer $L+1$. The initial value of b_{ij} is 0 as in line 1. For each routing iteration, we first judge if this is the last iteration. If not, as in lines 3 to 7, we update b_{ij} by calculating the directional similarity of the two vector matrices. If it is the last iteration, as in lines 8 to 14, we update b_{ij} by calculating the length similarity of the two vector matrices.

D. NETWORK ARCHITECTURE

The network architecture consists of the convolutional part, vector part, classification, and reconstruction. The input of the network is a 197×197 grayscale image. We first change the channel from 1 to 3 through a convolutional layer with a 1×1 kernel. Then, we input the $197 \times 197 \times 3$ feature to the convolution network, which is the InceptionV3 network shown in Figure 3. The number of layers in the convolutional part can vary. In this example, the pretrained InceptionV3 model with 4 inception blocks is applied to output a feature map with a size of 10×10 and a channel of 768. The tensor shape turns to 600×128 through vector normalization and reshaping. Finally, 51×32 (51 represents the number of categories, and 32 is the dimension of each capsule) vector features output is normalized by $L2$ normalization as the output of the network, which represents the probability of the existence of each category. The network on the right side in Figure 3 is a fully connected decoder network, which is used for the iris image reconstruction.

IV. EXPERIMENTS

In this section, we verify the validity of the proposed method on three iris datasets. First, we give the instructions of the iris image datasets and explain the parameters of the experiments. Next, we list the performance of several network structures composed of the convolutional part and vector part in different datasets. Then we compare them with the methods in [24], [26], and [27]. Finally, we simulate a specific environment affected by illumination and discuss the sensitivity of the network to pupil size. We design a dataset of different pupil sizes and show that the proposed method is more stable in this case through experiments. Accuracy and equal error rate (EER) are used as the evaluation criteria.

TABLE 3. The 3 experimental iris datasets.

Dataset	Class	Sample Size
JluV3.1	60	28 to 30
JluV4	86	600 to 1800
CASIA-V4 Lamp	822	20

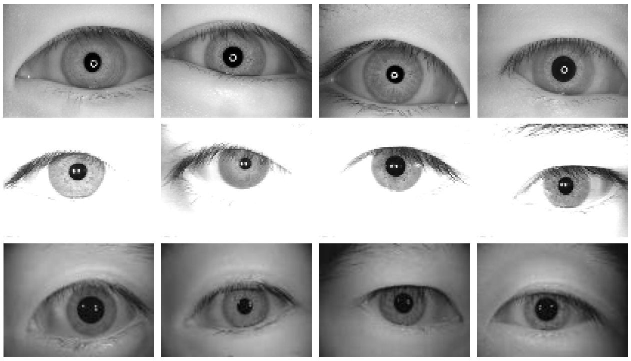


FIGURE 4. Sample images from the JluV3.1 (first row), JluV4 (second row), and CASIA-V4 Lamp (third row) datasets.

A. DESCRIPTION OF THE DATASETS

The three datasets, the JluV3.1, the JluV4, and the CASIA-V4 Lamp iris dataset, as in Table 3, comprise 1780, 114904 and 16215 iris images obtained from 60, 86, and 822 subjects respectively. Each subject has the same number of left eye and right eye images with the resolution of 640×480. The left and right iris images from the same subject are considered to be different classifications in the experiment. The samples from the three datasets are shown in Figure 4. It needs to be mentioned that the data augmentation technique was not employed to increase the number of samples in this work.

The three iris datasets in the experiment represent three types of datasets, i.e., (1) the JluV3.1 represents the kind of datasets with few categories and few samples, (2) the JluV4 represents the kind of datasets with few categories and many samples, and (3) the CASIA-V4 Lamp represents the kind of datasets with many categories and few samples.

B. EXPERIMENTAL SETTINGS

Iris images are divided into two parts for training and testing after quality evaluation and image preprocessing. We use random sampling method to select training and test sets proportionally. The training set and testing set of JluV3.1 contains 1275 and 255 images respectively. The training set and testing set of JluV4 contain 46800 and 5200 images respectively. We select 300 categories in the CASIA-V4 Lamp dataset, 4500 images for training and 1500 for testing. The epoch of training is 50 and the Learning Rate (LR) varies in different experiments. A GPU 1080ti is used to accelerate training in the experiment.

In the experiment of recognition, the decoder network is not used as an additional loss for training. On the one hand, the loss function of the decoder has a tiny effect on the

TABLE 4. Comparisons of different networks on the JluV3.1 dataset.

Network Structure	LR	Accuracy%	EER%
VGG16_4blocks+DRDL	0.00001	98.43	0.31
VGG16_5blocks+DRDL	0.00001	98.82	0.42
ResNet50_8blocks+DRDL	0.0005	57.25	17.79
ResNet50_9blocks+DRDL	0.0005	62.75	14.46
ResNet50_10blocks+DRDL	0.0005	83.53	6.91
ResNet50_11blocks+DRDL	0.0005	92.16	4.91
ResNet50_12blocks+DRDL	0.0005	96.47	2.83
ResNet50_13blocks+DRDL	0.0005	69.02	15.8
ResNet50_14blocks+DRDL	0.0005	98.04	0.51
ResNet50_15blocks+DRDL	0.0005	99.22	0.078
ResNet50_16blocks+DRDL	0.0005	98.82	0.64
InceptionV3_1block+DRDL	0.00002	89.02	6.3
InceptionV3_3blocks+DRDL	0.00002	81.96	8.46
InceptionV3_4blocks+DRDL	0.00002	98.04	0.89
InceptionV3_5blocks+DRDL	0.00002	99.37	0.039
InceptionV3_6blocks+DRDL	0.00002	98.83	0.21
InceptionV3_7blocks+DRDL	0.00002	98.43	0.45
InceptionV3_8blocks+DRDL	0.00002	97.25	1.7
InceptionV3_9blocks+DRDL	0.00002	97.27	0.78
InceptionV3_10blocks+DRDL	0.00002	85.88	5.31
InceptionV3_11blocks+DRDL	0.00002	72.94	11.36
Iris-Dense+DRDL	0.00001	97.26	1.55
Iris-Inception+DRDL	0.00001	97.65	0.9
Iris-SE+DRDL	0.00001	94.51	1.8
Daugman	–	95.7	2.48
DenseNet_6layers+SVM	–	97.8	2.09
non-seg+nonnorm+ResNet50	0.001	97.93	0.32

TABLE 5. Comparisons of different networks on the JluV4 dataset.

Network Structure	LR	Accuracy%	EER%
VGG16_4blocks+DRDL	0.00001	98.12	0.998
VGG16_5blocks+DRDL	0.00001	98.52	0.65
ResNet50_8blocks+DRDL	0.00001	97.75	0.76
ResNet50_9blocks+DRDL	0.00001	95.33	1.02
ResNet50_10blocks+DRDL	0.00001	95.31	1
ResNet50_11blocks+DRDL	0.00001	97.06	1.13
ResNet50_12blocks+DRDL	0.00001	96.87	1.1
ResNet50_14blocks+DRDL	0.00001	95.23	1.84
ResNet50_15blocks+DRDL	0.00001	94.75	2.16
ResNet50_16blocks+DRDL	0.00001	92.46	3.27
InceptionV3_1block+DRDL	0.00001	98.79	0.35
InceptionV3_2blocks+DRDL	0.00001	95.25	1.33
InceptionV3_3blocks+DRDL	0.00001	97.56	1.13
InceptionV3_4blocks+DRDL	0.00001	98.88	0.295
InceptionV3_5blocks+DRDL	0.00001	98.15	0.67
InceptionV3_6blocks+DRDL	0.00001	98.60	0.84
InceptionV3_7blocks+DRDL	0.00001	97.81	1.42
InceptionV3_8blocks+DRDL	0.00001	94.15	3.55
InceptionV3_9blocks+DRDL	0.00001	93.96	3.49
InceptionV3_10blocks+DRDL	0.00001	80.35	8.15
InceptionV3_11blocks+DRDL	0.00001	67.98	13.54
Iris-Dense+DRDL	0.001	99.42	0.13
Iris-Inception+DRDL	0.001	99.38	0.11
Iris-SE+DRDL	0.001	99.04	0.39
Daugman	–	98.6	0.69
DenseNet_6layers+SVM	–	96.97	2.59
non-seg+nonnorm+ResNet50	0.001	99.14	0.15

training of the entire network. On the other hand, the decoder will bring a huge amount of parameters when reconstructing the image with a higher rate of separation.

C. ANALYSIS OF PERFORMANCE ON STRUCTURES

In this section, we conduct a series of experiments in three iris datasets through different network structures. The networks with pretrained VGG16, ResNet50, and InceptionV3 model substructure are fine-tuned. The networks Iris-Dense+DRDL, Iris-Inception+DRDL, and Iris-SE+DRDL with shallow convolution structures are fully trained. We explore the effect on the recognition results of different convolutional parts combined into a capsule structure, as listed in Table 4, Table 5, and Table 6. The networks are all trained with 3 routing iterations, and the output of the vector part has 32 dimensions. The results of accuracy less than 50% are not shown.

We use three methods with the best performance in [24], [26], and [27] as the baseline for comparison. The first method is Daugman's gabor phase-quadrant feature extraction method combined with Hamming distance matching [26]. The second one applies pretrained DenseNet of 6 layers without fine-tuning to extract features from the normalized iris images and uses a multi-class Support Vector Machine (SVM) for classification, with a one-against-all strategy[24]. The third baseline method is the model learned on the pretrained ResNet50 architecture with fine-tuning using non-segmented and nonnormalized images[27]. The results of the baseline are also shown in Table 4, Table 5, and Table 6.

Daugman's method [26] achieves recognition accuracies of 95.7%, 98.6%, and 92.4% on the JluV3.1, the JluV4 and the CASIA-V4 Lamp datasets, respectively. The DenseNet_6layers+SVM [24] achieves recognition accuracies of 97.8%, 96.97%, and 93.65% on the JluV3.1, the JluV4 and the CASIA-V4 Lamp datasets, respectively. The non-seg+nonnorm+ResNet50 [27] achieves recognition accuracies of 97.93%, 99.14%, and 93.57% on the JluV3.1, the JluV4 and the CASIA-V4 Lamp datasets, respectively.

In the experiment involving the JluV3.1 dataset, we change the vector dimension of the output on the same network to investigate its influence on the result. Furthermore, we compare the DRDL and the dynamic routing algorithms as well, as presented in Table 7.

We find that when using the pretrained model substructures combined with the dynamic routing algorithm, the networks are unable to converge in the training process, resulting in bad performance. But the problem is alleviated when using the pretrained model substructures combined with the DRDL.

TABLE 6. Comparisons of different networks on the CASIA-V4 Lamp dataset.

Network Structure	LR	Accuracy%	EER%
VGG16_4blocks+DRDL	0.000005	93.87	1.21
VGG16_5blocks+DRDL	0.000005	82.13	2.4
ResNet50_8blocks+DRDL	0.00001	86.87	1.52
ResNet50_9blocks+DRDL	0.00001	90.71	1.3
ResNet50_10blocks+DRDL	0.00001	90.60	1.14
ResNet50_11blocks+DRDL	0.00001	90.53	1.04
ResNet50_12blocks+DRDL	0.00001	90.73	1.31
ResNet50_13blocks+DRDL	0.00001	87.87	1.5
ResNet50_14blocks+DRDL	0.00001	81.60	2.34
ResNet50_15blocks+DRDL	0.00001	70.40	5.59
ResNet50_16blocks+DRDL	0.00001	70.27	5.33
InceptionV3_1block+DRDL	0.0001	91.13	1.01
InceptionV3_2blocks+DRDL	0.0001	53.73	5.05
InceptionV3_3blocks+DRDL	0.0001	63.00	4.07
InceptionV3_4blocks+DRDL	0.0001	87.27	1.4
InceptionV3_5blocks+DRDL	0.0001	92.27	1.17
InceptionV3_6blocks+DRDL	0.0001	82.40	2.5
InceptionV3_7blocks+DRDL	0.0001	82.93	2.74
InceptionV3_8blocks+DRDL	0.0001	63.47	7.71
InceptionV3_9blocks+DRDL	0.0001	52.01	14.18
Iris-Dense+DRDL	0.001	78.07	3.37
Iris-Inception+DRDL	0.001	70.00	5.25
Daugman	–	92.4	3.63
DenseNet_6layers+SVM	–	93.65	4.05
non-seg+nonnorm+ResNet50	0.001	93.57	2.08

It shows that the dynamic routing algorithm works well with the fully trained shallow networks, but it is incompatible with the pretrained model substructures in our datasets.

Through the experimental results, we can see that the best performance is not from the network with the most or least layers in the convolutional part. On the contrary, the appropriate level of feature extraction achieve better results. As shown in the result, among the networks using pretrained VGG16-based architecture, VGG16_5blocks+DRDL (VGG16 with 5 convolutional blocks) achieves the best accuracy of 98.82% and 98.52% (wathet) on the JluV3.1 and the JluV4 datasets. VGG16_4blocks+DRDL achieves the best accuracy of 93.87% on the CASIA-V4 Lamp dataset. For the pretrained VGG16, choosing 4 blocks is generally better, regardless of sample size or the number of labels. Among the networks using pretrained ResNet50-based architecture, ResNet50_15blocks+DRDL (ResNet50 with 15 residual blocks) achieves the best accuracy of 99.22% on the JluV3.1 dataset. ResNet50_8blocks+DRDL achieves the best accuracy of 97.75% on the JluV4 dataset. ResNet50_12blocks+DRDL achieves the best accuracy of 90.73% on the CASIA-V4 Lamp dataset. ResNet50 performs well at a deeper depth due to its skip connection structures. So the performance is good when applying more residual blocks. But once the number of labels increases significantly, the networks with 9-12 residual blocks get a better result than those with more deeper structures. Among the networks using pretrained InceptionV3-based architecture, InceptionV3_5blocks+DRDL (InceptionV3 with 5 inception blocks) achieves the best accuracy of 99.37% on the JluV3.1 dataset. InceptionV3_4blocks+DRDL achieves the best accuracy of 98.88% on the JluV4 dataset. InceptionV3_5blocks+DRDL achieves the best accuracy of 92.27% on the CASIA-V4 Lamp dataset. The networks with the pretrained InceptionV3 architectures of 4 or 5 inception blocks work well when the number of samples is small. But when the training set is sufficient, except for networks less than 4 or more than 9 inception blocks, the results are all basically good.

According to the results, we have the following observations. In the experiment of the JluV3.1 dataset, the results of the networks with pretrained models and shallow convolution models are basically similar, and the ones with pretrained models perform slightly better. In the experiment of the JluV4 dataset, because there are a sufficient number of samples, the networks with shallow convolution models can be fully trained, so the effect is slightly better compared with the fine-tuned networks. In the experiment of the CASIA-V4 Lamp dataset, the first dimension of output vectors increases because the number of categories rises. As a result, the expressive capacity of the primary features learned by shallow convolution is very limited with insufficient samples, leading to the results of networks with shallow convolution models to decrease obviously. The performance of networks with transfer learning method also declines, but according to the feature expression ability learned by pretrained models, the decline is not very dramatic.

We investigate the influence of another two hyperparameters in the network. As listed in Table 7, in the Iris-Inception network, we set the vector dimensions of output to 16, 24, and 32, and adopt the DRDL algorithm and the dynamic routing algorithm respectively. We find that the effect of vector dimension on the classification performances is not very obvious. The network with higher dimension vectors is not necessarily better than that with lower dimension vectors. The vectors with higher dimensions have richer views for describing features, but it is not directly proportional to the accuracy of recognition. By contrast, the networks applying the DRDL algorithm get a better result than those applying the dynamic routing. We show the training loss curve and training accuracy curve in Figure 5. It can be found that the training tends to be stable after approximately 20 iterations, and the network with the output vectors of high dimension has a faster convergence speed.

D. ENVIRONMENTAL SIMULATION OF LIGHT INTENSITY

To prove that a network with capsule architecture can learn the relationship between features and the rule of feature changing, we design a specific iris recognition environment for testing the performance of the networks with varied architectures. Different illumination intensity is generally used to distinguish genuine and fake irises or verify iris localization

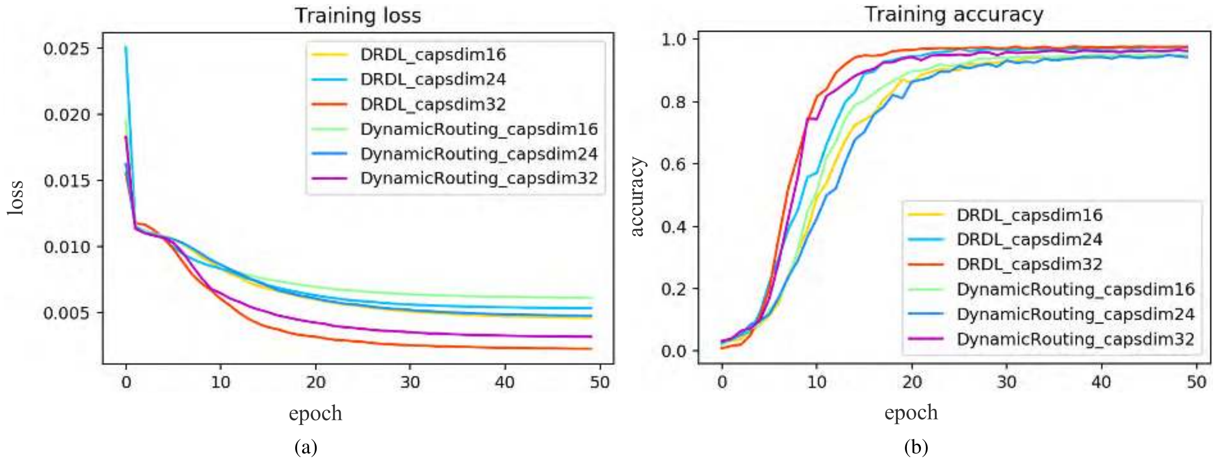


FIGURE 5. Loss Curve (a) and Accuracy Curve (b) of Iris-Inception model trained on the JluV3.1 dataset.

TABLE 7. Comparisons of different Iris-Inception architectures on the JluV3.1 dataset.

Network Structure	Dimension of Capsule	LR	Accuracy%	EER%
Iris-Inception+Dynamic routing	16	0.001	96.29	1.8
Iris-Inception+DRDL	16	0.001	96.47	1.67
Iris-Inception+Dynamic routing	24	0.001	95.78	2.12
Iris-Inception+DRDL	24	0.001	98.43	1.23
Iris-Inception+Dynamic routing	32	0.001	96.86	1.08
Iris-Inception+DRDL	32	0.001	97.65	0.9

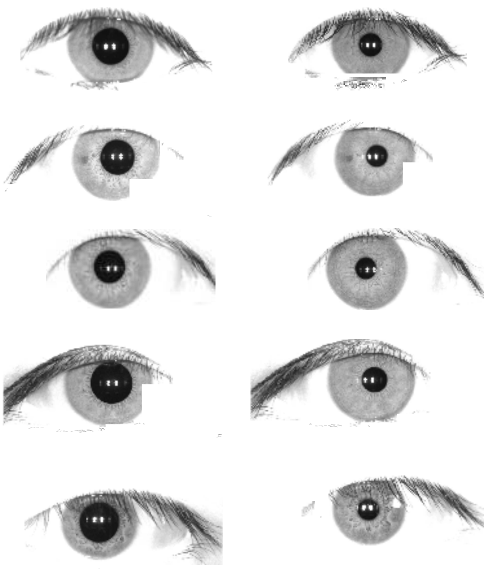


FIGURE 6. Large pupil under weak light (left), small pupil under intense light (right).

algorithms. However, little research has been published on the influence of illumination intensity on iris recognition. We assume that the model is trained only in the strong light, and then validated in a weak light environment, and vice versa. To imitate this particular environment, we build a small simulated light intensity experimental dataset, including images selected from JluV4, which has a larger number of samples under different illumination intensities. Because the iris images are collected through a fixed focus mode, we divide the pupil size into three grades, level 1, level 2, and level 3, based on the pupil diameter. To make the simulation environment more extreme, we consider iris images of level 1 and level 3, i.e., large pupil and small pupil representing iris images captured in weak and intense light (each image set has 1200 images from 20 categories) in this section, as shown in Figure 6 (each row represents the same classification).

Two training strategies are applied: (a) one is to train with iris images of the large pupil and test with the small ones; (b) the other one is the opposite. In this way, we can verify the performance and stability of the model in different environments. The output vector dimension is 32 and the epoch of training is 50.

The results including the three baseline methods are listed in Table 8 and Table 9. We choose three networks, the 12-layer CNN (6 convolution layers, 2 max pooling layers, and 1 global average pooling layer), VGG16, and InceptionV3 to represent the convolution network without vector structure. We additionally choose three networks, the capsule network,VGG16_5blocks+DRDL, and InceptionV3_6blocks+DRDL to represent the network with vector structure.

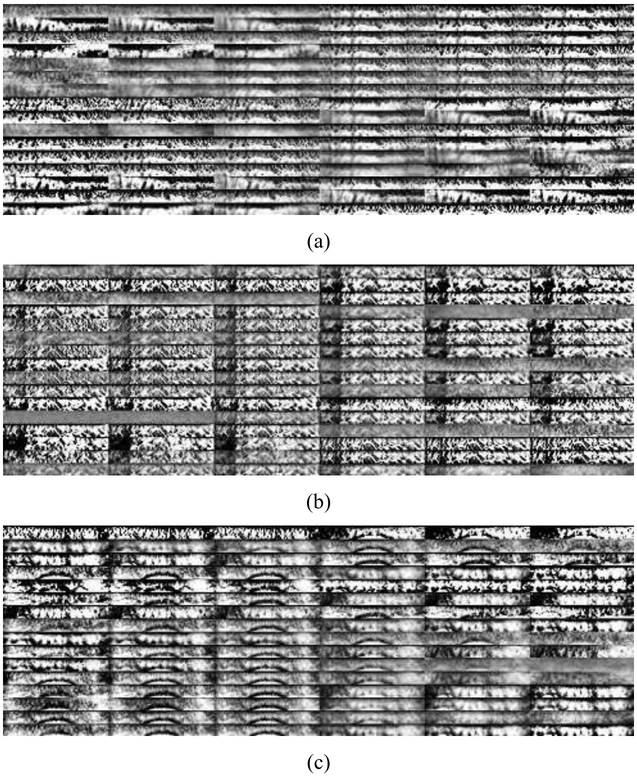
Although iris normalization will weaken the effect of pupil dilation on iris texture stretching to a certain

TABLE 8. Comparisons of different networks with training strategy (a).

Network Structure	LR	Training Accuracy%	Testing Accuracy%	DR%	EER%
12-layer CNN	0.001	88.09	40.02	54.6	18.26
Capsule Network	0.001	97.71	83.48	14.56	4.74
VGG16	0.0002	86.09	43.51	49.46	13.26
VGG16_5blocks+DRDL	0.0002	99.36	83.34	16.12	4.11
InceptionV3	0.00001	99.55	44.50	55.3	15.79
InceptionV3_6blocks+DRDL	0.00001	99.64	92.34	7.33	2.98
Daugman	–	96.52	18.96	81.2	75.09
DenseNet_6layers+SVM	–	98.26	32.09	67.34	36.88
non-seg+nonnorm+ResNet50	0.001	99.91	72.13	27.8	6.47

TABLE 9. Comparisons of different networks with training strategy (b).

Network Structure	LR	Training Accuracy%	Testing Accuracy%	DR%	EER%
12-layer CNN	0.001	79.55	39.79	49.98	18.24
Capsule Network	0.001	97.47	82.61	15.25	4.9
VGG16	0.0002	97.82	61.50	37.13	10.89
VGG16_5blocks+DRDL	0.0002	99.64	84.93	14.76	4.74
InceptionV3	0.00001	99.73	43.46	56.42	15.74
InceptionV3_6blocks+DRDL	0.00001	99.45	86.13	13.39	3.76
Daugman	–	98.2	14.5	98.05	80.07
DenseNet_6layers+SVM	–	98.89	59.6	39.73	25.1
non-seg+nonnorm+ResNet50	0.001	99.82	65.39	34.49	9.95


(a)

(b)

(c)

FIGURE 7. Reconstructing iris images, 16 rows representing 16 vector feature dimensions. Each row has 6 reconstructions tweaked in [−4, −2, −1, 1, 2, 4].

extent by using the linear transformation method, we can still find that this particular situation will interfere with iris recognition. Obviously, the network with capsule structure is more stable in this particular situation. The results show that the accuracy of the networks with capsule architecture decreases by approximately 15%. The accuracy of InceptionV3_6blocks+DRDL (wathet) network reaches 92.34%, with a decrease of only 7.33% in the first simulation environment, and reaches 86.13%, with a decrease of 13.39% in the second environment. CNN models lose a specific input source for training, so its generalization ability decreases correspondingly. The accuracy of the three deep convolution networks decreases by approximately 50%. The test results show that these models have basically lost the ability to classify correctly.The baseline also does not achieve good results in this experiment. The performance of [27] non-seg+non-norm+ResNet50 decreases relatively small, and the testing accuracy of more than 65% could be retained. However, the performance of the other two baseline methods decreased notably on the testing set.

E. RECONSTRUCTION

Image reconstruction is applied for us to observe the way of feature vectors express the instances they represent in the network more intuitively. We use a decoder network to reconstruct the iris image, and the dimension of the output feature vectors is 16. By adding offsets (−4 to 4) to the feature dimensions, we can see how the reconstructing images change. It can be noticed that the differences of reconstructed images are reflected in the image energy, fuzzy degree, texture extension direction, and edge thickness.

As shown in Figure 7, we display three reconstruction examples of the iris images in the JluV4 acquired from a 9-layer CNN+DRDL(4 convolution layers and 1 global average pooling layer) network. It is interesting that the reconstructed image of (c) has obvious noise of the eyelid, but with the change of offset, some dimension disturbances weaken or even ignore the eyelid portion.

V. CONCLUSION

In this paper, we introduce the capsule network method into iris recognition. We build several convolution structures with different depths according to different outputs of pretrained classic networks to dock with the capsule structure. The dynamic routing algorithm is adjusted with the consideration of direction and length of the vector. We set up several network instances with different structures and depths, and we validate them using three iris datasets. Finally, we simulate an environment with different illumination intensities, and we train and test iris images with different pupil sizes (represent different light intensities) to show the recognition ability of this method when the environment is varied.

Experiments show that a deep network with capsule architecture is feasible in iris recognition. In the test of the JluV3.1 iris dataset, InceptionV3_5blocks+DRDL achieves the highest accuracy of 99.37%. In the test of the JluV4 dataset, Iris-Dense+DRDL achieves the highest accuracy of 99.42%. In the test of the CASIA-V4 Lamp dataset, VGG16_4blocks+DRDL achieves the highest accuracy of 93.87%. Convolution structures with different depths have obvious effects on the results. Convolution structures with too deep or too shallow depths in different networks sometimes make it difficult for the model to learn the correct and appropriate feature description. In addition, the network performance with a pretrained model structure as convolutional part is more stable than that using shallow full-training convolution structure. The change in sample size leads to performance variation among networks with shallow convolution structures, and this performance variation indicates the necessity of transfer learning. Finally, the simulation experiment shows that the network with capsule architecture guarantees the accuracy to a certain extent compared with CNNs and the baseline. InceptionV3_6blocks+DRDL network performs best, and the test accuracy reaches 92.34% and 86.13% under the two training
strategies.

This paper describes an attempt to apply the capsule network in the field of iris recognition and the proposed method needs to be further optimized. In future, our research will develop in the following directions: (1) Exploring other vector forms or vector structures and corresponding iteration or other learning methods, (2) Applying capsule (vector) feature learning network to deal with iris recognition problems of heterogeneous iris and other specific environments, (3) Researching network processing capacity and processing methods when the number of categories becomes extremely large.

REFERENCES

[1] R.P. Wildes, "Iris recognition: An emerging biometric technology," *Proc. IEEE*, vol. 85, no. 9, pp. 1348–1363, Sep. 1997.

[2] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.

[3] R. P. Wildes *et al.*, "A machine-vision system for iris recognition," *Mach. Vis. Appl.*, vol. 9, no. 1, pp. 1–8, 1996.

[4] W. W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1185–1188, Apr. 1998.

[5] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Personal identification based on iris texture analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1519–1533, Dec. 2003.

[6] H. Proenca and L. A. Alexandre, "Toward noncooperative iris recognition: A classification approach using multiple signatures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 607–612, Apr. 2007.

[7] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[8] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3859–3869.

[9] *Jlu Iris Database*. Accessed: Apr. 2018. [Online]. Available: http://www.jlucomputer.com/Irisdb.php

[10] Chinese Academy of Sciences Institute of Automation. *CASIA Iris Image Database*. Accessed: Aug. 2017. [Online]. Available: http://biometrics.idealtest.org/

[11] A Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst* Red Hook, NY, USA: Curran Associates, 2012, pp. 1–9.

[12] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/arXiv:1409.1556

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[14] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, Jul. 2017, vol. 1, no. 2, pp. 4700–4708.

[15] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.

[17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AAAI*, vol. 4, 2017, p. 12.

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. (2016). "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size." [Online]. Available: https://arxiv.org/abs/arXiv:1602.07360

[19] J. Hu, L. Shen, G. Sun, and E. Wu. (2017). "Squeeze-and-excitation networks." [Online]. Available: https://arxiv.org/abs/arXiv:1709.01507

[20] Y. Yang, Z. Zhong, T. Shen, and Z. Lin, "Convolutional neural networks with alternately updated clique," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2413–2422.

[21] N. Liu, M. Zhang, H. Li, Z. Sun, and T. Tan, "DeepIris: Learning pairwise filter bank for heterogeneous iris verification," *Pattern Recognit. Lett.*, vol. 82, pp. 154–161, Oct. 2016.

[22] A. Gangwar and A. Joshi, "DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2301–2305.

[23] P. J. Phillips *et al.*, "FRVT 2006 and ICE 2006 large-scale experimental results," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 831–846, May 2010.

[24] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan, "Iris recognition with off-the-shelf CNN features: A deep learning perspective," *IEEE Access*, vol. 6, pp. 18848–18855, 2018.

[25] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[26] J. Daugman, "How iris recognition works," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 21–30, Jan. 2004.

[27] L. A. Zanlorensi, E. Luz, R. Laroca, A. S. Britto, Jr., L. S. Oliveira, and D. Menotti. (2018). "The impact of preprocessing on deep representations for iris recognition on unconstrained environments." [Online]. Available: https://arxiv.org/abs/arXiv:1808.10032

[28] N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, and T. Tan, "Accurate iris segmentation in non-cooperative environments using fully convolutional networks," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2016, pp. 1–8.

[29] E. Jalilian and A. Uhl, "Iris segmentation using fully convolutional encoder–decoder networks," *Deep Learning for Biometrics*. Cham, Switzerland: Springer, 2017, pp. 133–155.

[30] E. Severo *et al.* (2018). "A benchmark for iris location and a deep learning detector evaluation." [Online]. Available: https://arxiv.org/abs/arXiv:1803.01250

[31] D. Menotti *et al.*, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 864–879, Apr. 2015.

[32] J. Tapia and C. Aravena, "Gender classification from NIR iris images using deep learning," *Deep Learning for Biometrics*. Cham, Switzerland: Springer, 2017, pp. 219–239.

[33] L. He, H. Li, F. Liu, N. Liu, Z. Sun, and Z. He, "Multi-patch convolution neural network for iris liveness detection," in *Proc. IEEE 8th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2016, pp. 1–7.

[34] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "A deep learning approach for iris sensor model identification," *Pattern Recognit. Lett.*, vol. 113, pp. 46–53, Oct. 2018.

[35] N. F. Geoffrey, E. Hinton, S. Sabour, "Matrix capsules with EM routing," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[36] W. Zhao *et al.* (2018). "Investigating capsule networks with dynamic routing for text classification." [Online]. Available: https://arxiv.org/abs/arXiv:1804.00538

[37] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao. (2018). "Accurate reconstruction of image stimuli from human fMRI based on the decoding model with capsule network architecture." [Online]. Available: https://arxiv.org/abs/arXiv:1801.00602

[38] P. Afshar, A. Mohammadi, and K. N. Plataniotis. (2018). "Brain tumor type classification via capsule networks." [Online]. Available: https://arxiv.org/abs/arXiv:1802.10200

[39] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan. (2018). "CapsuleGAN: Generative adversarial capsule network." [Online]. Available: https://arxiv.org/abs/arXiv:1802.06167

[40] P.-A. Andersen. (2018). "Deep reinforcement learning using capsules in advanced game environments." [Online]. Available: https://arxiv.org/abs/arXiv:1801.09597

TIANMING ZHAO received the M.S. degree from Jilin University, China, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include applications of computer vision and deep learning techniques for biometrics.

YUANNING LIU received the Ph.D. degree from Jilin University, China, in 2004. He completed the Ph.D. Research with the University of Vienna, Austria, in 2007. He was a Visiting Scholar with the University of Missouri, USA, in 2015. He is currently a Professor in computer science with Jilin University. His research interests include software engineering, iris biometrics, pattern recognition, and bioinformatics.

GUANG HUO received the Ph.D. degree from Jilin University, China, in 2016. He is currently an Associate Professor with Northeast Electric Power University. His research interests include pattern recognition, machine learning, biometrics, and image processing.

XIAODONG ZHU received the Ph.D. degree from Jilin University, China, in 2004, where he is currently a Professor of computer science. His research interests include software engineering, pattern recognition, and machine learning techniques for biometrics.