

Article

# FM<sub>net</sub>: Iris Segmentation and Recognition by Using Fully and Multi-Scale CNN for Biometric Security

Rachida Tobji <sup>1,\*</sup>, Wu Di <sup>1,\*</sup> and Naeem Ayoub <sup>2</sup><sup>1</sup> Department of Computer science and technology, Dalian University of technology, Dalian 116000, China<sup>2</sup> Department of Mathematics & Computer Science, University of Southern Denmark, Cam-pusvej 55, DK-5230 Odense M, Denmark; naeemayoub666@gmail.com

\* Correspondence: racha@mail.dlut.edu.cn (R.T.); wudi@dlut.edu.cn (W.D.)

Received: 4 April 2019; Accepted: 11 May 2019; Published: 17 May 2019



**Featured Application:** The research work proposes an iris segmentation and recognition method “FM<sub>net</sub>” that can reduce computational complexity for clinical investigations.

**Abstract:** In Deep Learning, recent works show that neural networks have a high potential in the field of biometric security. The advantage of using this type of architecture, in addition to being robust, is that the network learns the characteristic vectors by creating intelligent filters in an automatic way, grace to the layers of convolution. In this paper, we propose an algorithm “FM<sub>net</sub>” for iris recognition by using Fully Convolutional Network (FCN) and Multi-scale Convolutional Neural Network (MCNN). By taking into considerations the property of Convolutional Neural Networks to learn and work at different resolutions, our proposed iris recognition method overcomes the existing issues in the classical methods which only use handcrafted features extraction, by performing features extraction and classification together. Our proposed algorithm shows better classification results as compared to the other state-of-the-art iris recognition approaches.

**Keywords:** iris recognition; iris segmentation; Fully Convolutional Network (FCN); multi-scale Convolutional Neural Network (MCNN); convolutional neural networks (CNN)

## 1. Introduction

Nowadays iris recognition is becoming more important security feature in biometric security systems. Biometrics refers to the science of biological variation and related phenomena. A system is called biometric recognition if it is able to automatically determine the identity of a human being based on the measurement of biological traits variables between individuals. Biometric systems recognize a person (or authenticate his identity) whose identity has previously been registered in a database (of N “authorized” persons).

General terms of authentication or recognition cover both identification and verification. Identification in the sense of the term implies a closed group context. It means that the certain user of the biometric system belongs to the N authorized persons, which is about determining the one of the N’s best suited to the user, and therefore, it is not necessary to set a threshold for accepting or rejecting the user. We often speak of identification “1-among-N”. Verification in the of term of senses operates in a context of the open group. In other words, we are absolutely not certain that the identity of the user is actually known by the system. In practice, the user claims the identity of N individuals from the database. If it is not recognized as such, it is an impostor. We often speak “1-for-1” verification. On the margins of military and police applications, more and more consumer biometric systems are being implemented to replace (or secure) the personal codes, passwords, cards, keys, which today respond to the various authentication needs of individuals.

In general, an algorithm which is applied to recognize human activity, conditioned by pretreatment of the training data i.e., SVM, Random Forest and HMM [1–3]. However, all of these previous works are mainly performed in controlled environments and are not very robust, and sometimes very demanding in terms of computational time cost as well as memory space.

Deep Neural Networks (DNNs) [4] have been successfully applied to various issues such as gesture recognition and handwriting classification [5], where DNNs are directly applied to the data flow without pre-processing or feature selection step. We have analyzed the results obtained using the neural network to better understand how it works.

The convolutional neural network or ConvNet CNN is very used in the field of computer vision if we only evoke imageNet, and pascal voc which are problems of detection and classification of objects in an image. CNNs also have a lot of success in the biometric surveys. The convolutional neural network specializes in the processing of matrix and signals data. In convNet, two new layer types are presented to the network: the convolutional layer and the pool layer. We describe them in the following parts. Fully automatic processing is performed in 4 steps: Segmentation, Normalization, Feature extraction for each eye image, and finally Classification. In this paper, we propose an approach based on the vision deep neural networks. The first and last two steps are merged with deep neural networks that perform both the automatic segmentation and extraction of features and their classification. Moreover, a fully approach is proposed using several deep networks at different scales (size of the input), and whose outputs are merged.

#### *Why Deep Learning Used in Iris Recognition*

Deep learning is a branch of machine learning. A field in which the computer tests, program and learns from algorithms to improve by itself. Machine learning is not new and its roots date back to the mid-20th century. From the last two decades, several artificial intelligence techniques have been developed such as neural networks. Neural networks-based algorithms support deep learning which play a pivotal role in the recognition and computer vision problems. Inspired by the neurons that make up the human brain, neural networks combine layers of neurons that are interconnected to form several interconnected layers. For example, you want a neural network to recognize images that contain at least one eye of individual person. However, factors may effect the iris recognition methods such as different colors, sizes and forms of eye, different lightening and backgrounds. So we need to compile a training package of images - thousands of examples of person eyes. The neural network is then nourished by this image which transforms into data moving in the neural network between the neurons. In the end, the last layer collects all these pieces of information to reach a result of recognizing the person.

The rest of the paper is organized as follows: Section 2 describes the background of the iris recognition and the methods working in deep neural networks. Section 3 presents the related work in contrast to basic concept of the convolutional neural networks (CNNs) and different architectures of CNNs. In Section 4 proposed model is described briefly, while the experimental results and discussions are presented in Section 5. Finally, conclusion is made in Section 6.

## **2. Background**

The iris placed behind the cornea of the eye is a variable diaphragm pierced by a hole X circular called pupil, controlled by a sphincter and a dilator formed of fiber antagonistic, smooth, radiant and circular muscle fibers. Practical observation through an optical system only allows detecting the edges macroscopic, and not to go down to the level of the elementary tubes. These random patterns of the iris are unique for each individual: they constitute somehow a human bar code from the filaments, hollows and streaks in the colored circles that surround the pupil of each eye. In addition, in the case of an internal tissue, these iridian impressions are relatively safe lesions.

The neural network appeared in the 1940 under the name “multilayer perceptrons [6,7]. Multilayer perceptron is defined by Warren McCulloch and Walter Pitts, who showed that any function could

be approximated arithmetical or logical with multilayer perceptrons. In 1969, the perceptrons could only do binary classifications and on the other hand, difficulties were encountered while performing the classification of the XOR logic gate. In 1980, the complete re-propagation algorithm is developed by David E Rumelhart et al. [8]. In the same year, a group of researchers led by Geoff Hinton at the University of Toronto, Canada, found a way to train the neural network without falling into the local minima problem [9]. In this era, graphics processing units (GPUs) were also developed, which facilitated the handling of images on the PC. Before this researchers used super giant computers just to experience the technology of learning images. The change occurred in 2007, when computer scientist Faye Leigh of Stanford University and Lee of Princeton University launched ImageNet, a network of images that contains millions of images described by humans. Yann LeCun popularized neural networks with LeNet the first convolutional neural network for character recognition in 1998 [8,10].

In 2001, Paul Viola and Michael Jones at the Mitsubishi Electric Research Laboratory in the United States used Gartem (arithmetic equation) of machine learning techniques to recognize the faces in real-time images. Instead of using neurons of different weights, we use the technique of passing images in a series of simple decisions, such as whether the image contains a light-colored point between the dark spots, which may be the top of the nose? Are there two dark areas above a large pale area? Which represent the eyes and cheeks in black and white pictures. The process of describing hard images was done by technology developed by Amazon called Amazon Mechanical Turk, which pays cents per user describing an image. Today, with the improved computing power of processors and graphics processors, Deep Learning [4] has become a much more affordable method. The use of neural networks has been a success in many disciplines in view of the results obtained, in particular that of Krizhevsky used on images of ImageNet 4 and Cifar 5 databases [11,12]. The use of such image-pre-trained networks as feature generators has been shown to be more effective than the use of expert features for SVM [13]. In this way, Lagrange et al. [14] achieved excellent mapping results by combining superpixels and deep features in the Data Fusion Contest 2015 [15]. The use of convolutional neural network smulti-scale [16] then fully convolutive [17] haveallowed to improve this first approach. Indeed, these fully-convolutional networks (Fully-Convolutional Networks-FCN) learn not only the semantics of individual pixels, but also the spatial structures that connect them, making them particularly suitable for mapping from images. Many recent work has shown the effectiveness of the networks of fully convolutive neurons in this setting, following the principles introduced by Long et al. [18]. By densifying the last layers of a classical classifier network, it is possible to obtain either a vector of probabilities but heat maps indicating the probability of different classes in every way. Several subsequent adaptations have been proposed, such as the removal of subsampling [19] and layer expansion convolutive [20] to keep the resolution. Many models are also inspired by convolutional auto-encoders [21] and present a symmetrical architecture encoder/decoder, such as DeconvNet [22] and SegNet [23]. The integration of structured models such as Random Markovs were also studied [24,25]. Hassan et al. [26] developed a model that relies on the concept of visual attention that presents a considerable improvement over the simple transfer of learning CNN and developed an algorithms by combining a CNN and an RNN.

### 3. Related Work

Our approach is built on the recent CNN architectures for image classification [11,27–30] and learning. Firstly, the transfer was demonstrated from a set of visual tasks of recognition [31,32] then the detection and segmentation is performed same as [33]. We re-architect and refine the classification nets in direct prediction. We map the FCN space and link the previous models for segmentation and extract features by Multi-scale feature extraction method for recognition.

#### 3.1. Convolutional Neural Networks (CNNs)

CNNs are also called multi-layer neural networks, which are mostly used in pattern recognition tasks [34,35]. CNNs increase the robustness of the algorithm to low input variations. The low

pre-treatment rate necessary for the operations that do not require any choice of extractors of specific characteristics. The proposed architecture is based on different deep neural networks by introducing link between convolution and aggregation layers (pooling). These DNNs basically belong to different class of models inspired by the work of Hubel and Wiesel on the primary visual cortex of the cat [36]. The beginning of the architecture consisting of a succession of convolution and aggregation layers is dedicated to the automatic extraction of characteristics, while the second part, composed of completely connected layers of neurons, is dedicated to the classification.

### 3.1.1. Convolution Layer

The  $N$  number of convolution maps in  $M_j^i$  ( $j \in \{1, \dots, N\}$ ) parameterize the convolutional layer  $C_i$  (network layer  $i$ ),  $K_x \times K_y$  denotes the size of the convolution cores (often square) and the connection scheme to the previous layer is represented by  $L^{i-1}$ . Each convolution map  $M_j^i$  is the result of a convolution sum of the maps of the previous layer  $M_j^{i-1}$  by its respective convolution core. A bias  $b_j^i$  is then added and the result is passed to a nonlinear transfer function  $\Phi(x)$  [35]. In this case, the map of the completely connected layer can be calculated as follows

$$M_j^i = \Phi(b_j^i + \sum_{n=1}^N M_j^{i-1} * K_n^i) \quad (1)$$

where  $*$  denotes the convolution operator.

### 3.1.2. Transfer Learning

The Transfer learning [8] is a strategy that seeks to optimize performance on a learning machine from knowledge and other tasks performed by another learning machine. In practice, after Yosinski et al. [37], training a ConvNet model from the beginning (with initialization) is not recommended because training a ConvNet model requires a lot of data and takes a lot of time. On the other hand, it is more usual to use ConvNet models already training and to rehabilitate them for the problem, this is called transfer learning, which is about transferring learning from one model dealing with a problem to another type of problem. There are two types of transfer learning:

- The extraction of variables from the ConvNet: here, the ConvNet is used as an extractor [8,38], i.e., a vector is extracted from a certain layer of the model without modifying its structure or its weight and the previously extracted vector is used for a new spot.
- The fine tuning of the ConvNet model [39–41]: here, the new ConvNet is initialized with the weights and the structure of the pre-trained model is used. The structure of the pre-trained model is slightly modified for the new task and finally the new model is trained for the new task.

### 3.1.3. Max-Aggregation Layer

In conventional neural network architectures, the convolutional layers are followed by subsampling layers. A subsampling layer reduces the size of the cards and introduces invariance to rotations and translations that may appear as input. A max-pooling layer is a variant of a layer that has shown some advantages to the previous layers [42]. Max-aggregation layer gives the maximum activation value within the input layer for different regions of size  $K_x \times K_y$  without overlapping.

### 3.1.4. Classic Neural Layer

The convolution and pooling layers are parametrized with the activation cards of the last layer equal to 1, which results in a vector 1D of attributes. Completely connected classical layers composed of neurons are then added to the network to perform the classification. The classes can contain number of the desired neurons in the case of supervised learning.

### 3.2. Different Architectures of Convolutional neural networks (CNNs)

**AlexNet:** is the winner of ILSVRC 2012, obtained with the first position with a top-5 error of 16.4%. Krizhevsky et al. [11] achieved breakthrough in ILSVRC large-scale competition in 2012 by using Deep Convolutional Neural Network (DCNN). AlexNet is a scaled version of convolutional LeNet, it has 25 layers (including five convolutional layers). Here, CNN Features are generated for the iris recognition task by extracting 2 fully connected layers and the outputs of 5 convolutional layers.

**GoogLeNet:** is the winner of ILSVRC 2014, with a top-5 error of 6.7%. Szegedy et al. [27] introduced the Inception V1 architecture called GoogLeNet with the new insight using  $(1 \times 1)$  convolutional blocks to reduce and aggregate the number of feature. The improvement was added by employing more inception modules and redesigning the filter in this inception modules [43,44]. Here, the CNN Features are generated for the iris recognition task by extracting 12 inception layers and the outputs of 1 convolutional layer.

**VGG:** is the runner-up of ILSVRC 2014, Simonyan and Zisserman [28] used smaller filter  $3 \times 3$  in each convolutional layer for the performance improvement. Different versions of VGG have been introduced from the last 5 years but the 2 most popular are: VGG16 and VGG19 which contains 16 and 19 layers respectively. Here, CNN Features are generated for the iris recognition task by extracting 2 fully connected layers and outputs of the 16 convolutional layers.

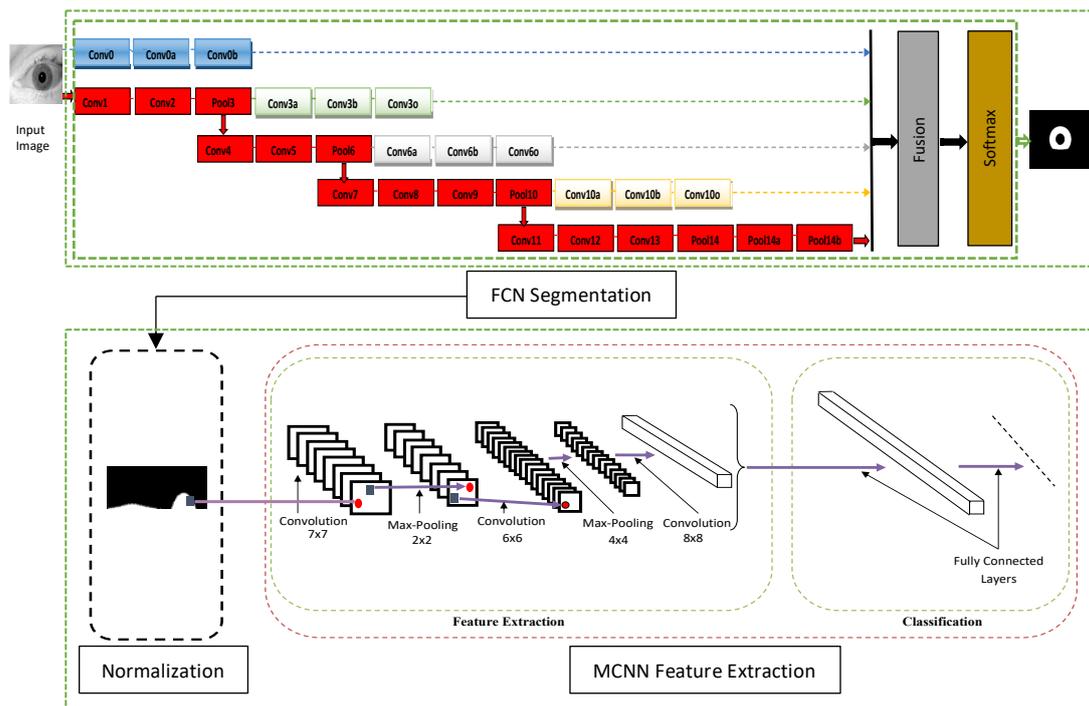
**ResNet:** is the winner of ILSVRC 2015, which is also a CNN from the ImageNet collection with an ensemble of residual networks. This network worn with only 3.6% error [29]. Here, the CNN Features are generated for the iris recognition task by extracting 17 bottleneck layers and the outputs of 1 convolutional layer.

**DenseNet:** In 2016, Huang et al. [30] proposed DenseNet which is used to connect each layer of CNN in a feed-forward fashion. Here, CNN Features are generated for the iris recognition task by extracting the outputs of a number selected 15 dense layers.

## 4. Proposed Method

### 4.1. Pre-Processing of Data

High resolution images can not be processed in a single pass by CNN because of their large size . As the average tile size of the Vaihingen ISPRS dataset is  $2493 \times 2063$ , while the majority of CNNs are limited to a  $256 \times 256$  pixel entry. Given the limitations of the amount of GPU memory, we divide the high-resolution images by slicing them out of a sliding window. In the case where the pitch of the sliding window is smaller than its size, the predictions are averaged over the pixels subject to recovery. This makes it possible to refine the predictions along the edges of the window and to smooth any discontinuities that may appear. Figure 1 shows flowchart of our proposed method.



**Figure 1.** The framework for iris recognition using FCN segmentation then Normalization, feature are next extracted by using MCNN.

#### 4.1.1. Segmentation Using FCN

In computer vision, segmentation is a process in which each coherent region of the image is assigned by a class. This can be done in particular by classifying each pixel of the eye image. The size of the images varies and is not necessarily square. So the size of  $256 \times 256$  is retained for the size of the input images. A padding is then performed on all the images of the base that they are this size.

In this work, we use FCN which reside of convolutional layers, pooling layers and unsampling, minimize computation time and number of parameters. FCN does not hold fully connected layers like traditional CNN. Furthermore, the network can work without each number fixed of original image size, FCN in input entire iris eye image without any repeated computation overlapped regions. Thus it make 2 bottlenecks of CNNs and segmentation iris very fast and more accurate. We can calculate the final output  $O$  by the equation bellow:

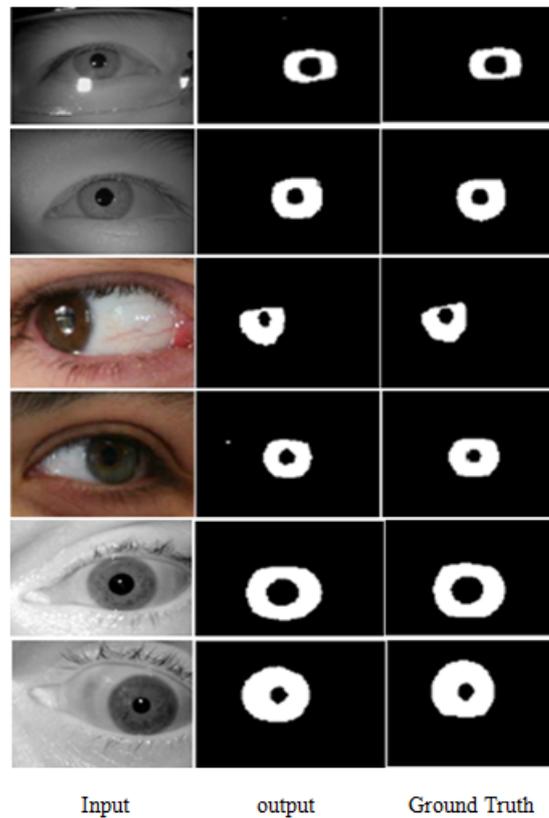
- Softmax layer

$$F_{(p,q)}^{(\theta)} = - \sum_{f=1}^M 1\{y_{(p,q)} = f\} \log \frac{\text{Exp}(\theta_f^T x_{(p,q)})}{\sum_{l=1}^M \text{Exp}(\theta_l^T x_{(p,q)})} \quad (2)$$

$F_{(p,q)}^{(\theta)}$  represents the output of softmax layer.  $\theta$  is parameter of the model,  $M$  represents the number of classes and  $f$  the  $i$ th number of column of class  $M$ .  $x_{(p,q)}$  represents the response at coordinate  $(p, q)$ . If  $O$  is the output of the heat map containing width, height and the number of label classes of  $m \times n \times c$  of the input image. So the loss of any pixel  $(p, q)$  between final output and the ground truth map can be calculated as follows

$$\text{loss} = \sum_{(p,q) \in O} F_{(p,q)}^{(\theta)} \quad (3)$$

Hence, whole loss at each pixel can be summarized to measure the accuracy. Figure 1 shows illustration of FCN introduced in this work, which contains 23 convolutional layers and 5 pooling layers. In our model, Fusion layer manipulates the sum of the output of these layers after upsampled to the input size. Figure 2 shows segmentation results obtained by FCNs.



**Figure 2.** Segmentation results by FCNs. The two first, two second and Two third rows are the three database: CASIA-Iris-Thousand, UBIRIS.v2 and LG2200, respectively. The first, second and third columns are the input images, output of our results and the ground truth, respectively.

#### 4.1.2. Normalization

For comparison between two irises, fixed size of the segmented iris region is aligned and perform the normalization by using Daugmans’s rubber sheet model [45]. The benefit of using such kind of model is that the circular region can easily be transformed to a rectangular shape. In this process, the center of the radial circle across the iris region is considered as the reference point. During the encoding process, the bit pattern which contain the information bits, corresponding noise mask based on corruption areas within the iris pattern and the mask bits as corrupted pattern are obtained. Each pixel of the iris in the Cartesian domain is assigned a correspondent in the pseudo-polar domain according to the distance of the pixel from the centers of the circles and the angle that it makes with these centers as shown in Figure 3. More precisely, the transformation is done according to the following equation:

$$x(r, \theta) = (1 - r)x_p(\theta) + rx_l(\theta) \tag{4}$$

$$y(r, \theta) = (1 - r)y_p(\theta) + ry_l(\theta) \tag{5}$$

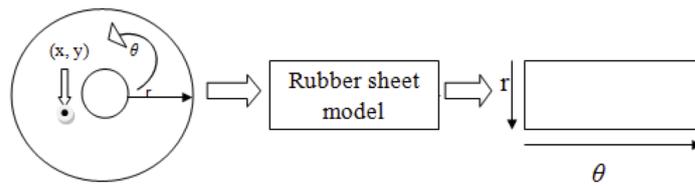


Figure 3. Daugman’s Rubber sheet model for the normalization of the iris.

Where  $x_p(\theta)$  denotes the point of the detected boundary of the segmented pupil and  $\theta$  is the angle of the center of the pupil.  $x_s(\theta)$  and  $y_s(\theta)$  denotes the coordinates of the points on contour of the iris, obtained by the same principle. Figure 4 shows a normalized image obtained by this process which is rectangular and of constant size, generally the chosen size is  $80 \times 512$  pixels. The width of the image represents the variation on the angular axis while the height represents the variations on the radial axis.



Figure 4. Normalized iris image.

#### 4.1.3. MCNN Feature Extraction

In this work, N number of convolutional neural networks with different retinal sizes (i.e., the size of the retina corresponds to the size of the input images). A given input image is resized N times to the retina in each network. The issue of optimization of classifier outputs is a recurring issue in the field of pattern recognition. For the recognition of iris, it has been shown [46] that a simple averaging of the outputs of each classifier makes it possible to obtain classification rates at least as good as with a linearly combined weights that can be learned via a cross validation method [47]. As the classifiers act at the same resolution [46], so there is no reason to weight them and also these classifiers are differ from each other in terms of input distortions. In this work, we assume that the classifier with the lowest resolution are not as important as the one with the highest resolution. The final results are therefore calculated based on the linearly combined N CNNs classifiers. Figure 5 shows the overall architecture of MCNN proposed in this work.

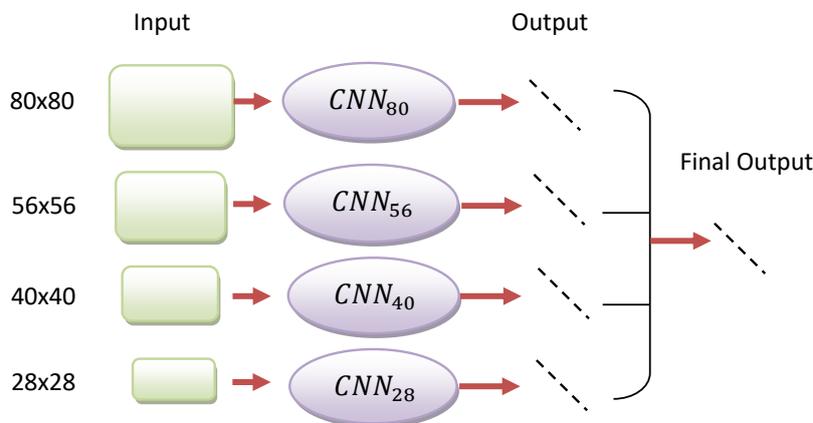


Figure 5. The global architecture of the multi-scale convolutional neural network (MCNN).

We build 4 CNNs acting at different resolutions. Here, we keep the size of the images with the resolution of  $80 \times 80$ . Which is then divided by a factor  $\sqrt{2}$ ,  $\sqrt[2]{2}$ ,  $\sqrt[3]{2}$  according to each axis to obtain respective images of size  $56 \times 56$ ,  $40 \times 40$  and  $28 \times 28$ . The networks called  $CNN_{80}$ ,  $CNN_{56}$ ,  $CNN_{40}$

and  $CNN_{28}$  are built similar to the LeNet model [35] and contain different sizes of the convolution kernel and retina. In our recognition system, we use 6 layers to make the extraction of characteristics for each CNN (Figure 6, Table 1) as follows:

- **Layer to enter:** A picture size  $28 \times 28$ ;
- **First convolution layer:** Number of convolution kernel (filters) is 6 of Size  $5 \times 5$ ; the result is a set of  $24 \times 24$  convolutional maps;
- **Subsampling layer:** number of maps is 6; kernel size:  $2 \times 2$ ; Size of the maps:  $12 \times 12$ ;
- **Second convolution layer:** Number of convolution kernel (filters) is 6 of Size  $5 \times 5$ ; the result is a set of  $8 \times 8$  convolutional maps;
- **Subsampling layer:** number of maps is 6; kernel size:  $2 \times 2$ ; Size of maps:  $4 \times 4$ ;
- **Third convolution layer:** Number of convolution kernel (filters) is 6 of Size  $4 \times 4$ ; the result is a set of  $1 \times 1$  convolutional maps.

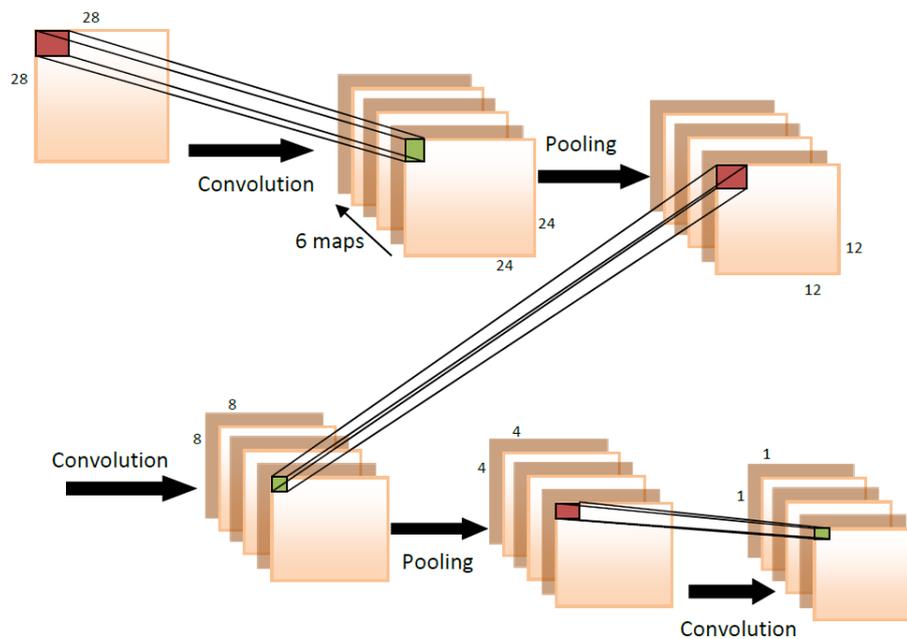
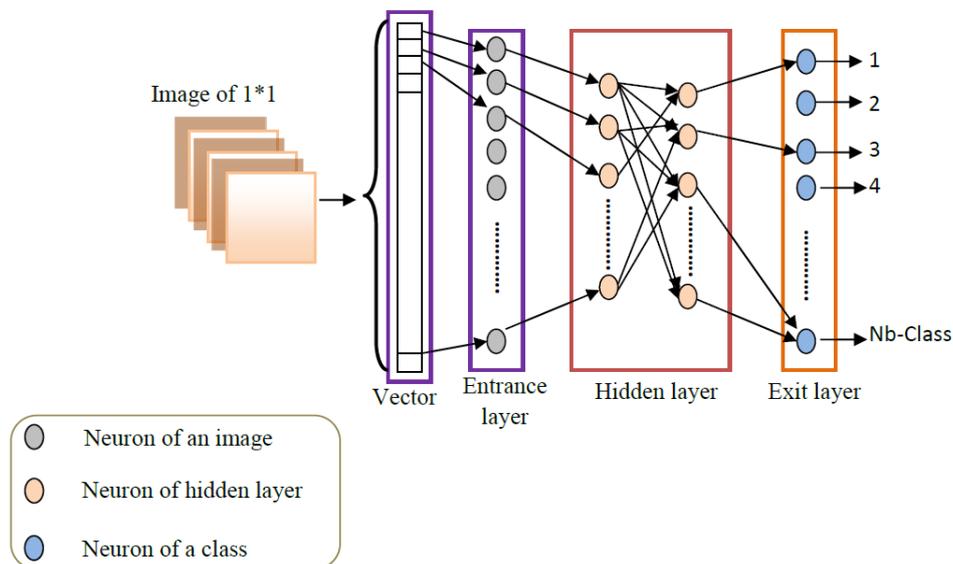


Figure 6. Example of Extraction phase by CNN.

Table 1. Network architectures in each box of the table; Top: size of the segmented eye image and Bottom: size of the map.

		$CNN_{80}$	$CNN_{56}$	$CNN_{40}$	$CNN_{28}$
Layer	Type	$80 \times 80$	$56 \times 56$	$40 \times 40$	$28 \times 28$
C1	Convolution	$7 \times 7$ $74 \times 74$	$7 \times 7$ $50 \times 50$	$5 \times 5$ $36 \times 36$	$5 \times 5$ $24 \times 24$
P2	Max-pooling	$2 \times 2$ $37 \times 37$	$2 \times 2$ $25 \times 25$	$2 \times 2$ $18 \times 18$	$2 \times 2$ $12 \times 12$
C3	Convolution	$6 \times 6$ $32 \times 32$	$6 \times 6$ $20 \times 20$	$5 \times 5$ $14 \times 14$	$5 \times 5$ $8 \times 8$
P4	Max-pooling	$4 \times 4$ $8 \times 8$	$4 \times 4$ $5 \times 5$	$2 \times 2$ $7 \times 7$	$2 \times 2$ $4 \times 4$
C5	Convolution	$8 \times 8$ $1 \times 1$	$5 \times 5$ $1 \times 1$	$7 \times 7$ $1 \times 1$	$4 \times 4$ $1 \times 1$

After the extraction, classification is performed to classify the images. Here, output in the last layer gives one neuron per class (each neuron represents a class of eyes of each person) (Figure 7).



**Figure 7.** Classification phase by CNN.

## 5. Experimental Results and Discussion

In this section, proposed model is applied on a task of segmentation and extraction of iris from eye images. It is compared on the scale of the images with the other state of the art methods on different databases. We begin by introducing the basics of databases and their composition and then experimental details are presented.

### 5.1. Databases

In this section, we show experimental results of our proposed method on three iris databases (Figure 8): CASIA-Iris-Thousand, UBIRIS.v2 and LG2200.

#### 5.1.1. CASIA-Iris-Thousand

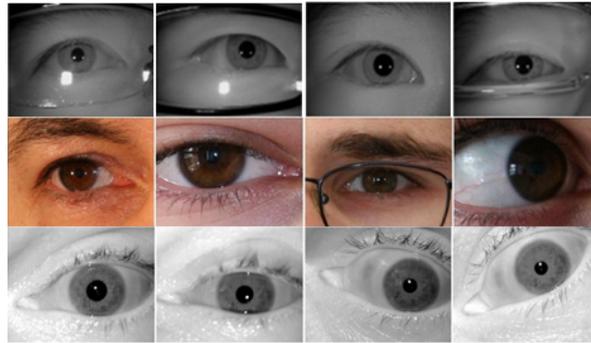
This database contains more than 20,000 iris images of about 1000 people with 20 images for each person, 10 images for the right iris and 10 images for the left iris with a resolution of  $640 \times 480$  pixels in JPG format, these images were collected using an IKEMB-100 camera produced by IrisKing, IKEMB-100 is an iris camera (double-eye) with a visual feedback, the bounding boxes indicated in the front LCD screen allow users to adjust their position for acquiring the image of the iris with high quality [48].

#### 5.1.2. UBIRIS.v2

The UBIRIS [49] database was published by SOCIA Lab (Soft Computing and Image Analysis Group of the University of Beira Interior, Portugal) which was established in 2004 for the purpose of evaluating images taken under uncontrolled conditions. The peculiarity of this database is that the images are acquired in visible contrary to the usual databases that are acquired in NIR. UBIRIS-v2 [50] contains a total of 11,102 images resolution  $300 \times 400$  pixels in TIFF format with a typical camera Canon EOS 5D Camera representing 500 subjects of different world origins (70 countries). UBIRIS-v2 is acquired remotely (between 4 and 8 m).

### 5.1.3. LG2200

This database was originally published for the CrossSensor iris recognition which is a challenge. The ND-CrossSensor consists of 27 data sessions with 676 people, a session contains on average 160 unique people who have multiple images. There are 116,564 images captured with the LG2200 device, each person is at least in two sessions on the entire dataset. The resolution of the image is  $640 \times 480$  pixels [51].



**Figure 8.** Sample images from the three databases: CASIA-Iris-Thousand (first row), UBIRIS.v2 (second row) and LG2200 (third row).

## 5.2. Discussion

The proposed model avoids manual extraction of features and is more robust in design as compared to the traditional approaches which typically unable to address difficult problems. The approach of using multiple convolutional neural networks for the same task has already been used in [46]. The idea behind this is that each network will learn different characteristics of the same input. To reach this goal, the authors of [47] apply different distortions (color, shape) to the learning set. This approach cannot be used here since the application of distortions can change their chromatin size or distribution, which is not viable for classification. Our model contain the following advantages:

- The design of feature extraction is robust and easy to calculate.
- Perform the selection, calculation and evaluation of the relevant characteristics and their relevance for class separation at different level.
- Avoid and circumvents the difficulties occurred in delicate step related to characteristics.

## 5.3. Results

Here all the experiments were carried out several times to ensure reliability of the results. The error rates presented in Table 2 are the average error rates calculated for each achievement. As expected, these rates decrease when the resolution of input images increases because high resolutions are able to capture more information.

Table 2 presents the numerical results obtained on different stages in our method. The network is learned only on “network-images” to the results obtained when the network is refined on the entire images. Regarding the transition to learning whole images, we have noted that the results produced by the refined network on the images contain fewer false positives.

**Table 2.** Error Rate evaluation of different architectures used in our Method.

Method	Error Rate (%)		
	UBIRIS.v2	LG2200	CASIA-Iris-Thousand
CNN <sub>28</sub>	1.07%	1.35%	1.21%
CNN <sub>40</sub>	1.02%	1.29%	1.10%
CNN <sub>56</sub>	0.98%	1.17%	1.02%
CNN <sub>80</sub>	0.89%	1.11%	0.93%
SVM	0.95%	1.07%	0.79%
MCNN	0.85%	1.00%	0.71%
FCN	0.56%	0.96%	0.63%

The left and right iris images of each subject are supposed to two different classes. At random, we choose 70% data for training of each class and 30% for testing.

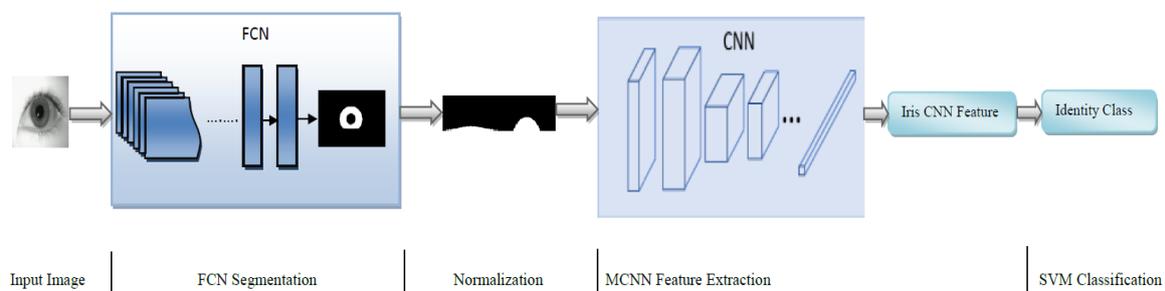
To segment the iris, first we use FCN and then normalize the segmented region into a rectangle of size  $64 \times 256$  pixels by using Daugman’s rubber sheet method [45]. Then the normalized iris region is fixed into the size  $256 \times 256$  and extracted biometric features are employed by using MCNN to define and extract local patterns that fit the iris texture. To increase the recognition accuracy of the proposed method.

The experimental results show highest accuracy and the best accuracy rates on the three databases with the FCN and MCNN classification. Our approach FCN achieve average segmentation error rate of 0.63%, 0.56% and 0.96% on CASIA-Iris-Thousand, UBIRIS.v2 and LG2200 respectively, and MCNN achieve average feature extraction error rate of 0.71%, 0.85% and 1% respectively, by using Equation (6), which not only exceeds the performance of traditional methods and other previously proposed deep learning-based architectures, but is competitive with the most state-of-art iris recognition algorithms.

$$Error_{rate} = \frac{1}{N \times m \times n} \sum_{x,y \in (m,n)} G_{(x,y)} \oplus M_{(x,y)} \tag{6}$$

where  $N, m, n$  are the number of width and length of the images tested, respectively.  $G$  is the ground truth and  $M$  is the generated iris mask, and  $x, y$  are the coordinates of  $G$  and  $M$ . The  $\oplus$  represents the XOR operation in equation for evaluate the quarreling pixels between  $G$  and  $M$ .

In addition, we have also compared the results by a multiclass Support Vector Machine(SVM) classifier [52], which have a depth of for kernel output or feature space and linear combination that produces the output with the pre-trained of MCNN without do modified (Figure 9). The training images are used to train the SVM [52]. To make the comparison, we used the same testing and training data. Our proposed MCNN classifier shows lower error than the SVM because the optimal features can be easily extracted with the more convolutional layers.



**Figure 9.** The framework for iris recognition using SVM Classification.

### 5.4. Performance Metric and Baseline Method

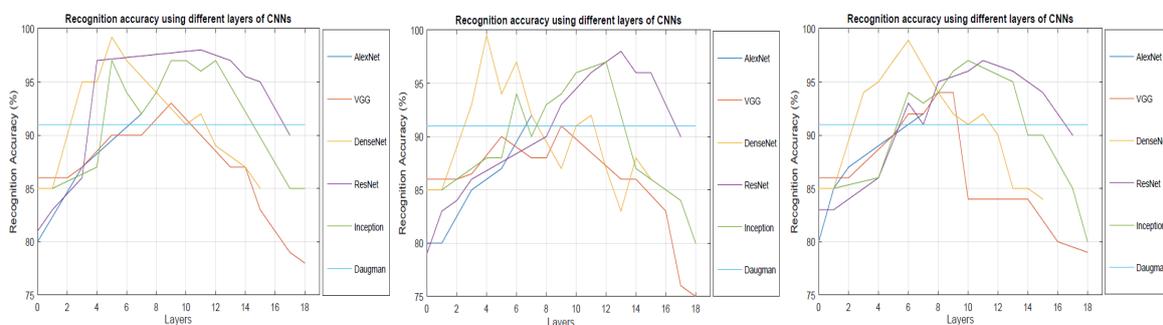
In our proposed work, recognition rate is reported at FAR = 0.1% and used as the baseline feature descriptor for performance comparison. For matching, 1D-Log Gabor with the matching operator Adaptive Hamming Distance is used same as our previous work [53]. Our method shows recognition accuracy of 95.63%, 99.41% and 93.17% and FRR 4.27%, 0.49% and 6.73% on three databases: CASIA-Iris-Thousand, UBIRIS.v2 and LG2200, respectively. Quantitative results between the previous methods and our proposed method are shown in (Table 3).

**Table 3.** Recognition Accuracy comparison of the proposed method with previous methods.

		Recognition Accuracy (%)		
Methods		UBIRIS.v2	LG2200	CASIA-Iris-Thousand
Liu et al. [54]	HCNNs and MFCNs	98.40%	-	-
Zhao et al. [55]	AttNet+FCN-peri	98.79 %	-	-
Bazrafkan et al. [56]	End to End FCDNN	99.30%	-	-
Nguyen et al. [57]	CNN	-	90.7%	91.1%
Our proposed	FCN + MCNN	99.41%	93.17%	95.63 %

### 5.5. Performance Analyses

For the Qualitative performance analysis, we use three databases: CASIA-Iris-Thousand, UBIRIS.v2 and LG2200. For three datasets analysis are performed and tested on different layers. Results in terms of recognition accuracy peaks are shown in Figure 10. In some middle layers for all CNNs, for CASIA-Iris-Thousand dataset: VGG on layer 9, Inception on layer 10, ResNet on layer 11 and DenseNet on layer 5. For the UBIRIS.v2 dataset: VGG on layer 9, Inception on layer 10, ResNet on layer 13 and DenseNet on layer 4. For the LG2200 dataset: VGG on layer 9, Inception on layer 10, ResNet on layer 11 and DenseNet on layer 6. All this three databases uses rather low Daugman matcher rate of 91%. As inception layers is done on layer 10, it converges to the peak more quickly than other layers. ResNet allow the gradient to flow throughout the network due to its property of skip connections. Due to gradient flow condition, ResNet make the network performance much better in terms of iris recognition accuracy. DenseNet allows neurons to interact easily due to its rich dense connections. It can observed that peak results do not occur toward the later layers of the CNNs (Figure 10). The main benefit of normalization is that the original iris images are more complex compared to the normalized iris image. Hence, encoding the normalized iris image do not need large number of layers for the process and the accuracy can be achieved in the middle layers. Among all five CNNs, DenseNet achieves the best recognition accuracy of 99.2% at layer 6 on the CASIA-Iris-Thousand database and 99.5% at layer 4 on the UBIRIS.v2 database and 98.9% at layer 6 on the LG2200 database.



**Figure 10.** Quantitative analysis by comparison with Recognition accuracy different layers of CNNs for three databases from left to right respectively: CASIA-Iris-Thousand dataset,UBIRIS.v2 and LG2200.

## 6. Conclusions

In this paper, we present an iris segmentation and recognition method based on a set of convolutional neural networks (CNN) and on several networks acting at different resolutions. Our approach avoids the tedious step of manual segmentation using FCN and extraction of features using MCNN. The proposed approach show the efficiency of convolutional networks on iris recognition and our results achieves the best classification rates on different three databases: CASIA-Iris-Thousand, UBIRIS.v2 and LG2200 compared to conventional methods of the state of the art. Future work is moving towards the implementation of best attribute extractors for classification (residual network models, wide area networks, etc.). The application of the proposed methods to other modalities of iris database is also considered.

**Author Contributions:** Conceptualization, R.T.; Formal analysis, R.T., W.D. and N.A.; Funding acquisition, W.D.; Investigation, R.T. and N.A.; Methodology, N.A.; Project administration, W.D.; Software, R.T. and N.A.; Supervision, W.D.; Validation, W.D.; Writing—original draft, R.T.; Writing—review & editing, N.A.

**Funding:** The work presented in this paper was supported by the National Natural Science Foundation of China under Grant no. 61370201.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

CNN	Convolutional Neural Network
DNN	Deep Neural Networks
DCNN	Deep Convolutional Neural Network
FCN	Fully Convolutional Network
FCDNN	Fully Convolutional Deep Neural Network
GPUs	Graphics Processing Units
HMM	Hidden Markov Model
HCNNs	heterogeneous Convolutional Neural Networks
ILSVRC	ImageNet Large Scale Visual Recognition Competition
MCNN	Multi-scale Convolutional Neural Network
RNN	Recurrent neural network
SVM	Support Vector machine

## References

- Zhu, C.; Sheng, W. Multi-sensor fusion for human daily activity recognition in robot-assisted living. In Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, La Jolla, CA, USA, 9–13 March 2009; pp. 303–304. [\[CrossRef\]](#)
- Ghosh, A.; Riccardi, G. Recognizing human activities from smartphone sensor signals. In Proceedings of the ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 865–868. [\[CrossRef\]](#)
- Rahman, S.A.; Merck, C.; Huang, Y.; Kleinberg, S. Unintrusive eating recognition using Google Glass. In Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare, Istanbul, Turkey, 20–23 May 2015; pp. 108–111. [\[CrossRef\]](#)
- Bengio, Y.; Goodfellow, I.; Courville, A. *Deep Learning*; MIT Press: Cambridge, London, UK, 2015. preparation.
- Ouchi, K.; Doi, M. Smartphone-based monitoring system for activities of daily living for elderly people and their relatives etc. In Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication, Zurich, Switzerland, 8–12 September 2013; pp. 103–106. [\[CrossRef\]](#)
- Aparicio, M., IV; Levine, D.S.; McCulloch, W.S. Why are neural networks relevant to higher cognitive function. *Neural Network Knowledge Represent Inference*; Lawrence Erlbaum Associates, Inc.: Hillsdale, NJ, USA, 1994; pp. 1–26.
- Parizeau, M. *Neural Networks*; GIF-21140 and GIF-64326; Laval University: Québec, QC, Canada, 2004.
- Karpathy, A. Neural Networks Part 1: Setting Up the Architecture. *Notes for CS231n Convolutional Neural Networks for Visual Recognition*; Stanford University: Stanford, CA, USA, 2016.

9. Ng, A. Machine Learning Yearning. *Technical strategy for AI Engineers, In the Era of Deep Learning*; Stanford University: Stanford, CA, USA, 2016.
10. LeCun, Y.; Jackel, L.D.; Bottou, L.; Cortes, C.; Denker, J.S.; Drucker, H.; Guyon, I.; Muller, U.A.; Sackinger, E.; Simard, P.; et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Netw. Stat. Mech. Perspect.* **1995**, *261*, 276.
11. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
12. Krizhevsky, A.; Hinton, G.E. Convolutional Deep Belief Networks on Cifar-10. Available online: <https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf> (accessed on 1 April 2017).
13. Penatti, O.A.B.; Nogueira, K.; dos Santos, A.J. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 44–51. [[CrossRef](#)]
14. Lagrange, A.; Le Saux, B.; Beaupere, A.; Boulch, A.; Chan-Hon-Tong, A.; Herbin, S.; Randrianarivo, H.; Ferecatu, M. Benchmarking classification of earthobservation data: From learning explicit features to convolutional networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 4173–4176. [[CrossRef](#)] [[CrossRef](#)]
15. Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Tiede, D.; d’Oleire-Oltmanns, S.; Baraldi, A.; Shimoni, M.; Moser, G.; Tuia, D. Processing of Extremely High Resolution LiDAR and RGB Data: Outcome of the 2015 IEEE GRSS Data Fusion Contest—Part B: 3-D Contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 5560–5575. [[CrossRef](#)]
16. Zhao, W.; Du, S. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 155–165. [[CrossRef](#)]
17. Marmanis, D.; Wegner, J.D.; Galliani, S.; Schindler, K.; Datcu, M.; Stilla, U. Semantic Segmentation of Aerial Images with an Ensemble of CNNs. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 473–480. [[CrossRef](#)]
18. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [[CrossRef](#)]
19. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]
20. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
21. Zhao, J.; Mathieu, M.; Goroshin, R.; LeCun, Y. Stacked What-Where Auto-encoders. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, USA, 2–4 May 2016.
22. Noh, H.; Hong, S.; Han, B. Learning Deconvolution Network for Semantic Segmentation. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Washington, DC, USA, 7–13 December 2015; pp. 1520–1528. [[CrossRef](#)]
23. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**. [[CrossRef](#)]
24. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H. Conditional Random Fields as Recurrent Neural Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Washington, DC, USA, 7–13 December 2015; pp. 1529–1537. [[CrossRef](#)]
25. Arnab, A.; Jayasumana, S.; Zheng, V.; Torr, P. Higher Order Conditional Random Fields in Deep Neural Networks. *Eur. Conf. Comput. Vis.* **2016**. [[CrossRef](#)]
26. Hasan, S.A.; Ling, Y.; Liu, J.; Sreenivasan, R.; Anand, S.; Arora, T.R.; Datla, V.V.; Lee, K.; Qadir, A.; Swisher, C.; et al. PRNA at ImageCLEF 2017 Caption Prediction and Concept Detection Tasks; In Proceedings of the Working Notes of CLEF 2017—Conference and Labs of the Evaluation Forum, Dublin, Ireland, 11–14 September 2017.
27. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]

28. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
30. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [[CrossRef](#)]
31. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. DeCAF: A deep convolutional activation feature for generic visual recognition. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
32. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 818–833. [[CrossRef](#)]
33. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [[CrossRef](#)]
34. Nebauer, C. Evaluation of convolutional neural networks for visual recognition. *IEEE Trans. Neural Netw.* **1998**, *9*, 685–696. [[CrossRef](#)]
35. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
36. Hubel, D.H.; Wiesel, T. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **1962**, *160*, 106–154. [[CrossRef](#)] [[PubMed](#)]
37. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the NIPS 2014—Neural Information Processing Systems Conference, Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
38. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. Cnn features off-the-shelf: An astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 Jun 2014; pp. 806–813. [[CrossRef](#)]
39. Shankar, S.; Garg, V.K.; Cipolla, R. Deep-carving: Discovering visual attributes by carving deep neural nets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 Jun 2015; pp. 3403–3412. [[CrossRef](#)]
40. Huval, B.; Coates, A.; Ng, A. Deep learning for class-generic object detection. *arXiv* **2013**, arXiv:1312.6885.
41. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.; Ciompi, F.; Ghafoorian, M.; Van Der Laak, J.A.; Van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Elsevier Med. Image Anal.* **2017**, *42*, 60–88. [[CrossRef](#)] [[PubMed](#)]
42. Scherer, D.; Müller, A.; Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 92–101. [[CrossRef](#)]
43. Bowyer, K.W.; Hollingsworth, K.; Flynn, P.J. Image understanding for iris biometrics: A survey. *Comput. Vis. Image Understand.* **2008**, *110*, 281–307. [[CrossRef](#)]
44. Bowyer, K.W.; Hollingsworth, K.; Flynn, P.J. A survey of iris biometrics research 2008–2010. In *Handbook Iris Recognition*; Springer: London, UK, 2013. [[CrossRef](#)]
45. Daugman, J. The importance of being random: Statistical principles of iris recognition. *Pattern Recognit.* **2003**, *36*, 279–291. [[CrossRef](#)]
46. Meier, U.; Claudiu Ciresan, D.; Gambardella, L.M.; Schmidhuber, J. Better digit recognition with a committee of simple neural nets. In Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; pp. 1250–1254.
47. Ueda, N. Optimal linear combination of neural networks for improving classification performance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, 18–21. [[CrossRef](#)]
48. Chinese Academy of Sciences; Institute of Automation. Biometrics Ideal Test. CASIA iris Database. Available online: <http://biometrics.idealtest.org/> (accessed on 1 April 2017).
49. Proença, H.; Alexandre, L.A. UBIRIS: A noisy iris image database. In *Proceedings of Image Analysis and Processing—ICIAP 2005*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3617, pp. 970–977. [[CrossRef](#)]

50. Proença, H.; Filipe, S.; Santos, R.; Oliveira, J.; Alexandre, L.A. The UBIRIS.v2: A database of visible wavelength iris images captured on-the-move and at-a-distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1529–1535. [[CrossRef](#)]
51. Phillip, P.J.; Scruggs, W.T.; O'Toole, A.J.; Flynn, P.J.; Bowyer, K.W.; Schott, C.L.; Sharpe, M. FRVT 2006 and ICE 2006 large-scale experimental results. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 831–846. [[CrossRef](#)]
52. Tang, Y. Deep Learning using Linear Support Vector Machines. In Proceedings of the International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013.
53. Tobji, R.; Di, W.; Ayoub, N.; Haouassi, S. Efficient Iris Pattern Recognition method by using Adaptive Hamming Distance and 1D Log-Gabor Filter. (*IJACSA Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 662–669. [[CrossRef](#)]
54. Liu, N.; Zhang, M.; Li, H.; Sun, Z.; Tan, T. Deepiris: Learning pairwise filter bank for heterogeneous iris verification. *Pattern Recognit. Lett.* **2016**, *82*, 154–161. [[CrossRef](#)]
55. Zhao, Z.; Kumar, A. Towards More Accurate Iris Recognition Using Deeply Learned Spatially Corresponding Features. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3809–3818. [[CrossRef](#)]
56. Bazrafkan, S.; Thavalengal, S.; Corcoran, P. An End to End Deep Neural Network for Iris Segmentation in Unconstraint Scenarios. *ELSEVIER Neural Netw.* **2018**, *106*, 79–95. [[CrossRef](#)]
57. Nguyen, K.; Fookes, C.; Ross, A.; Sridharan, S. Iris Recognition With Off-the-Shelf CNN Features: A Deep Learning Perspective. *IEEE Access* **2018**, *6*, 18848–18855. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).