IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 24, 2019, accepted February 19, 2019, date of publication February 27, 2019, date of current version March 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901235

Fingerprint Liveness Detection Using an Improved CNN With Image Scale Equalization

CHENGSHENG YUAN[1,2,3], ZHIHUA XIA[1,2], (Member, IEEE), LEQI JIANG[1,2], YI CAO[1,2], Q. M. JONATHAN WU[3], (Senior Member, IEEE), AND XINGMING SUN[1,2], (Senior Member, IEEE)

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada

Corresponding author: Zhihua Xia (xia_zhihua@163.com)

This work was supported in part by the National Key R& D Program of China under Grant 2018YFB1003205, in part by the National Natural Science Foundation of China under Grant U1836208, Grant U1536206, Grant U1836110, Grant 61602253, and Grant 61672294, in part by the Jiangsu Basic Research Programs-Natural Science Foundation under Grant BK20181407, in part by the Canada Research Chair Program and the NSERC Discovery Grant, in part by the Jiangsu Postgraduate Research and Innovation Program under grant KYCX17_0899, in part by the State Scholarship Fund, China, under Grant 201708320316, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions Fund, and in part by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology Fund, China.

ABSTRACT Due to the lack of pre-judgment of fingerprints, fingerprint authentication systems are frequently vulnerable to artificial replicas. Anonymous people can impersonate authorized users to complete various authentication operations, thereby disrupting the order of life and causing tremendous economic losses to society. Therefore, to ensure that authorized users' fingerprint information is not used illegally, one possible anti-spoofing technique, called fingerprint liveness detection (FLD), has been exploited. Compared with the hand-crafted feature methods, the deep convolutional neural network (DCNN) can automatically learn the high-level semantic detail via supervised learning algorithm without any professional background knowledge. However, one disadvantage of most CNNs models is that fixed scale images (e.g., 227×227) are essential in the input layer. Although the scale problem can be handled by cropping or scaling operations via transforming an image of any scale into a fixed scale, they can easily cause some key texture information loss and image resolution degradation, which will weaken the generalization performance of the classifier model. In this paper, a novel FLD method called an improved DCNN with image scale equalization, has been proposed to preserve texture information and maintain image resolution. Besides, an adaptive learning rate method has been used in this paper. In the performance evaluation, the confusion matrix is applied into FLD for the first time as a performance indicator. The amounts of the experimental results based on the LivDet 2011 and LivDet 2013 data sets also verify that the detection performance of our method is superior to other methods.

INDEX TERMS Fingerprint liveness detection, supervised learning, biometrics, spoof detection, adaptive learning rate.

I. INTRODUCTION

With the rapid development of multimedia technology, it is possible for us to capture amounts of high quality images with the aid of some sophisticated high-resolution images acquisition devices, meanwhile we frequently use diverse biometrics to login or confirm users' information. Among them, because of the characteristics of rapidity and convenience, authentication systems based on fingerprints are widely exercised in fingerprint quick payment, fingerprint boot, fingerprint attendance, etc. Trouble is that these personal fingerprints are not always stable and safe, that is to say, authorized users' fingerprints could be copied under the users' cooperation. Except the collaborative approach, real fingerprints are able to be imitated by illegal attackers using fingerprint membranes of authorized users left on the surface of the object. Both above methods can impersonate the identity of authorized users and realize diverse

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

authentication operation that only authorized users can perform. Thus, how to protect our biometrics and distinguish correctly live or fake fingerprint samples has become an urgent demand in fingerprint recognition application. For those traditional knowledge-based authentication methods, users have to remember or use some relevant passwords, secret questions or tokens to login or access to personal accounts [1]. However, it is easy for us to forget these knowledge after long periods of no use. The biometric based authentication methods solve the deficiency of traditional schema, and they are getting more and more popular. Common biometrics include face, fingerprint, vein, palm print, pupil, iris, etc., and fingerprint is one of the oldest and most mature biological features among them. Advocates of fingerprinting technology believe that fingerprint recognition technology can provide higher security and simplicity than techniques that use identification codes to identify identity information. However, in recent years, with the emergence of various emerging scam technologies, these systems confront many severe safety problem and new challenges. Such as, they are easily spoofed by artificial replicas [2], [3] produced using common materials such as silicone, latex and wood glue with the help of cooperative or non-cooperative devices, and amounts of studies and literatures have also confirmed this problem. In summary, one outstanding fingerprint authentication system can not only verify the user's identity, but also correctly differentiate the activity of fingerprints. Based on the overhead analysis, how to protect fingerprint information of authorized users and prevent artificial replicas attacking fingerprint authentication systems has become a research hotspot.

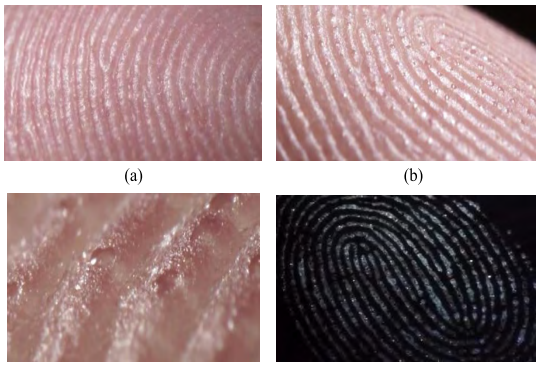


FIGURE 1. Sweat samples (live fingerprints). (a) Original fingerprint. (b) Sweat fingerprint. (c) Magnified Sweat fingerprint. (d) Captured sweat fingerprint using sensor.

In recent years, scholars have devoted considerable effort to put forward various reliable ways to counter spoof attacks, and fingerprint liveness detection (FLD) is one of these countermeasures [14]–[20].

As shown in Figure 1, fingerprint image consists of some alternating ridges and valleys. There are such a characteristic for ridges that pattern of each individual is all different, unique and constant throughout life. Figure 1 (a) shows a live fingerprint, and we can notice that sweat goes through the pores of sweat glands with the help of microscope in Figure 1 (b) as well as Figure 1 (c) is a magnified finger. Finger has many pores on the ridges, which are actually small openings in the skin from sweat glands, hair follicles, and sebaceous glands deep within the dermis. Those phenomena described only appear in live fingerprints, however, there are no such similar features in those artificial replicas. Hereby, FLD techniques are proposed based on above phenomenon. Especially since 2009, for global scholars, a fingerprint liveness detection competition is held every two years, and it has been successfully held for 5 sessions so far. The purpose of this competition is to encourage more scholars to participate in the study of FLD, to further enhance the security of personal identity information and minimize social losses.

After the research and analysis of the existing FLD methods, they are broadly categorized into two main stream anti-spoofing methods: hardware-based and software-based methods. The previous methods need to measure inherent properties of the given fingerprints utilizing some auxiliary sensor devices [4]–[7], such as skin distortion, oxygen saturation or odor. Although they can discriminate the real and spoof fingerprints, these auxiliary devices add to the cost of the authentication systems [8]. Moreover, the stability of the hardware-based method is relatively weak, since fingerprints are highly susceptible to interference of external environment, thus, the detection performance based on hardware method is still not applicable to the detection of harsh environment. Meanwhile, stains, breakage and dryness on the fingers can also weaken the detection performance of the measurement devices. To save the cost and enhance the detection performance, software-based detection methods, differentiating live and fake fingerprints only using image processing technology without any extra sensor devices, are applied to FLD. Fake fingerprints, for example, are distinguished from authentic ones by analyzing and extracting a better dynamic or static features of the fingerprint samples. Compared with hardware-based methods, only a single fingerprint rather than fingerprints sequences [9] are used to detect the fingerprint liveness. As mentioned at the beginning of this paper, the patterns of real and fake fingerprint images are different, thus, texture patterns based features extraction algorithms are the most common methods to distinguish live from fake fingerprints in software-based approaches.

Texture properties of live fingerprints in continuity, clarity and ridge strength are better than artificial replicas, so we can differentiate the live fingerprints from the specified fingerprint images sets. Over the years, many texture descriptor algorithms are generated in FLD, such as LBP, SIFT, LPQ, etc. Meanwhile, some improved algorithms based on this basis have been proposed. Different from those basis methods, the improved algorithms have better detection performance and stronger robustness. Local Binary Pattern (LBP) is a gray-scale and rotation invariant texture descriptors [10], and it has been widely utilized in image segmentation [11], camera identification [12], or image retrieval [18], etc. LBP is first used in the FLD [13], in which energy of wavelet-domain

transformation is complemented by the LBP descriptor. After that, many improved LBP algorithms, such as LBP with wavelets [27], uniform LBP coding schemes, are all applied to FLD, meanwhile those methods can obtain some satisfactory results. Gragnaniello *et al.* [24] constructed a discriminative texture descriptors via calculating the orientation component and differential excitation of each pixel block, and their method was called Weber Local Descriptor (WLD). It is noteworthy that WLD is a robust to illumination change and powerful texture descriptor, and it is more adapted for high-contrast patterns. Eventually, training model based on statistical joint histograms of orientation components and differential excitation is obtained with the help of SVM classifier with a linear kernel function. A new local feature representation associated with fingerprints is extracted in [25], and a bi-dimensional contrast-phase histogram is formed by calculating information on the phase (frequency domain) and the local amplitude contrast (spatial domain). The local phase for the purpose of FLD is used in [22]–[28], which is similar to LBP. Reference [7] proposed a novel fingerprint liveness detection based on Binarized statistical image features (BSIF), representing those texture descriptors using the statistics of fingerprint patches and maximizing the statistical independence of the filter responses rather than a fixed set of filters. Feature fusion based fingerprint liveness detection method has been proposed in [8], in which two features based on convolutional neural networks (CNN) with random weights and Local Binary Patterns (LBP) are extracted respectively. Features of a set of filters are automatically extracted using independent component analysis (ICA) in [7], and then, these features are fed into a support vector machine (SVM).

Until now, most of the aforementioned algorithms are all based on handcrafted features representations. Thus, the key to FLD is how to extract better feature representations of given images, while handcrafted feature extraction mainly relies on the experience and professional knowledge. Moreover, due to loss of spatial location information and lack of considerations for the details of live fingerprint images, it is difficult to achieve a balance between discrimination and robustness of methods. Recently, however, deep learning strategies for object classification and analysis of big data achieved great success and attracted widespread concern in the field of pattern recognition and computer vision. On one hand, because of different types of fingerprint sensors, it is possible to collect fingerprint samples of different scales. On the other hand, different strengths pressed against the surface of fingerprint sensor will eventually yield different sizes of fingerprint images. Existing CNNs methods can achieve a better object classification or detection task via extracting the high-level semantic features of images, but they [22] all are based on a fixed scale image and do not make full use of image scale information. As we all know, in the CNN, convolution and pooling operations have no restriction on the sizes of input images, and while they only affect the scales of features maps of each convolution and pooling layers.

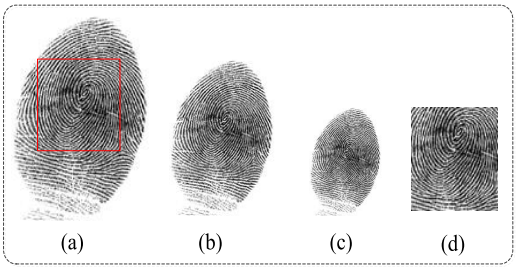


FIGURE 2. Fingerprints of cropping and scaling operations.

However, the full connection layer has strict limits on the scale of the output image of the previous convolution layer or pooling layer. That is, the neurons of the full connection layer are fully connected to each neuron of the output layer. Since the number of neurons is constant, the input image scale of the full connection layer are fixed. To solve the problem of different scales in CNN, cropping and scaling are two most common operations. But the question is that the cropping is easy lose some texture information, presenting the red window in Figure 2(a) (The size of the whole image is 312×372). After cropping, the area inside the red border is preserved, while the outside texture information is discarded. Figure 2(d) also denotes that fingerprint image is not complete after cropping. And the spatial structures after scaling operation are very susceptible to image deformation and resolution, shown in Figure 2(b) (The size of image is 156×186). To satisfy the requirements of the fixed scale of the input image, the image scaling is another executable operation. After scaling, the resolution of the image is reduced obviously. If you rescale the image once, the resolution of the image will be further reduced. As shown in Figure 2(c) (The size of image is 78×93), it has been difficult to visually observe adjacent ridges and valleys in the fingerprint image. As we all know, human eye structure is a complex and powerful neural network. If the human eye hardly distinguish fingerprints, neither will computers. Thus, classification performance is compromised due to lack of effective discriminant information.

As discussed above, the full connection layers have strict limits on the scale of the input image. Inspired by this constraint, could we design a new structure of CNN that diverse images scales can be converted to fixed length representations before entering the full connection layer? Hereby, to eliminate image scale limitations, in this paper, we propose a novel fingerprint liveness detection method based on an improved convolutional neural network with Image Scale Equalization. Finally, images of any size can be expressed in fixed length vectors without any cropping or scaling operations. Moreover, our method is also robust to fingerprint image deformation without scaling operation [27], [32], [33]. The flowchart of this paper is shown in Figure 3. In the training phase of the model, fine-tune the model parameters based on two fingerprint samples sets, testing based on learning rate adaptive adjustment and testing based on model

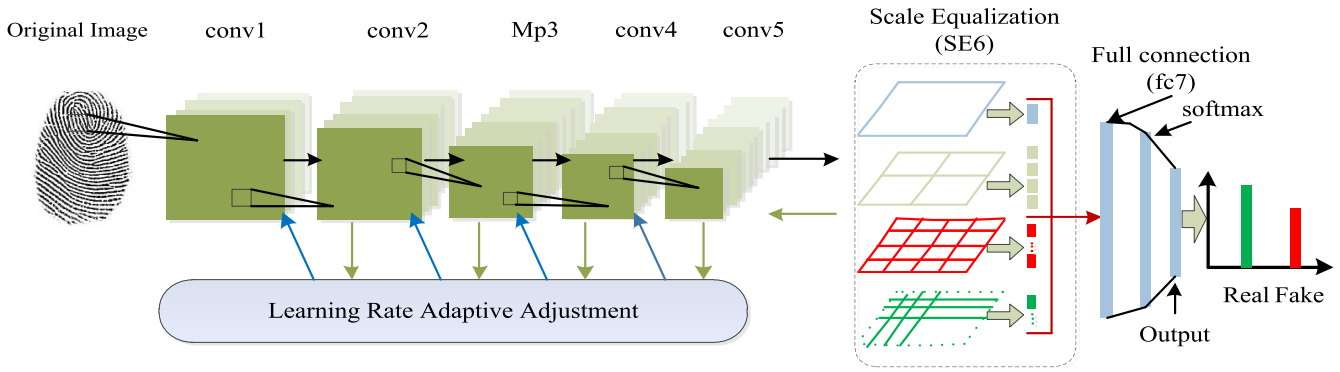


FIGURE 3. Flowchart of our network model structure.

weight initialization have been implemented in this paper. To improve the generalization performance of the training model, many experiments based on multi-scale Equalization have also been done. It is noteworthy that the proposed model structure in this paper is a supervised learning. That is to say, in the forward propagation, the output will be compared with the corresponding real label, and the parameters of the model are continually modified by the reverse retransmission of the gradient derivation. When the difference between the output value and the predicted value is less than a given threshold, the trained model will be the optimal model classifier at that time. In addition, data augmentation of training sets is necessary to prevent over-fitting. Specifically, the major contributions of this paper can be summarized as follows:

1) The input to most of DCNN models requires a fixed-scale image, and the scales of samples are diverse affected by any causes. Hereby, cropping or scaling operation for images is two most common methods of converting images of different scales into fixed-scale images. Whereas the cropping operation will cause the image to lose some key information, making the original image no longer complete. Similarly, the scaling operation compresses the image, causing the original image to produce geometric distortion and reduce the resolution of the image. This will eventually lead to erroneous judgments. In this paper, a novel fingerprint liveness detection based on an improved DCNN with image scale equalization has been proposed to eliminate the restrictions on fixed scale in CNNs without cropping and scaling operations.
2) The most of CNN models use a fixed learning rate to adjust model parameters when learning. In this paper, a method based on a learning rate adaptive adjustment has been proposed to prevent the weight learned from falling into local optimum. Besides, we perform multi-scale equalization operations on high-level features, and thus, our method has scale invariance as well as more robust to image geometric deformation.
3) In this paper, we use the confusion matrix into FLD for the first time as an indicator of FLD. In addition, a number of experiments have been done to evaluate the performance of our method. Such as, fine-tuneing the model parameters based on fingerprint sample sets, testing based on optimal training model, testing based on learning rate adaptive adjustment and testing based on model weight initialization.

The remainder of this paper is organized as follows. In Section II, the Methodology is introduced, including our model structure and the basic theory of ISE. The experimental results and analysis are reported in Section III. Conclusions are finally drawn in Section IV.

II. METHODOLOGY

The main goal of FLD is to eliminate the interference of artificial replicas before identity recognition. Currently, most FLD algorithms are based on texture descriptors, thus, the extracted features play a crucial role in FLD. Deep learning can automatically learn high-level semantic features without experience and professional knowledge of image processing, however, the problem is that the most network models have strict limits on the scale of the image, and the task of this paper is to solve it.

A. CONVOLUTIONAL NEURAL NETWORK WITH LEARNING RATE ADAPTIVE ADJUSTMENT

Different from traditional handcrafted based feature extraction algorithm, each convolutional operation can automatically extract different level features, such as vertical edges, horizontal edges, colors, textures, etc. During each convolutional operation, weights of each neuron connected data window are fixed, and each neuron focuses on only one characteristic. Note that the output of the convolution also needs to go through an excitation function, that is, making the feature do a nonlinear projection. The pooling layer is sandwiched between successive convolutional layers to compress the amount of data and parameters, reducing overfitting. In short, if the input is an image, then the primary role of the pooling layer is to compress the image. The features processed by the pooling layer are invariant, that is, the scale invariance of the features we often mention in image processing. The pooling operation can be regarded as the resize of the image. The information removed during image

compression is only irrelevant. The information, while the information retained has scale invariance and is the most characteristic of the image. At the same time, we all know that the amount of information contained in an image is very large, and there are many features, but some information does not have much use or repetition for our image classification [6]. In CNN, the convolution layer is responsible for extracting features [6], [21], [22], and the pooling layer can be seen as a feature selection operation that removes unimportant features from the feature map extracted by the previous convolution operation. Finally, the features after the n-layer convolution and pooling operations are fed input to the full connection layer, and the full connection layers are responsible for classification. In Figure 5, we visualize a feature map using a convolutional operation and the pooling operation. Convolutional features are represented through computing the inner product of original fingerprint image and filters, and the process of convolution is considered as the process of feature extraction. Next, ReLU is used as the activation function to compute feature maps. After the convolution operation, max-pooling operation is conducted to reduce the dimensionality of feature maps and prevent overfitting. The principle of max-pooling counts the maximum in the sliding windows. Such as the green solid line window in Figure 5, the size of green solid line window is 2×2, and the value of window is calculated as a new value (it is the maximum in the green solid line window) after max-pooling operation. The advantage of the convolutional neural network is that the weights of the convolution kernel are shared, and the processing speed of the high-dimensional data is fast and the precision is high. Meanwhile, it is not necessary to manually select features, and the feature classification effect is good after training the weights. But the drawback is that you must train a model classifier based on big data samples, while adjusting a large number of parameters. Currently, another most important problem is about the learning rate α selection when the gradient derivative updates the model parameters. The learning rate α is a fixed value when weight is updated in most convolutional neural network models, and the calculation of weights update is shown in formula (1). In formula (1), J is an error function about w, and we need to reach position w_{t+1} from current position w_t point. We should be aware that learning rate α shouldn't be set too large or too small. If it is too small, it may lead to delay to the lowest point, as shown in Figure 4.(a). But if it is too big, it will be easy to miss the lowest point, as shown in Figure 4.(b).

$$w_{t+1} = w_t - \alpha * \nabla J(w) \tag{1}$$

Different from the most model structure based on fixed learning rate, we propose an adaptive learning rate algorithm, which can be chosen by using formula (2), In formula (2), α denotes learning rate, and f is learning rate reduction factor. In our method, we will monitor the evaluation metrics in program. If we find the evaluation index no longer promotion, the learning rate and weights will be updated using formula (1) (2). Then, continue to learn the model parameters

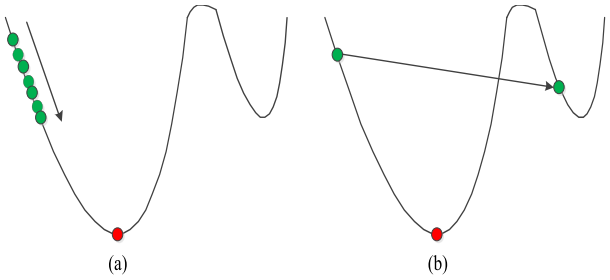


FIGURE 4. Examples of steps to minimize weight at different learning rates. (a) Small learning rate needs lots of setps. (b) Bigger learning rate misses the minimum.

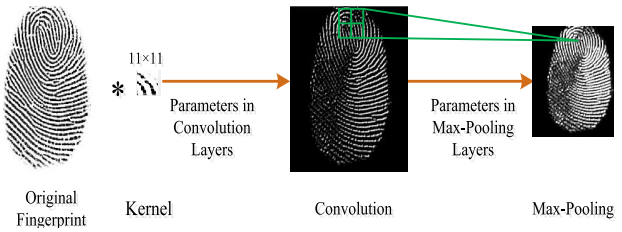


FIGURE 5. The visualization diagram of the single layer convolutional neural network feature extraction.

using the new learning rate α until the adaptive adjustment learning rate is less than the given threshold T. Finally, record the model parameters when the evaluation indexes keep constant during training.

$$\alpha = \begin{cases} (1-f) * \alpha, & f \in [0.2, 1], \quad \alpha \in [min\, lr, 1] \\ \alpha, & \alpha \in [0, min\, lr] \end{cases} \tag{2}$$

B. IMAGE SCALE EQUALIZATION

The input to the CNN model all need a fixed scale image, however, the scales of the captured fingerprint image are diversified because of various fingerprint sensors or human factors. Hereby, two most common operations are necessary to obtain fixed scale input images: one is cropping operation, another is scaling operation. We all know that the cropping operation can preserve the cropped or interest area, and the vast majority of uncut parts will be discarded. As a result, a lot of important texture information will be lost. While scaling operation will reduce the resolution of the image due to scaled image scale. Finally, it will weaken the detection performance of training model. Our goal is to propose a convolutional neural network model structure that is not limited by fingerprint image scale, and a new layer, called image scale equalization (ISE) layer, has been added to our model in this paper. Our proposed ISE layer is next to the last convolutional layer, between the last convolutional layer and the full concatenation layer, as shown in Figure 3. In Figure 3, eight basic layers, which are one input layer, four convolutional layers, one pooling layer, one proposed ISE layer, one full connection layer and one output layer, make up our model frame. Except ISE layer, the specific role of each of the other layers has been detailed in subsection 2.1. Next, we will elaborate in detail the definition and implementation of ISE layer.

As shown in Figure 3, after a series of convolution and pooling operations in the first six layers, we can extract a number of images describing the high-level semantic information of the original image, which we call them feature maps. Following, all features maps of the last convolutional layers are fed into our designed ISE layer. In this paper, the number of feature maps for the last convolutional operation layer is 64. In order to better convert the image of any scale into a fixed-length vector after ISE layer operation, we first need to assume that the size of the feature map of the last convolutional layer output is $a \times a$. If we want to divide the feature maps into n^2 parts, where $n = 2^{i-1}$, $i \in [1, +\infty)$. At the same time, the size of each sub-area window is winSize $= \lceil a/n \rceil$. Since it is not guaranteed that every a/n is an integer, $\lceil \bullet \rceil$ is an up rounding operation. In this paper, n is set to 1, 2, 4, 8, 16, $\cdots$, 2^{i-1}. When n is 1, 2, 4, 8, 16,$\cdots$, 2^{i-1}, the total number of sub-blocks is 1, 5, 21, 85, 341, $\cdots$, $(4^{m+1} - 1)/3 = (4^i - 1)/3$, respectively, where m$\in$ N or i$\in$ N+(or N*). Then, the maximum value of each sub-block is calculated as a feature of the current sub-block. That is, how many sub-blocks an image is divided into, and how many features each image has. In our experiment, n is 1, 2, 4, 8, 16, respectively, and the number of features of the corresponding n is 1, 5, 21, 85, 341, respectively, After ISE operation, all the sub-areas are spliced into one dimensional features vectors, and images (or feature maps) of arbitrary scales are transformed into a vector of a fixed length. Finally, the fixed length vectors are fed into the full connection layer.

C. WEIGHTS INITIALIZATION AND PARAMETERS FINE-TUNING

If the image data set is small, the trained model is prone to over-fitting. That is to say, the precision of the training set is very high. After 5 iterations, the test accuracy is close to 100%. The verification results on the validation set and the test set are very poor. The reason is that the performance of the convolutional neural network model has a great relationship with the amount of data in the image. Thus, we implement three kinds of operations to solve over-fitting problem. The first one is data augmentation operation. Four image expansion techniques are performed in this paper, including image rotation, image scaling, image flip and image brightness enhancement. The second one is weights initialization, that is, in our fingerprint liveness detection model, we will train on the expanded fingerprint set. Suppose that the number of model training iterations is n, the training accuracy is observed by setting the corresponding monitoring program. When the training accuracy is less than 5 times, we think that the model training is completed and the program stops running. There are two possibilities in the model training process: one is that the model may be trained n times to end, and the other is that the model training ends just 5 times of training. The parameters of the training at this time are the training results of our model, and the trained model parameters are used to verify on the test set. The third one is the training based on parameters fine-tuning. First, we will randomly select a batch of fingerprint images in the expanded fingerprint training set and train them as input to the model. After the training is completed, the trained parameters are saved. Next, load the above trained parameters into the model, and then use the training set to fine tune our model. The fine tuning ends well, and the trained model parameters are saved again and used as the final training model parameters.

III. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the detection performance of our proposed DCNNISE, the experiments have been implemented on the fingerprint data sets of LivDet 2011 [1] and LivDet 2013 [29]. Two data sets are respectively from 2011 and 2013 Fingerprint Liveness Detection Competition, and final experimental results are compared with the results of other references in recent years. In this subsection, we first give a brief introduction on the two public data sets, and then we will also introduce the needs of our experimental environment. In addition, performance evaluation criteria will be given in this subsection. Finally, some numerical experiments on two data sets are reported to illustrate the effectiveness of our algorithm, and the experimental results are compared with the state-of-the art methods.

A. LIVDET 2011, 2013 AND OPERATION ENVIRONMENT

Attackers attempt to use artificial replicas of a biometric, a type of presentation attack, to circumvent fingerprint authentication system, so many FLD methods have been proposed so far to recognize a presentation attack and avoid artificial replicas bypassing the authentication system. The fingerprint liveness detection competition promoted by Dr. Gian Luca Marcialis of the Department of Electrical and Electronic Engineering of the University of Cagliari (Italy) [29] and Prof. Stephanie Schuckers of the Department of Electrical and Computer Engineering of Clarkson University (USA) has attracted the attention of many academic and industrial institutions. The goal of Liveness Detection (LivDet) Competition is to compare different FLD algorithms using standardized evaluation protocol and sample set, and the LivDet competitions have been hosted in 2009, 2011, 2013, 2015 and 2017. And after each competition is completed, the competition organizing committee will release live and spoof fingerprint data set. After registration, we can download for free the fingerprint data set for the corresponding year from the competition website via the link ''livdet.org/registration.php''. Since most of the algorithms are based on two data sets, LivDet 2011 and LivDet 2013, hereby, all the experiments in this paper utilize the above two fingerprint data sets. Below we will introduce the above two fingerprint data sets in detail.

The LivDet 2011 data set, releasing in LivDet 2011 competition, collected 16056 samples of both live and spoof fingerprints using four different flat optical sensors, and a detailed description of the LivDet 2011 data set has been reported in Table 1, including the image size and resolution, the number of different fingerprint types, etc. In Table 1,

TABLE 1. The samples distribution of the LivDet2011 and LivDet2013 data sets.

Database ID	Sensor	Abbreviation	Res.(dpi)	Image Size	Training Samples Dataset		Testing Samples Dataset	
					Real	Fake	Real	Fake
Liv2013-1	Biometrika	Bio	569	352 × 384	1000	1000	1000	1000
Liv2013-2	CrossMatch	Cro	500	800 × 750	1250	1000	1250	1000
Liv2013-3	Italdata	Ita	500	480 × 640	1000	1000	1000	1000
Liv2013-4	Swipe	Swi	96	1500 × 208	1221	979	1153	1000
Liv2011-1	Biometrika	Bio	315	372 × 208	1000	1000	1000	1000
Liv2011-2	Digital	Dig	355	391 × 208	1004	1000	1000	1000
Liv2011-3	Italdata	Ita	640	480 × 208	1000	1000	1000	1000
Liv2011-4	Sagem	Sag	352	384 × 208	1008	1008	1000	1036

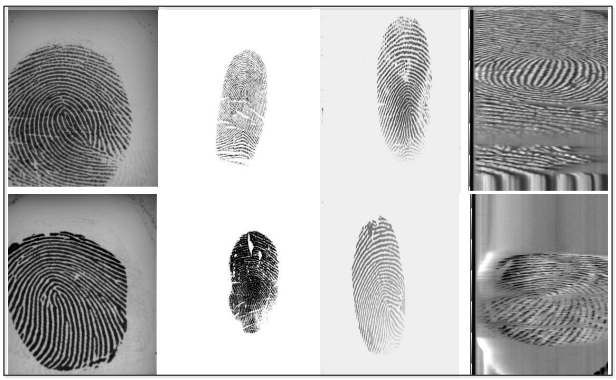


FIGURE 6. Llive (above) and spoof (below) fingerprints captured via 4 different sensors. (a) Biometrika. (b) Crossmatch. (c) Italdata. (d) Swipe.

each fingerprint sensor contains two types of fingerprint data sets: Training set with 8020 fingerprint samples and Testing set with 8036 samples. Besides, these samples are collected via four different optical sensors, for example, Biometrika, Italdata, Digital and Sagem. Whether it is Training data set or Testing data set, they are all divided into two parts: Live samples and spoof samples, and we should know that they have no overlap with each other. The live samples are captured via above four different optical sensors, and the spoof artificial replicas are generated using some common fingerprint materials under the cooperation of testers.

The LivDet 2013 [29] data set, releasing in LivDet 2013 competition, consists of 16853 live and spoof fingerprints also captured by means of four different flat optical sensors, which are Biometrika, CrossMatch, Italdata and Swipe, respectively. Figure 6 lists some fingerprint samples from four different optical sensors. Note that it is difficult to observe the slight difference between the real fingerprints and the fake ones by the naked eyes. Similar to the description in LivDet 2011, two types of fingerprint samples datasets are included: Training data set with 8450 samples and Testing data set with 8403 samples. The goal of Training data set is used to obtain a classifier model, and the performance evaluation of the model is verified by using Testing samples. The detailed distribution of the LivDet 2013 has also been reported in Table 1. In addition, we can observe that the ratio of live or spoof samples in each data set is 1:1 approximately. The scales of the given samples are diverse from 315 × 372 to 1500 × 208. Generally speaking, two most common methods, the cropping and scaling operations, will be operated when encountering images of different scales. However, the above two operations may result in loss of image texture information or reduce the resolution of the original image, which ultimately affects the performance of the algorithm. Therefore, a novel network model framework using an improved Deep Convolutional Neural Network with Image Scale Equalization (DCNNISE) has been proposed in this paper to protect texture information and maintain image resolution invariant.

B. PERFORMANCE EVALUATION AND EXPERIMENTAL ENVIRONMENT

Average Classification Error (*ACE*), which is a standard performance indicator [8], [30], is used to evaluate detection performance, and the calculation formula of evaluation metrics of *ACE* is defined in Equation (3):

$$ACE = \frac{FAR + FRR}{2}, \tag{3}$$

where in equation (1), *FAR* (False Accept Rate) is the percentage of misclassified live samples and *FRR* (False Reject Rate) denotes the percentage of misclassified as spoof samples. For *FAR* and *FRR*, two calculation equations are defined separately as:

$$FAR = \frac{misclassified\ real\ images}{total\ real\ images} * 100, \tag{4}$$

$$FRR = \frac{misclassified\ fake\ images}{total\ fake\ images} * 100, \tag{5}$$

The *ACE* is able to be represented by any number between 0 and 100. The smaller the *ACE*, the better the performance of the proposed method. All experiments are operated using python 3.5.2 programming on a single GeForce GTX 1080 GPU (64G memory) with two weeks. About the operational environment, two conditions are vital hardware

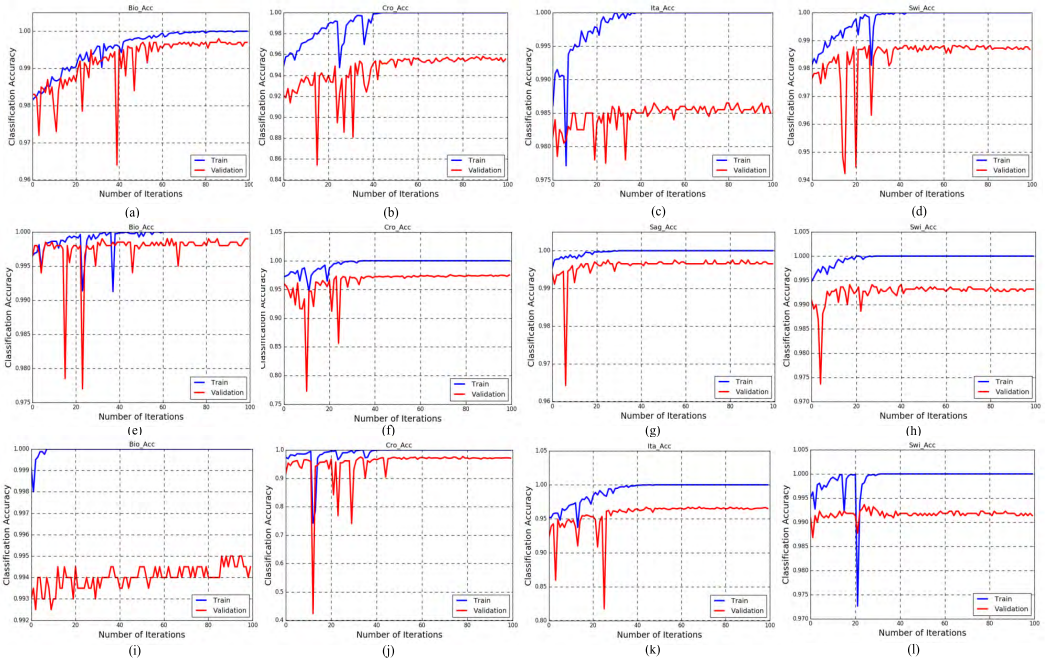


FIGURE 7. Training and validation sets precision change trend with the number of iterations in LivDet 2013 datasets.

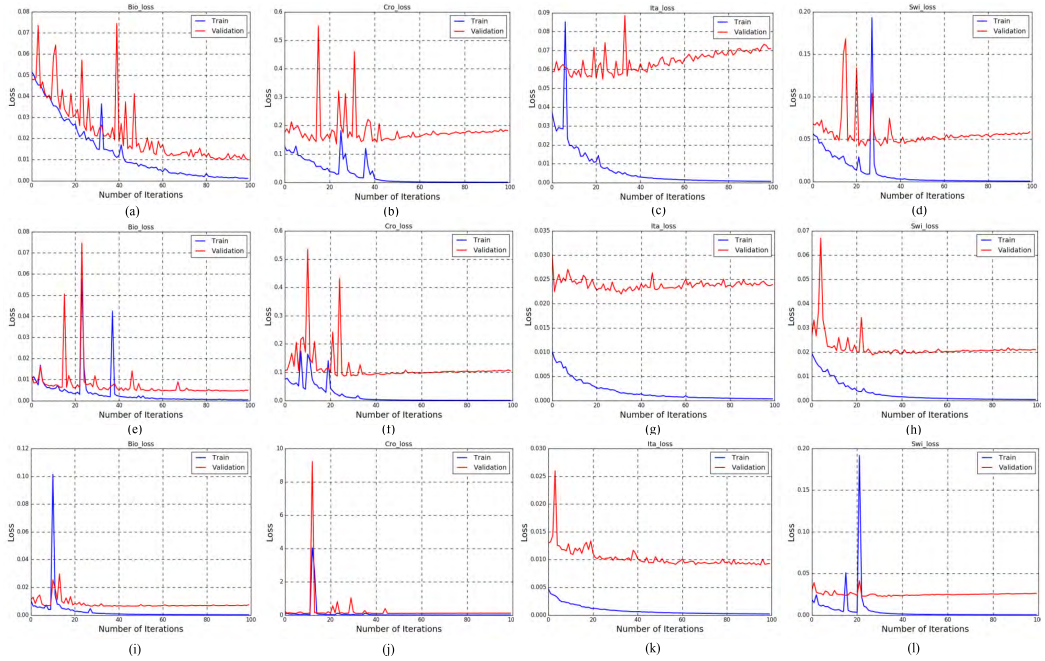


FIGURE 8. Training and validation sets loss change trend with the number of iterations in LivDet 2013 datasets.

condition and software condition. The detailed description about our experimental environment has been reported in Table 3.

In addition to using generic *ACE* to evaluate the performance of FLD, the paper for the first time applies the confusion matrix to FLD as a performance indicator. We performed performance verification on model fine-tune, and four parameters, *FRR* (False Reject Rate, which shows the percentage of live samples that were incorrectly marked as spoof samples), *FPR* (False Positive Rate, which shows the percentage of spoof samples that were incorrectly marked as live samples), *Recall* (which shows the probability that live samples are predicted correctly) and $F1$-Score (which is a weighted average of the precision and recall rates in the binary model), are calculated as performance evaluation indicators. The confusion matrix, also called the error matrix,

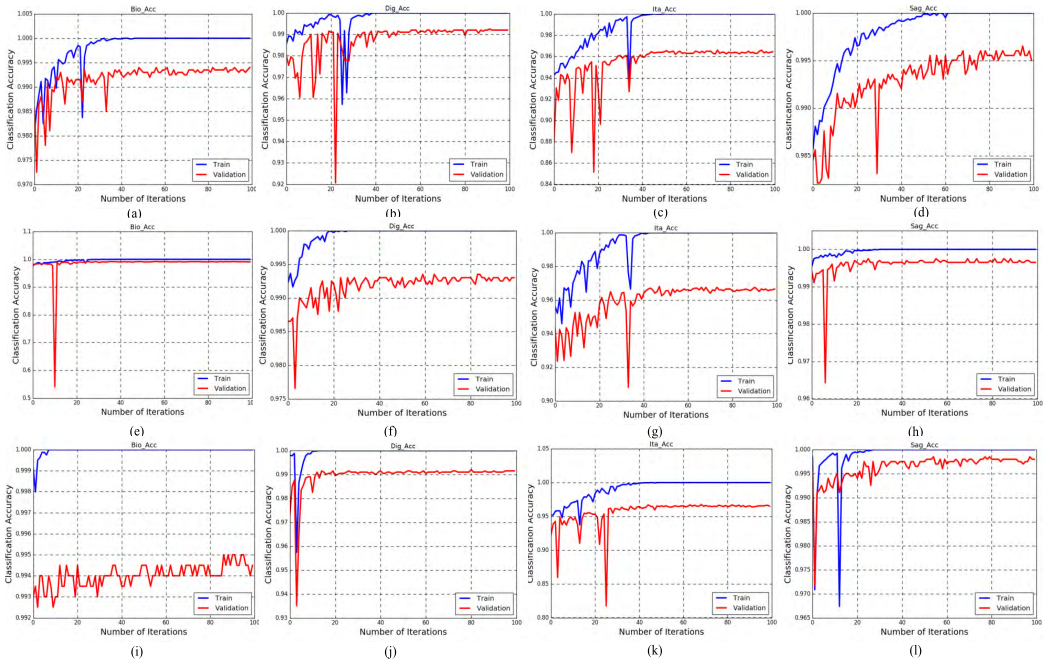


FIGURE 9. Training and validation sets precision change trend with the number of iterations in LivDet 2011 datasets.

TABLE 2. Confusion matrix of live or spoof fingerprints.

Confusion Matrix		Predicted Class		Total
		0	1	
Actual Class	0	A	B	E=A+B
	1	C	D	F=C+D
Total		G=A+C	H=B+D	

TABLE 3. Computer software and hardware configuration.

Hardware Condition	Software Condition
CPU: Intel@ Core i7-6700	Operating System: Ubuntu
Memory: 64G	(Version 16.04)
GPU: NVIDIA@ GeForce-	Run enviroment: Python 3.5.2
GTX 1080 8GB	+ Cuda 8.0

is a standard format for the accuracy evaluation. The confusion matrix of the two-category image used in this paper is shown in Table 2. In Table 2, 0 denotes the number of live samples categories, 1 indicates the number of spoof samples categories, and A denotes the TP (True Positive, which shows that live samples that were correctly classified as live samples), B denotes the FN (False Negative, which shows that live samples that were incorrectly marked as spoof samples), C denotes the FP (False Positive, which shows that spoof samples that were incorrectly marked as live samples), D denotes the TN (True Negative, which shows that spoof samples that were correctly classified as spoof samples), E denotes the number of actual live samples, F denotes the number of actual spoof samples, G denotes the number of predicted live samples, and H denotes the number of predicted spoof samples. For the four paramaters values, the calculation formulas of *FRR*, *FPR*, *Recall* and $F1$-Score are FRR =B/(A+B), FPR = C/(C+D), $Reacll$ = A/(A+B) and $F1$-Score = 2A/(2A+B+C), respectively.

C. EXPERIMENTAL PROCESS AND RESULTS

In this paper, several different types of experiments have been implemented to prevent presentation attack bypassing the authentication systems. Firstly, in image preprocessing stage, region of interest of each fingerprint sample has been extracted to eliminate the interference in blank areas. Then, to solve the problem of insufficient fingerprint samples, we use four different image processing techniques, including image rotation, image scaling, image flip and image brightness enhancement, to extend the given fingerprint data set. After all image preprocessing operations, fingerprint samples are imported into our built model. Since our fingerprint image set is limited, in our experiment, and the number of epochs is 100, batch size is set to 32, the number of batch is 313, and Iteration per epoch is set to 313. The number of convolution kernels of the first convolutional layer and the second convolutional layer is 32, and the number of convolution kernels of the third convolutional layer to the fifth convolutional layer is 64, and the sizes of the convolution kernel of the five layers is 3×3. Additionally, the training samples of each batch need to be scrambled before entering the convolutional layer.

In our experiments, we used three different methods to conduct experiments separately. First, we directly input the processed training and testing sample set into the built model. This experiment does not perform any model parameter initialization. Because our algorithm is a supervised

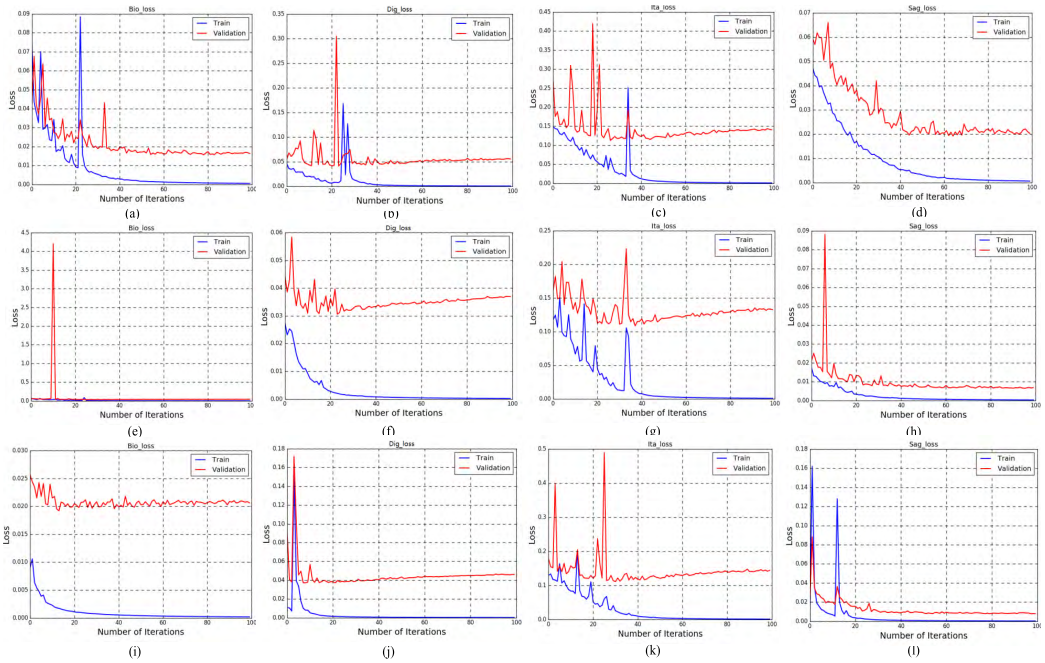


FIGURE 10. Training and validation sets loss change trend with the number of iterationsl in LivDet 2011 datasets.

learning process, each sample and its corresponding label are required. The labeling of the label is done, determined by us according to the liveness of each fingerprint sample, by us in the image preprocessing stage, that is, the label corresponding to the live fingerprint is 1, and the label corresponding to the spoof sample is -1. During the training phase of our model, convolution or pooling operations are first performed layer-wise from the input to output. Unlike most CNN models, the last convolutional layer is followed by a full connection operation. However, its shortcoming is that there are strict restrictions on the input scale of images, and this is also one of the problems that our paper has to solve. As shown in Figure 3, a new scale equalization layer is added to the end of the last convolutional layer. That is, the output of the last convolutional layer will be the input to our scale equalization layer. After our method proposed in this paper, images of any scale will be transformed into fixed-length feature vectors. Below we will give a concrete example to illustrate the process of scale equalization operation. Next, the output of scale equalization layer will be used as the input to the next layer of full connection layer. It should be emphasized that we only utilize a full connection layer in this paper. Finally, the output of scale equalization layer is used as the input of softmax layer, and the final output, called the predicted value, will be compared with the label of the image, called the actual value. If the difference between them is less than a threshold value, the model parameter of the learning is the learned parameter, and then the next operation is performed. If the difference between the predicted value and the actual value is larger than the given threshold, the gradient is derived and the weight obtained from the training is fine-tuned. Since the learning rate is a certain value in the process of deriving the error gradient of the predicted value and the actual value, the learned weights are easily caught in the local optimum or difficult to converge to the optimal value. Based on this, this paper improves on the basis of the traditional CNN model and proposes a CNN based on adaptive adjustment of learning rate. The implementation of this operation has been described in detail in Section 2.2, and the framework of the process implementation is shown in Figure 3. When the gradient is derived to the first convolutional layer, the pre-propagation is performed again with the fine-tuned weights. And the error between the predicted value and the actual value is calculated again, at this time, if the minimum error is smaller than the given threshold, the first learning process is terminated. Otherwise repeat the previous process again. According to the above operation, all the images in the training sample set are operated once and the final training model parameters are obtained. In our experiments, the total number of trainable parameters is 102564. Wherein, the number of parameters of the first convolutional layer is 320, the number of the second convolutional layer is 9248, the number of parameters of the third convolutional layer is 18496, and the number of parameters of the fourth convolutional layer is 36928, and the number of parameters for the five convolutional layers is 36928. Since the scale normalization layer does not involve convolution operations, the learning parameter of this layer is zero, and the number of parameters of the last fully connected layer is 642. In addition, the number of classification is 2, which is either a live fingerprint or a spoof fingerprint.

The second experiment is about maximizing the training weight model. The training process is exactly the same as the first experiment. The difference is that we will set a threshold during the running process, that is, the training accuracy of

TABLE 4. The ACEs of different methods in LivDet 2013.

Methods	The Average Classification Error ACE in (%)				
	Bio	Cro	Ita	Swi	Ave.
Our method	4.35	**7**	**1.4**	**2.05**	**3.7**
HOG [33]	**2.75**	7	7.05	4.27	5.27
ULBP [16]	10.68	46.09	13.7	14.35	21.205
Anonym3 [29]	5.7	53.11	2.8	5.25	16.72
LBP PCA [22]	1.7	49.45	2.3	3.34	14.2
MSDCM [9]	3.55	20.84	2.35	5.25	7.59
LLF [30]	3.9	28.8	1.7	14.4	12.2
HIGMC [30]	4.3	39.96	10.6	32.41	21.82
HIDBP [30]	3.9	34.13	8.3	14.44	15.19
Winner [29]	4.7	31.2	3.5	14.07	13.37

TABLE 5. The ACEs of different methods in LivDet 2011.

Methods	The Average Classification Error ACE in (%)				
	Bio	Dig	Ita	Sag	Ave.
Our method	9.2	**1.35**	12.35	**2.9**	**6.45**
LLF [27]	7.89	6.25	8.1	5.36	6.9
MSLBP1 [8]	7.3	2.5	14.8	5.3	7.475
MSLBP2 [8]	10.6	6.7	12.6	5.6	8.875
GBTF [15]	**6.7**	4.75	11.75	3.34	6.635
LBP+PCA [22]	8.85	4.15	12.34	7.54	8.22
SURFPHO [27]	8.76	6.9	**7.4**	6.23	7.32
Best Res. [1]	20	36.1	21.8	13.8	22.925

the trained model will not increase 5 times, then the training is terminated, and the training parameters are learned at this time. The weights of the optimal model are preserved. Next, the performance detection of the model is been performed based on the optimal training model. The third experiment is the fine tuning of the model parameters. That is, at the beginning, any part of the image is selected from the training set and trained. This process is the same as in the first experiment until the selected image is trained, until the conditions similar to first experiment are met, we can learn a model parameter. The trained model will be used as the pre-trained model for our next stage. We then fine-tune the training set using LivDet 2011 and LivDet 2013 fingerprint samples to get the final training model, and the pre-training process will end.

For the sake of simplicity, convolutional, pooling, scale equalization and full connection layers are abbreviated as conv, pl, SE and fc, respectively. For instance, conv2 indicates that the second layer is a convolution layer; Mp3 indicates that the third layer is a max pooling layer; SE6 indicates that the sixth layer is a scale equalization layer; fc7 indicates that the seventh layer is a full connection layer. Supposed that the scale of original input fingerprint sample is 200×200, and the size and sliding step of the convolutional kernel of the conv1 layer are 3 and 1, respectively. Thus, the scale of feature map of the conv1 layer is $((200-3)/1+1)\times((200-3)/1+1)=198\times198$. In this paper, the pooling operation uses the maximum value in the given window as the output value of the step operation, and the size of the pooling window is 2 and stride is 1. After pooling operation, the scale of the Mp3 layer output is 98×98. Supposed that (86-3)/2 indivisible, then the edge of the window will be filled with 0. The parameters of conv2 layer are the size of convolutional kernel 3 and sliding step 1, so the size of conv2 output image is 196×196. According to the above calculation process, the sizes of conv4 and conv5 are 96×96 and 94×94, and the total number of feature maps is 64. Next, all the feature maps are fed into SE6 layer, and we have already talked about the definition and calculation process of ISE in part 2. In this example, we divide each feature map into n^2 blocks, where n is set to 16 and the total number of sub-areas is 341. After convolution and pooling operation, the scale of output of the last convolutional layer is 94×94,

TABLE 6. Average classification accuracy under different image scales and different block combinations in LivDet 2013.

		No Operation			Weights Init. Operation			Fine-tune Operation		
		64 x 64	112 x 112	224 x 224	64 x 64	112 x 112	224 x 224	64 x 64	112 x 112	224 x 224
Biometrika	1, 2, 4	93.9	93.9	95.45	94.4	94.25	93.8	90.6	90	95.15
	1, 2, 4, 8	86.45	95.65	92.9	90.4	95.45	94.3	89.5	94.4	94.3
	1, 2, 4, 8, 16	89.45	92.1	94.7	90.15	92.5	94.85	88.4	93.3	94.6
CrossMatch	1, 2, 4	88.98	91.07	90.71	90.13	90.58	90.62	88.93	90.98	90.93
	1, 2, 4, 8	89.47	89.73	90.8	89.3	93	90.62	89.47	89.73	90.8
	1, 2, 4, 8, 16	89.11	90.49	90.53	89.02	90.62	92.31	89.42	89.78	90.44
Italdata	1, 2, 4	96.05	98.15	98	94.3	97	98.6	95.75	96.85	97.6
	1, 2, 4, 8	94.55	96	97.05	95.6	98	97.15	89.2	97.35	90.89
	1, 2, 4, 8, 16	92.9	97.3	96.95	93.65	95.05	98.1	92.64	95.9	91.28
Swipe	1, 2, 4	96.47	97.07	96.66	96.56	96.57	97.1	94.7	95.82	96.93
	1, 2, 4, 8	96.05	96.75	96.52	96.84	97.68	96.84	95.91	97.95	96.6
	1, 2, 4, 8, 16	96.7	97.03	97.82	96.19	97.9	97.86	95.4	97.03	97.53

TABLE 7. Average classification accuracy under different image scales and different block combinations in LivDet 2011.

		No Operation			Weights Init. Operation			Fine-tune Operation		
		64 x 64	112 x 112	224 x 224	64 x 64	112 x 112	224 x 224	64 x 64	112 x 112	224 x 224
Biometrika	1, 2, 4	86.15	88.65	89.35	85.7	90.1	90.05	88.09	90.49	90.8
	1, 2, 4, 8	86.2	89.7	86.9	89.85	88.75	88.9	90.6	90.27	89.42
	1, 2, 4, 8, 16	84.65	85.55	87.4	85.55	86.9	88.54	89.51	89.69	90.22
Digital	1, 2, 4	94.8	98.45	96.3	96.89	98.45	97.05	96.15	98	96.35
	1, 2, 4, 8	95.95	94.85	96.2	95.35	96	96.6	95.1	95.6	96.75
	1, 2, 4, 8, 16	94	95.7	95.5	95.05	97.5	96.1	94.05	95.55	94.85
Italdata	1, 2, 4	79.75	78.1	78.25	82.05	83.40	87.1	80.14	83.15	86.95
	1, 2, 4, 8	79.35	85.5	81.8	80.4	87.65	83.2	78.4	86.75	84
	1, 2, 4, 8, 16	79.15	82.7	86.79	78.8	84.1	86.9	79.85	84.25	86
Sagem	1, 2, 4	95.38	96.17	96.22	95.33	96.76	97.1	92.98	95.92	95.78
	1, 2, 4, 8	93.76	94.79	93.76	94.40	96.12	95.97	92.78	94.79	94.45
	1, 2, 4, 8, 16	90.52	94.94	95.53	94.45	94.84	95.28	94.3	93.47	95.43

and the scale of each sub- block is 5×5. The maximum value of each block is then calculated as a feature of the sub-block. Finally, we concatenate these features of all the sub-blocks and use them as the final feature of the image. That is, the final output is a fixed-length feature vector, whose size is $Km = 96 \times 341$, where $m = 341$ denotes the number of features and $K = 96$ is the number of the features maps of in the last convolutional layers. The fixed length vectors are fed into the input of the full connection layer, and we can obtain the final predicted value.

To evaluate the detection performance of our proposed algorithm, our experimental results are compared with several state-of-the-art approaches, including ULBP [16], HOG [33], MSDCM [9], Winner LCP [29], LLF and HIGMC [30]. Tables 4 and 5 respectively list the newly proposed algorithms on the two data sets LivDet 2013 and LivDet 2011, and we can observe that the *ACE*s of our method are 1.57 and 0.185 lower than the second place respectively, which are highlighted in bold. And in the single performance evaluation, our algorithm has higher detection accuracy in Swipe, Italdata-2013, Sagem and Digital. The reason why the other single items are irrational is that: 1. The image training set is not large enough; 2. The quality of the image itself is relatively poor.

Figure 7 and Figure 9 denote the precision change trend of training set and validating set with the number of iterations in LivDet 2013 and LivDet 2011 two datasets, and in each Figure, the first row indicates the trend of classification accuracy as different iterations increase with different data acquisition equipment under the same image scale operation. The first column shows the trend of classification accuracy as the number of iterations increases under different scale operations under the same acquisition equipment. Generally speaking, we find that the overall classification accuracy will increase until it tends to level as the number of iterations increases, and all of them can achieve satisfactory detection accuracies. By contrast, Figure 8 and Figure 10 denote the loss change trend of training set and validating set with the number of iterations in LivDet 2013 and LivDet 2011 two datasets, and we can observe that the overall loss will decrease until it tends to level as the number of iterations increases, which also matches the reality.

TABLE 8. Performance evaluation of fine-tuning operation under different scale combinations when the number of sub-blocks is 1, 4, 16 in LivDet 2011.

		FRR	FPR	F1-Score	Reacll
Biometrika	64x64	12.9	14.5	86	86
	112x112	10.4	13	88	88
	224x224	8.3	13.5	89	89
Digital	64x64	7.2	1.9	95	95
	112x112	1.1	2.5	98	98
	224x224	3.3	4.7	96	96
Italdata	64x64	13	31.2	78	78
	112x112	11.4	21.1	84	84
	224x224	13.1	28.3	79	79
Sagem	64x64	4.7	6.1	95	95
	112x112	2.8	5.1	96	96
	224x224	5	4.6	95	95

All the experimental results, based on three different experiments, No operation, Weights Init. and Fine-tune, are reported in Table 6 and Table 7. Moreover, we have also shown the results of newly proposed performance evaluation in Table 8 and Table 9, and we can see that, basically the detection performance is the best when the scale of samples is 112×112. By observing Table 10, we found that the testing time of a sample on two different fingerprint data sets is acceptable. After learning these model parameters, classification performance based on testing samples is performed. And the testing time of a fingerprint image is satisfactory in practical application.

TABLE 9. Performance evaluation of fine-tuning operation under different scale combinations when the number of sub-blocks is 1, 4, 16 in LivDet 2013.

		FRR	FPR	F1-Score	Reacll
Biometrika	64x64	7.3	7.4	93	93
	112x112	6.4	5.2	94	94
	224x224	7.4	5	94	94
CrossMatch	64x64	1.9	17.5	87	87
	112x112	6.6	16.5	90	90
	224x224	3.3	4.7	96	96
Italdata	64x64	2	4.2	97	97
	112x112	1.2	3.4	98	98
	224x224	1	4.8	97	97
Swipe	64x64	71	6.1	94	94
	112x112	2.9	4.3	97	97
	224x224	3.5	3.2	97	97

TABLE 10. Total training and testing time under three different scale pre-training.

database	Training all Fingerprints (three different scales)	Testing a Fingerprint (three different scales)
LivDet2011	4.78-5.78 hours	2.22ms
LivDet2013	4.93-5.53 hours	2.33ms

IV. CONCLUSIONS

In 2012, it made a huge breakthrough in solving the ImageNet challenge and was widely regarded as the beginning of the deep learning (DL) revolution in 2010. Since then, more and more scholars and research institutes turned to DL techniques to solve problems related to computer vision, pattern recognition, etc. However, most previous CNNs models are constrained by the size of input images, that is to say, they need the fixed dimension of input (e.g., the VGG16 takes input with the dimension of $224\times224\times3$). We can process the problem of image scale via two common operations: cropping and scaling, but these two operations also yield some new problems. For instance, the former one can easily crop out important texture information that distinguish the live and spoof fingerprints, so that the learned feature descriptors do not have the ability to distinguish between true and false fingerprints; the latter forcibly changes the scale of the image, thereby reducing the resolution of the image. Both of them all affect the final the detection performance. In this paper, we propose a new FLD based on an improved CNNISE to eliminate the limitation of the image scale. On one hand, our method solves the problem of image scale, and images of any scale can be used as input to our model. On the other hand, an adaptive learning rate method has been added to the DCNNISE, and it prevents the weights from falling into a local minimum and makes the weights converge to global optimum during the gradient back derivation. Through many comparative experiments, the results show that our method is superior to other methods and is suitable for FLD to prevent presentation attack. In addition, we applied the confusion matrix to FLD for the first time, and the results in Table 8 and Table 9 show that our algorithm can achieve high accuracy on the whole. From the perspective of testing time in Table 10, 2.2 milliseconds or so is very ideal in practical application, which basically completes the identification of live and spoof fingerprints without being noticed by people. However, the main problem we face is still the shortage of fingerprint image set, and how to expand and produce a high resolution fingerprint set is the first problem we need to consider.

As we all know, in DL, to learn a better model classifier, we need as many training samples as possible. In the case where the data set is limited and cannot be collected, the conventional operation is to increase the amount of data of the training samples by image processing techniques. Inspired by this, how can I automatically learn more training sample sets based on the characteristics of fingerprint samples without image processing technology? The current popular generative adversarial network (GAN) is to generate a new image through continuous generation and judgment. Therefore, how to use the GAN [31] to produce high-resolution and extended live and spoof fingerprint training sets is the focus of our next study. In addition, the convolutional neural network can extract high-level semantic features. Theoretically, the features of the last layer of learning are the best, but whether it is absolute or not, we need to conduct further study. All these works will be done in our next phase of this research.

ACKNOWLEDGMENT

The authors are grateful for the anonymous reviewers who made constructive comments and improvements.

REFERENCES

[1] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli, and S. Schuckers, "LivDet 2011—Fingerprint liveness detection competition 2011," in *Proc. 5th IAPR Int. Conf. Biometrics (ICB)*, 2012, pp. 208–215.

[2] C. Yuan, X. Sun, and R. Lv, "Fingerprint liveness detection based on multi-scale LPQ and PCA," *China Commun.*, vol. 13, no. 7, pp. 60–65, Jul. 2016.

[3] C. Yuan, X. Sun, and Q. M. J. Wu, "Difference co-occurrence matrix using BP neural network for fingerprint liveness detection," *Soft Computing*, 2018, pp. 1–13. doi: 10.1007/s00500-018-3182-1.

[4] P. V. Reddy, A. Kumar, S. M. K. Rahman, and T. S. Mundra, "A new antispoofing approach for biometric devices," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 4, pp. 328–337, Dec. 2008.

[5] Y. Zhang, J. Tian, X. Chen, X. Yang, and P. Shi, "Fake finger detection based on thin-plate spline distortion model," in *Proc. Int. Conf. Adv. Biometrics*. Springer-Verlag, 2007, pp. 742–749.

[6] C. Yuan, X. Li, Q. Wu, J. Li, and X. Sun, "Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis," *Comput., Mater. Continua*, vol. 53, no. 4, pp. 357–372, 2017.

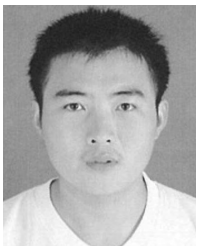
[7] L. Ghiani, A. Hadid, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection using binarized statistical image features," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst.*, Sep./Oct. 2013, pp. 1–6.

[8] X. Jia *et al.*, "Multi-scale local binary pattern with filters for spoof fingerprint detection," *Inf. Sci.*, vol. 268, pp. 91–102, Jun. 2014.

[9] C. Yuan, Z. Xia, X. Sun, D. Sun, and R. Lv, "Fingerprint liveness detection using multiscale difference co-occurrence matrix," *Opt. Eng.*, vol. 55, no. 6, pp. 063111-1–063111-10, 2016.

[10] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

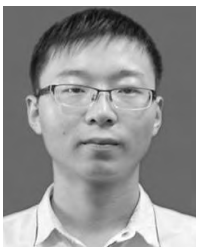
[11] W. Lixia and J. Dalin, "A method of parking space detection based on image segmentation and LBP," in *Proc. IEEE 4th Int. Conf. Multimedia Inf. Netw. Secur.*, Nov. 2012, pp. 229–232.
[12] O. Çeliktutan, B. Sankur, and I. Avcibaş, "Blind identification of source cell-phone model," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 553–566, Sep. 2008.
[13] S. B. Nikam and S. Agarwal, "Texture and wavelet-based spoof fingerprint detection for fingerprint biometric systems," in *Proc. 1st Int. Conf. Emerg. Trends Eng. Technol.*, Jul. 2008, pp. 675–680.
[14] E. Marasco and C. Sansone, "Combining perspiration- and morphology-based static features for fingerprint liveness detection," *Pattern Recognit. Lett.*, vol. 33, no. 9, pp. 1148–1156, 2012.
[15] Z. Xia, R. Lv, Y. Zhu, P. Ji, H. Sun, and Y.-Q. Shi, "Fingerprint liveness detection using gradient-based texture features," *Signal Image Video Process.*, vol. 11, no. 2, pp. 381–388, 2016.
[16] Y. Jiang and X. Liu, "Uniform local binary pattern for fingerprint liveness detection in the Gaussian pyramid," *J. Elect. Comput. Eng.*, vol. 2018, Jan. 2018, Art. no. 1539298.
[17] J. Jia, L. Cai, K. Zhang, and D. Chen, "A new approach to fake finger detection based on skin elasticity analysis," in *Advances in Biometrics*. Berlin, Germany: Springer, 2007, pp. 309–318.
[18] J. Sun, Z. Shisong, and W. Xiaosheng, "Image retrieval based on an improved CS-LBP descriptor," in *Proc. 2nd IEEE Int. Conf. Inf. Manage. Eng.*, 2010, pp. 115–117.
[19] Y. S. Moon, J. S. Chen, K. C. Chan, K. So, and K. C. Woo, "Wavelet based fingerprint liveness detection," *Electron. Lett.*, vol. 41, no. 20, pp. 1112–1113, Sep. 2005.
[20] J. Galbally, S. Marcel, and J. Fierrez, "Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 710–724, Feb. 2014.
[21] Q. Cui, S. McIntosh, and H. Sun, "Identifying materials of photographic images and photorealistic computer generated graphics based on deep CNNs," *Comput., Mater. Continua*, vol. 55, no. 2, pp. 229–241, 2018.
[22] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, "Evaluating software-based fingerprint liveness detection using convolutional networks and local binary patterns," in *Proc. IEEE Workshop Biometric Meas. Syst. Secur. Med. Appl. (BIOMS)*, Oct. 2015, pp. 22–29.
[23] G. L. Marcialis, F. Roli, and A. Tidu, "Analysis of fingerprint pores for vitality detection," in *Proc. 20th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2010, pp. 1289–1292.
[24] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "Fingerprint liveness detection based on Weber local image descriptor," in *Proc. IEEE Workshop Biometric Meas. Syst. Secur. Med. Appl. (BIOMS)*, Sep. 2013, pp. 46–50.
[25] D. Gragnaniello, G. Poggi, C. Sansone, and L. Verdoliva, "Local contrast phase descriptor for fingerprint liveness detection," *Pattern Recognit.*, vol. 48, no. 4, pp. C1050–C1058, 2015.
[26] L. Ghiani, G. L. Marcialis, and F. Roli, "Fingerprint liveness detection by local phase quantization," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. C537–C540.
[27] R. K. Dubey, J. Goh, and V. L. L. Thing, "Fingerprint liveness detection from single image using low-level features and shape analysis," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 7, pp. 1461–1475, Jul. 2016.
[28] Z. Xia, C. Yuan, R. Lv, X. Sun, N. N. Xiong, and Y.-Q. Shi, "A novel Weber local binary descriptor for fingerprint liveness detection," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published. doi: 10.1109/TSMC.2018.2874281.
[29] L. Ghiani *et al.*, "LivDet 2013 fingerprint liveness detection competition 2013," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–6.
[30] C. Gottschlich, E. Marasco, A. Y. Yang, and B. Cukic, "Fingerprint liveness detection based on histograms of invariant gradients,"in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Sep./Oct. 2014, pp. 1–7.
[31] D. Zeng, Y. Dai, F. Li, R. S. Sherratt, and J. Wang, "Adversarial learning for distant supervised relation extraction," *Comput., Mater. Continua*, vol. 55, no. 1, pp. 121–136, 2018.
[32] C. Yuan and X. Sun, "Fingerprint liveness detection adapted to different fingerprint sensors based on multiscale wavelet transform and rotation-Invarient local binary Pattern," *J. Internet Technol.*, vol. 19, no. 1, pp. 91–98, 2018.
[33] C. Yuan and X. Sun, "Fingerprint liveness detection using histogram of oriented gradient based texture feature," *J. Internet Technol.*, vol. 19, no. 5, pp. 1499–1507, 2018.



CHENGSHENG YUAN received the B.E. degree in software engineering from the Nanjing University of Information Science and Technology, China, in 2014, where he is currently pursuing the Ph.D. degree. He is also a Visiting Student with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. His research interests include biometric features' recognition, digital forensic, and machine learning.

ZHIHUA XIA received the B.E. degree from Hunan City University, China, in 2006, and the Ph.D. degree in computer science and technology from Hunan University, China, in 2011. He is currently an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include cloud computing security and digital forensics.

LEQI JIANG received the M.S. degree in software engineering from Nanchang Hangkong University, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include encrypted image processing and data security in cloud.



YI CAO received the B.S. degree from the Nanjing University of Information Science and Technology, China, in 2016, where he is currently pursuing the Ph.D. degree. His research interest includes network and information security.

Q. M. JONATHAN WU (M'92–SM'09) received the Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990. He was with the National Research Council of Canada for ten years, from 1995 to 2004, where he became a Senior Research Officer and a Group Leader. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. His current research interests include 3-D computer vision, active video object tracking and extraction, interactive multimedia, sensor analysis and fusion, and visual sensor networks.

XINGMING SUN received the B.S. degree in mathematics from Hunan Normal University, China, in 1984, the M.S. degree in computing science from the Dalian University of Science and Technology, China, in 1988, and the Ph.D. degree in computing science from Fudan University, China, in 2001. He is currently a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include network and information security, and digital watermarking.