

A Lightweight Convolutional Neural Network with Representation Self-challenge for Fingerprint Liveness Detection

Jie Chen¹, Chengsheng Yuan^{1,2,*}, Chen Cui², Zhihua Xia¹, Xingming Sun^{1,3} and Thangarajah Akilan⁴

¹School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, 210044, China

²Key Laboratory of Public Security Information Application Based on Big-Data Architecture, Ministry of Public Security, Zhejiang Police College, Hangzhou, 310053, China

³Jiangsu Yuchi Blockchain Research Institute, Nanjing, 210044, China

⁴Department of Software Engineering, Lakehead University, Thunder Bay, ON P7B 5E1, Canada

*Corresponding Author: Chengsheng Yuan. Email: yuancs@nuist.edu.cn

Received: 30 January 2022; Accepted: 30 March 2022

Abstract: Fingerprint identification systems have been widely deployed in many occasions of our daily life. However, together with many advantages, they are still vulnerable to the presentation attack (PA) by some counterfeit fingerprints. To address challenges from PA, fingerprint liveness detection (FLD) technology has been proposed and gradually attracted people's attention. The vast majority of the FLD methods directly employ convolutional neural network (CNN), and rarely pay attention to the problem of over-parameterization and over-fitting of models, resulting in large calculation force of model deployment and poor model generalization. Aiming at filling this gap, this paper designs a lightweight multi-scale convolutional neural network method, and further proposes a novel hybrid spatial pyramid pooling block to extract abundant features, so that the number of model parameters is greatly reduced, and support multi-scale true/fake fingerprint detection. Next, the representation self-challenge (RSC) method is used to train the model, and the attention mechanism is also adopted for optimization during execution, which alleviates the problem of model over-fitting and enhances generalization of detection model. Finally, experimental results on two publicly benchmarks: LivDet2011 and LivDet2013 sets, show that our method achieves outstanding detection results for blind materials and cross-sensor. The size of the model parameters is only 548 KB, and the average detection error of cross-sensors and cross-materials are 15.22 and 1 respectively, reaching the highest level currently available.

Keywords: FLD; lightweight; multi-scale; RSC; blind materials



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Biometric authentication refers to a technology that uses human physiological or behavioral traits for identification. With the rapid growth of multi-media technology, biometric identification has gradually emerged and developed rapidly [1–4]. Compared with traditional identification methods (keys, passwords, ID), biometrics has not been easily lost and forgotten and has been widely used in identity authentication. The biometrics commonly used for identity authentication include fingerprints, palms, faces, veins, etc. Compared with other biometrics, fingerprint-based identification is the most widely deployed and mature because of its uniqueness, stability, and long-term invariance. But it also faces huge hidden risks, such as attacking fingerprint sensors, attacking software systems, and attacking databases. Among them, forged fingerprint spoofing attacks are the threats that need to be addressed most [5,6], since do not need to master much professional knowledge to launch an attack. How to prevent the distinction between live and spoof fingerprints has become a research hotspot.

The aforementioned vulnerabilities can be counteracted using the fingerprint liveness detection (FLD) strategy. Liveness detection refers to the analysis of fingerprint characteristics to determine whether the fingerprint to be tested is from a live subject or from a forged one. In recent years, many fingerprint liveness detection methods (FLD) emerged and are proposed by the research community [7,8]. The current FLD methods are basically divided into two categories [9]: hardware-based and software-based. Hardware-based methods generally assist professional hardware to the fingerprint sensor to obtain vital signs (such as heartbeat, blood flow, skin impedance, smell, etc.), however, these approaches greatly increase the cost; Software-based methods refer to the design a framework of feature extraction, and get discriminant or differential characteristics between the live fingerprint images and the fake one. Moreover, software-based methods be easily integrated into the fingerprint authentication module at the software level without exorbitant hardware costs. Consequently, software-based FLD scheme has broad application value and market prospects.

Software-based methods are further divided into handcrafted and deep learning-based features [10,11]. The former heavily relies on experience and professional knowledge. Therefore, it will lack consideration of the details of the live and forged fingerprint, leading to the loss of some key spatial location information. In contrast, deep learning-based ways can automatically learn deep hidden features from data and achieve better performance. However, they are still some inadequacies and shortcomings that are difficult to solve, such as excessive parameters and mediocre generalization facing new unknown samples [12,13]. In spite of some classical deep learning models applied and reported on fingerprint liveness detection research, they do not consider practical application problems. That is, these models often have hundreds of megabytes of parameters, extremely inconvenient for mobile terminal deployment. Based on the above questions, the research of lightweight FLD methods has urgent practical value. In addition, the neural networks will excessively learn the features from data, so that the model makes the final decision of unknown materials and cross-sensors poor.

Aims at improving the generalization of the trained model of fingerprint liveness detection, the domain generalization (DG) strategy can be used to FLD to improve the reliability of the model and learn the general traits of spoof fingerprints made from blind materials. DG technique is to learn a model with strong generalization ability from new datasets with different data distributions. The current methods of domain generalization can be roughly categorized in three directions: data manipulation, representation learning, and learning strategy [14]. Data manipulation refers to operating on training data and expanding the data in the training set via using some data enhancement operations. Representation learning refers to the invariant features of the learning field, that is, learning a feature that can perform well on data with different distributions. Learning strategy refers to the

introduction of mature learning models in machine learning into multi-field training to make the model more generalized. It mainly includes methods based on ensemble learning and meta-learning. Domain generalization can make fingerprint liveness detection algorithms based on deep learning learn more robust features, improving the performance of fingerprints synthesized in the face of unknown materials.

In order to deal with the problem of the over-parameterization and poor generalization, a lightweight multi-scale FLD method is proposed in this paper, and the related algorithms of domain generalization are optimized so that the trained model can learn more dipartite texture features. Except for testing the performance on two public benchmark datasets, we also test the performance of the model in the FLD of cross-sensor and blind materials. The main contributions of this paper are as follows:

1. Multi-scale lightweight network. A multi-scale parallel neural network is proposed for the fingerprint liveness detection task. The parameter quantity of our network is far less than other traditional CNNs.
2. Hybrid space pooling pyramid. The spatial pooling pyramid divides the feature map into blocks and then uses global pooling to extract global features. Feature maps of different sizes can be extracted after block and global pooling operation. Introducing a spatial pooling pyramid enables the CNN to input fingerprint images of different sizes. Meanwhile, traditional ones often only use one type of pooling, here we simultaneously use the maximum pooling and the average pooling to extract richer features.
3. Optimized RSC (Representation self-challenge) module with attention mechanism. The DG algorithm, RSC is introduced and optimized. Compared with the general DG algorithm, it does not need the information of the target domain and only needs the source domain information to learn robust features. On this basis, we introduced a channel attention mechanism to further improve the performance of the algorithm.
4. Lightweight and generalization. While considering the weights of the model, the generalization ability of the model is also further considered, so that the model not only is less than the general NN in parameters, but also greatly exceeds the traditional CNN in the performance of generalization.

The rest of this paper is organized as follows. In the Section 2, we introduce the proposed method and related techniques. The experimental design and results are reported in the Section 3, and the final conclusion and future work are given in the Section 4.

2 Proposed Methods

In this section, we first introduce the structural details of the lightweight network model, and then design a hybrid spatial pyramid pooling module. Finally, an optimized RSC algorithm with the attention mechanism is introduced. The flowchart of our method is shown in [Fig. 1](#).

2.1 Multi-scale Lightweight Network

To differentiate from those forged fingerprint images, traditional deep learning-based methods often directly increase the depth and width of the network. Among them, the depth of the network refers to the number of layers of the network, and the width refers to the number of channels per layer. However, as the depth and width increase, the number of parameters also rapidly expand, and overfitting often occurs during training. The calculation power of the model will also be greatly

increased simultaneously. To deal with this problem, Szegedy et al. pointed out that the Inception V1 module can widen the network [15], reduce the number of parameters and extract high-dimensional features while ensuring the quality of the model. It uses different sizes of convolution kernels to obtain different sizes of receptive fields, and finally splices the outputs of different convolution modules to merge features of different scales; among them, a large number of 1×1 convolutions are used to reduce the dimensionality of the data. An activation function can also be added to introduce more nonlinearities.

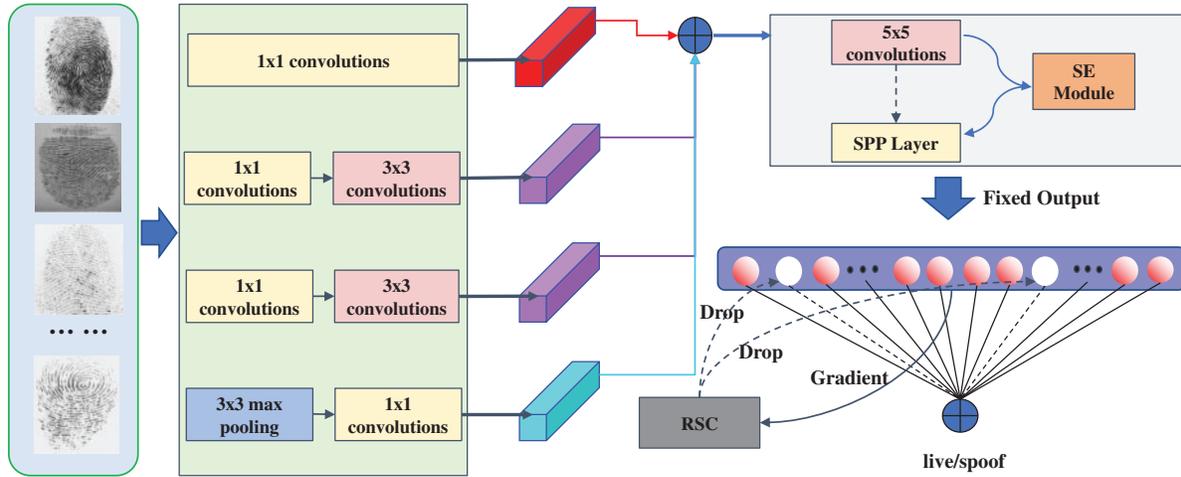


Figure 1: The flowchart of our proposed method

On this basis, we designed a similar parallel architecture module for extracting multi-scale features. This module mainly consists of four parallel convolution modules, as shown in Fig. 1. After performing different convolution operations, the feature maps are combined to obtain high-dimensional features of different scales. After the merged feature map is subjected to another convolutional pooling operation, it is input into the spatial pooling pyramid for block hybrid pooling to obtain a fixed one-dimensional vector, and finally input to the fully connected layer for subsequent detection. The specific structural parameters of the model are shown in Tab. 1.

Table 1: Network architecture and basic layer information

Block	Layers	in_channel	out_channel
Block1	1×1 conv, Sigmod	3	8
Block2_1	1×1 conv	3	8
Block2_2	3×3 conv, Sigmod	8	16
Block3_1	1×1 conv	3	8
Block3_2	3×3 conv, Sigmod	8	16
Block4_1	3×3 MaxPooling	3	3
Block4_2	1×1 conv, Sigmod	3	8

(Continued)

Table 1: Continued

Block	Layers	in_channel	out_channel
Block5	5 × 5 conv, BatchNorm2d, ReLU, 2 × 2 MaxPooling	48	96

The multi-scale of our model is reflected in feature extraction, and our model also supports multi-scale (arbitrary size) input. Through the above design, the size of our model parameters is only 548 KB, which is far smaller than the current two classical models, such as CNN-Alexnet and CNN-VGG [11] as reported in Tab. 2 that 222 and 545 MB respectively. At the same time, our lightweight model is easier to deploy on smart terminals with less computing power, such as mobile phones, platforms, etc.

Table 2: Parameter comparison information of different classic network model

Model	Input size	Total params	Model size (KB)
CNN-Alexnet [8]	(224,224)	57,012,034	222,703
CNN-VGG [8]	(224,224)	139,578,434	545,228
Ours	Arbitrary	140,226	548

2.2 Hybrid Spatial Pyramid Pooling

Spatial pyramid pooling [16], connected to the fully connected layer, can output a fixed-size one-dimensional vector for any size image. Among them, block refers to dividing a feature map into $n \times n$ sub-blocks, global pooling refers to global pooling of each sub-block to obtain a feature map, and finally connecting the features of each sub-block to obtain a one-dimensional feature vector of length n^2 . Spatial pyramid pooling is to use the above operations to connect the features extracted with different sizes of n , and finally input them into the fully connected layer as shown in Fig. 2. During the implementation, the kernel size and step size of the kernel of the pooling operation will be adaptively adjusted according to the size x and the parameter n of the feature map, which achieves the above effects. The specific calculation is shown in Eq. (1), where k_size is the kernel size and s_size is the stride size.

$$\begin{aligned} k_size &= \left\lceil \frac{x}{n} \right\rceil \\ s_size &= \left\lfloor \frac{x}{n} \right\rfloor \end{aligned} \quad (1)$$

Universal spatial pooling pyramids usually use only one type of pooling, that is, max pooling or average pooling. Average pooling operation can extract some background features of fingerprints, while max pooling operation is more inclined to extract texture features of fingerprints. Considering that spoof fingerprints may change the background information and texture features extracted by the sensor at the same time, we perform two kinds of pooling simultaneously to extract more generalizable features. The hybrid spatial pyramid pooling algorithm flow is as follows:

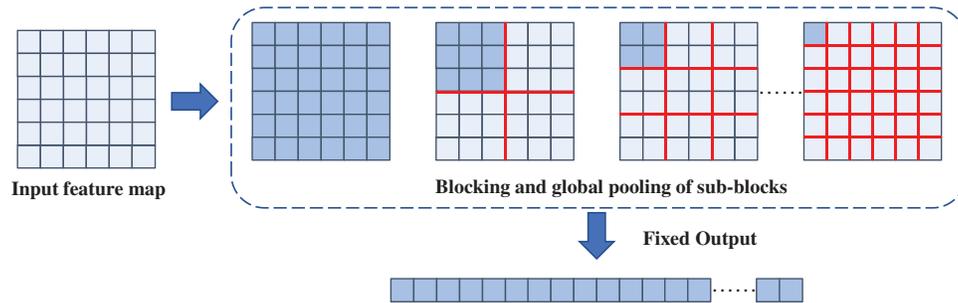


Figure 2: The flowchart of SPP module

Algorithm 1: Hybrid Spatial Pyramid Pooling

Input: n : an int list for the level of block, x : the input feature map, $\text{size}(x) = (\text{width}, \text{height})$

Output: out : a fixed output one-dimensional vector

1. $\text{out} = []$
 2. **for** i **in** n **do**
 3. $k_w, k_h = \lceil \text{size}(x)/i \rceil$
 4. $s_w, s_h = \lfloor \text{size}(x)/i \rfloor$
 5. $\text{out1} = \text{MaxPooling}(x, \text{kernel_size}=(k_w, k_h), \text{stride}=(s_w, s_h))$
 6. $\text{out2} = \text{AveragePooling}(x, \text{kernel_size}=(k_w, k_h), \text{stride}=(s_w, s_h))$
 7. $\text{out.append}(\text{out1.resize}(\text{out1.size}(x), -1))$
 8. $\text{out.append}(\text{out2.resize}(\text{out2.size}(x), -1))$
 9. **return** out
-

2.3 Optimized RSC Algorithm based on SE

The feature self-learning ability is unique to convolutional neural networks, and the overfitting problems often leads to the poorer generalization. Model regularization can improve the performance of the model and effectively improve the generalization of the model. However, most current model regularization uses dropout, randomly discarding the weights of some hidden layer nodes during model training. Those nodes that are not working can be temporarily considered not part of the network structure, and will not participate in the updating for the time being. With the advent of DG method, the RSC strategy is similar to dropout operation. It can improve the generalization and performance of the model by invalidating some neurons during training [17]. Unlike dropout, the RSC algorithm compares gradients to inhibit neurons with higher gradients from participating in propagation. By discarding the features related to higher gradients in each round of training, and forcing the model to use the remaining feature information to make predictions. It restrains the fully connected layer from using the most obvious feature to make predictions, so that the model pays more attention to all the features instead of just one of the most obvious features, thereby improving the generalization capacity of the neural network model.

Assume that the model has one-dimensional features z extracted via hybrid spatial pyramid pooling, which is connected to the fully connected layer. First, pass it through the fully connected layer to get the output y_{pred} and calculate the *loss* with the label y_{true} , then calculate the gradient g_z about z with respect to *loss*. The formula for calculating the gradient is as Eq. (2).

$$g_z = \frac{\partial loss}{\partial z} = \frac{\partial \sum CrossEntropy(y_{true}, y_{pred})}{\partial z} \tag{2}$$

After sorting the gradient of g_z , for the proportion of neurons to be discarded, p , sort each neuron in g_z from large too small to get the ranking $rank_g$ corresponding to the elements in z , and then find the location of the neuron to be discarded according to Eq. (3).

$$location(i) = \begin{cases} 0, rank_g / len(g_z) < p \\ 1, otherwise \end{cases} \tag{3}$$

Finally, the *location* is multiplied by z to get the new feature, which is input into the fully connected layer, and the loss is recalculated and backpropagated.

As mentioned above, the RSC algorithm improves the generalization of the model by suppressing the most obvious feature that is the highest gradient in each training. However, the most obvious feature may not be found by only relying on the gradient. In order to further optimize the performance of the RSC algorithm, we introduce the channel attention mechanism to FLD, and use the Squeeze-and-Excitation (SE) [18] module for the feature map after the last convolution pooling operation, so that when the RSC algorithm calculates the gradient, more obvious features can be found.

The principle of Squeeze-and-Excitation is to first perform the Squeeze operation on the feature map obtained by convolution, and convert the feature map of each channel into a value. The channel-level global features are obtained by global average pooling in this paper as shown in Fig. 3. Then the Excitation operation is carried out on the global features, that is, use the fully connected layer to predict the importance of each channel, learn the relationship between each channel, and get the importance of different channels. This article uses two full connections. The first one reduces the channel feature to 1/16 of the original dimension, the second restores it to the original dimension, and finally the learned channel weight is multiplied by the feature map of the corresponding channel to get a new one. The feature map is then input into the hybrid pyramid pooling and further processed using the RSC algorithm. It is found in the experiment that the optimized algorithm makes it possible to find important features more accurately. The optimized algorithm flow is as follows:

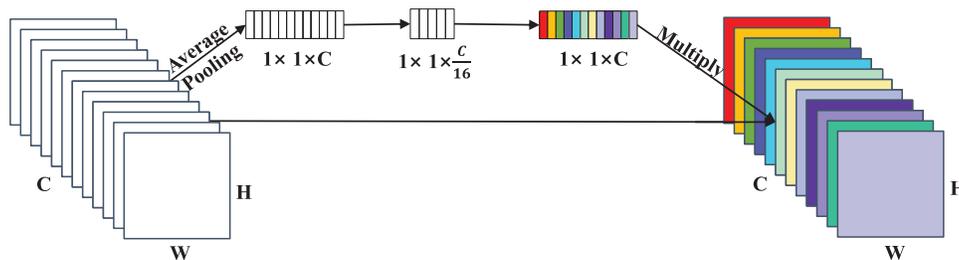


Figure 3: The flowchart of the SE module

Algorithm 2: RSC algorithm optimized by SE

Input: x: feature map, p: RSC drop rate, y: date's true label

Output: model after backpropagating

1. $f = \text{Average Pooling}(x, \text{kernel_size}=\text{size}(x))$
 2. $\text{len} = \text{length}(f)$
 3. $w1 = \text{linear}(f, \text{input_size} = \text{len}, \text{output_size} = \text{len}/16)$
 4. $w2 = \text{linear}(f, \text{input_size} = \text{len}/16, \text{output_size} = \text{len})$
 5. $x_{\text{new}} = x.*\text{sigmoid}(w2)$
 6. $z = \text{Algorithm1}(n = [1,2,3,4,5], x = x)$
 7. $y_{\text{pred1}} = \text{classifier}(z)$
 8. $\text{loss1} = \text{CrossEntropy}(y_{\text{pred1}}, y)$
 9. $g_z = \text{grad}(\text{loss1}, z)$
 10. calculate m with Eq. (3) using p and g_z
 11. $z_{\text{new}} = z.*m$
 12. $y_{\text{pred2}} = \text{classifier}(z_{\text{new}})$
 13. $\text{loss2} = \text{CrossEntropy}(y_{\text{pred2}}, y)$
 14. do backpropagation using loss2
-

3 Experimental Details and Performance Evaluation

The general fingerprint liveness detection dataset used to evaluate the performance of the proposed method and the evaluation indicators are introduced in this section, and then the operation details of the experiment are given. Finally, we compare the performance of the algorithms in different scenarios and illustrate the effectiveness of our proposed method.

3.1 Dataset and Evaluation Metric

The performance of this method is evaluated on two public datasets LivDet 2011 and LivDet 2013 [19,20]. The LivDet 2011 dataset is derived from the 2011 FLD challenge. It is publicly downloaded after registration and contains 12004 real and fake fingerprints. It is collected by four different kinds of sensors, namely Biometrika, Digital, Italdata and Sagem. LivDet 2013 consists of four sub-datasets, Biometrika, ItalData, Swipe, and CrossMatch, each of which is captured by a corresponding fingerprint reader. LivDet 2013's fake fingerprints are made of gelatin, latex, ecoflex, modasil, and wood glue. The specific details of the two fingerprint data sets are shown in Tab. 3.

The metric used for performance evaluation is the average classification error (ACE). As shown in Eq. (4), it is obtained by averaging the misclassification rate of live fingerprints ($F_{errlive}$) and false fingerprint misclassification rate ($F_{errfake}$).

$$ACE = \frac{F_{errfake} + F_{errlive}}{2} \quad (4)$$

Table 3: The detailed data information in LivDet 2011 and LivDet 2013

Dataset	Sensor (Acronym)	Size	Training (Live/Spoof) Testing (Live/Spoof)
LivDet2011	Biometrika (Bio)	312 × 372	1000/1000 1000/1000
	Digital (Dig)	355 × 391	1004/1000 1000/1000
	Italdata (Ita)	640 × 480	1000/1000 1000/1000
	Sagem (Sag)	352 × 384	1008/1008 1000/1036
LivDet2013	Biometrika (Bio)	312 × 372	1000/1000 1000/1000
	Italdata (Ita)	640 × 480	1000/1000 1000/1000
	CrossMatch (Cro)	800 × 750	1250/1000 1250/1000
	Swipe (Swi)	208 × 1500	1221/979 1153/1000

3.2 Implementation Details

The environment used in the experiment is Python: 3.6.5, PyTorch: 1.7.1, and the GPU used to run the experiment is GTX 1080Ti. The experimental data has been enhanced in advance through four methods: small-angle rotation, flipping, zooming, and brightness enhancement methods. The parameters used by RSC in the experiment refer to [17], gradient pruning rate is 0.3. In the cross-material experiment, the unnecessary fake fingerprints corresponding to the training dataset and the test dataset are deleted, and all the real fingerprints keep unchanged. Meanwhile, to speed up the training and better compare with the results of others, all the fingerprint images are resized to 224 × 224 for batch training, and the batch size is 32. And we train each model for only 60 epochs.

3.3 Results and Analysis

First, we perform performance tests on the datasets of sensors corresponding to LivDet 2011 and LivDet 2013, and the results are shown in Tabs. 4 and 5. Regardless of whether it is LivDet 2011 or LivDet 2013, our results obviously exceed the general CNN and methods, and the average ACE which attains 2.47 and 1.49, is better than others.

Table 4: The average classification error of different methods when dataset is LivDet 2011

Method	LivDet 2011				
	Biometrika	Digital	Italdata	Sagem	Average
LCPD [21]	4.9	4.2	11	2.7	5.7
CNN-Alexnet [11]	5.6	4.6	9.1	3.1	5.6
ROI [22]	7.6	2.1	11	2.5	5.8
CNN-VGG [11]	5.2	3.2	8	1.7	4.53
HSIC [23]	3.5	1.35	5.1	1.55	2.88
Ours	2.05	2	4	1.82	2.47

Table 5: The average classification error of different methods when dataset is LivDet 2013

Method	LivDet 2013				
	Biometrika	Italdata	CrossMatch	Swipe	Average
SURF [24]	5.75	6.08	4.6	4.6	5.26
CNN-Alexnet [11]	1.9	0.5	4.7	4.3	2.85
ROI [22]	2	1.8	7.42	0.74	2.99
CNN-VGG [11]	1.8	0.4	3.4	3.7	2.32
HSIC [23]	1.4	0.6	4.53	1.11	1.91
Ours	0.8	0.3	2.76	2.09	1.49

Then, we test the performance of our method in the case of cross-sensor, and also compared the performance of the RSC algorithm without SE module optimization and the RSC algorithm with SE module optimization on our lightweight model, as shown in Tab. 6. It can be seen that the performance after the SE optimization has been improved by about 1.7, and the generalization performance on the cross-sensor has been greatly improved compared with the general method, that the average ACE achieves 15.22.

Table 6: The Average Classification Error from cross-sensor detection

Train	Test	[11]	[25]	Without SE	ours
Bio2011	Bio2013	15.5	7.9	6.6	6.35
Bio2013	Bio2011	46.8	34.4	39.55	30.7
Bio2013	Ita2013	8.8	6.7	1.7	1.85
Ita2013	Bio2013	2.3	5.1	3.25	1.45
Bio2011	Ita2011	37.2	29.5	11.4	9.05
Ita2013	Bio2011	31.0	24.9	33.65	34.45
Ita2011	Ita2013	14.6	3.3	8.55	6.4
Ita2013	Ita2011	46.0	29.9	31.05	31.5
Average		25.28	17.71	16.97	15.22

Finally, we test the performance under cross-material conditions in Bio2013 and Ita2013, as shown in Tab. 7. We only keep spoof fingerprints made of Modalsil and Wood Glue in the training dataset, and only keep spoof fingerprints made of EcoFlex, Gelatine, and Latex in the test dataset.

Table 7: The average classification error from cross-material detection

Data	Materials used to fabricate fake fingerprints		Nogueira et al. [11]		Chugh et al. [25]	
	Materials for training	Materials for testing	CNN-VGG	LBP		
Bio2013	Modalsil, Wood Glue	EcoFlex, Gelatine, Latex	4.9	8.5	3.1	1.75
Ita2013	Modalsil, Wood Glue	EcoFlex, Gelatine, Latex	6.3	10.7	0.8	0.25
Average result			5.6	9.6	1.95	1

From the accuracy curve in Fig. 4, it can be seen that in the early stage of model training, the accuracy fluctuates greatly, but after about 20 epochs of training, it gradually becomes stable. In the early stage of training, the RSC algorithm could inhibit some of the more significant neurons, resulting in unstable results. In this process, the model is effectively prevented from overfitting, so that our method can finally learn more representative true/fake fingerprint features, thereby improving the generalization performance of the model. Our method also performed well on two cross-material trails as shown in Tab. 7, and the average ACE across materials reached 1.

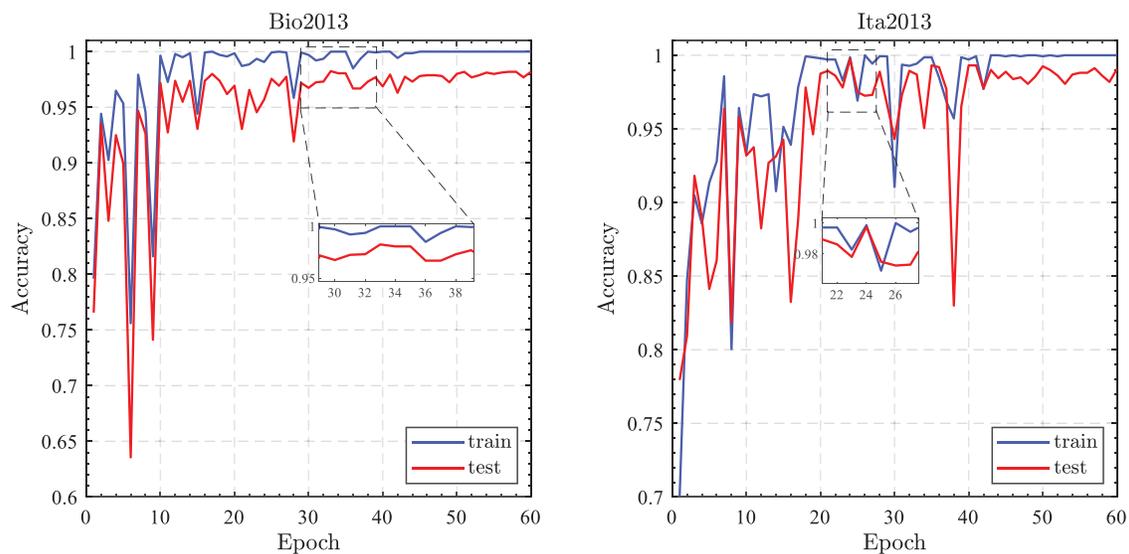


Figure 4: Train and test accuracy curve from cross-material detection

3.4 Supplementary Experiment

In order to better verify the superiority of the lightweight architecture proposed in this paper, we use the RSC algorithm to test the two mainstream lightweight network structures MobileNet [26] and ShuffleNet [27], and compares the performance with our proposed network.

First, we compare the parameters of each network. The parameters of MobileNet and ShuffleNet reach 12,521 and 13,788 KB as shown in Tab. 8, while our proposed method is only 548 KB, which is less than one-twentieth of the two mainstream networks. Our proposed architecture is lightweight enough.

Table 8: Parameter comparison of different lightweight architectures

Model	Input size	Total params	Model size (KB)
MobileNet [26]	(224,224)	3,202,976	12,512
ShuffleNet [27]	(224,224)	3,529,776	13,788
Ours	Arbitrary	140,226	548

Then, experiments are carried out in LivDet 2011 and LivDet 2013, and the obtained results are shown in Tabs. 9 and 10. It can be seen that in the LivDet 2011, MobileNet's average performance is 4.39 worse than ours, and ShuffleNet is 3 worse. In LivDet 2013, MobileNet is 0.76 worse than ours, and ShuffleNet is 0.01 worse. This point proves that the architecture designed in this paper performs better on the fingerprint liveness detection task.

Table 9: Different lightweight architectures' ACE when dataset is LivDet 2011

Method	LivDet 2011				
	Biotrika	Digital	Italdata	Sagem	Average
MobileNet [26]	8.4	2.65	13.7	2.7	6.86
ShuffleNet [27]	7.5	2.65	10	1.72	5.47
Ours	2.05	2	4	1.82	2.47

Table 10: Different lightweight architectures' ACE when dataset is LivDet 2013

Method	LivDet 2013				
	Biometrika	Italdata	Crossmatch	Swipe	Average
MobileNet [26]	0.85	0.75	5.87	1.53	2.25
ShuffleNet [27]	0.85	0.5	3.78	0.89	1.50
Ours	0.8	0.3	2.76	2.09	1.49

Cross-sensor and cross-material experiments are then performed on MobileNet and ShuffleNet. When cross-sensor, our proposed method outperforms MobileNet by 9.26 on average and 8.12 better than ShuffleNet, as shown in Tab. 11. When cross-material, ours outperforms MobileNet and ShuffleNet by 4.45 and 0.95, respectively, as shown in Tab. 12.

From the comparison of the results of our proposed method and two common lightweight models under the training of an optimization algorithm, it can be seen that our model is not only more lightweight than them, but also has less than one-twentieth of their parameters. Meanwhile, our proposed method outperforms the other two models under the test of each task, which verifies the superiority of the lightweight architecture proposed in this paper.

Table 11: Different lightweight architectures' ACE from cross-sensor detection

Train	Test	MobileNet [26]	ShuffleNet [27]	ours
Bio2011	Bio2013	20.75	18.9	6.35
Bio2013	Bio2011	43.55	41.9	30.7
Bio2013	Ita2013	4.55	5.9	1.85
Ita2013	Bio2013	2.9	3.45	1.45
Bio2011	Ita2011	36.85	35.4	9.05
Ita2013	Bio2011	33.65	28.8	34.45
Ita2011	Ita2013	11.7	16.15	6.4
Ita2013	Ita2011	41.85	36.25	31.5
Average		24.48	23.34	15.22

Table 12: Different lightweight architectures' ACE from cross-material detection

Data	Materials used to fabricate fake fingerprints		[26]	[27]	ours
	Materials for training	Materials for testing			
Bio2013	Modalsil, Wood Glue	EcoFlex, Gelatine, Latex	7.19	2.71	1.75
Ita2013	Modalsil, Wood Glue	EcoFlex, Gelatine, Latex	3.71	1.19	0.25
Average result			5.45	1.95	1

4 Conclusion and Further Work

This paper proposes a lightweight multi-scale neural network for fingerprint liveness detection. The network achieves multi-scale feature extraction with a small number of parameters. At the same time, the introduced hybrid spatial pyramid pooling can make the model input fingerprints figure of any size image. Furthermore, for unknown forged fingerprints, we designed an optimized RSC algorithm based on the SE attention mechanism, greatly improving the generalization of the detection model, and achieved excellent performance in cross-sensor and cross-material experiments.

In general, compared with other fingerprint liveness detection scheme, our proposed solution has the characteristics of lightweight, great performance and excellent generalization. In the future, we will further study the performance of this method in other fields such as forged face detection, and introduce simultaneously more domain generalization methods to further improve the performance of attack detection against spoofed fingerprints from unknown material. Moreover, the design of our

proposed architecture and the setting of parameters are manually adjusted depending on the personal experience, it can be combined with the network architecture search to develop a better lightweight fingerprint liveness detection model.

Funding Statement: This work is supported by the National Natural Science Foundation of China under grant, 62102189, U1936118, U1836208, U1836110, 62122032; by the Jiangsu Basic Research Programs-Natural Science Foundation under grant BK20200807; by the Key Laboratory of Public Security Information Application Based on Big-Data Architecture, Ministry of Public Security (2021DSJSYS006); by the Research Startup Foundation of NUIST 2020r15.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] P. Jayapriya and K. Umamaheswari, "Performance analysis of two-stage optimal feature-selection techniques for finger knuckle recognition," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1293–1308, 2022.
- [2] N. H. Alsaedi and E. S. Jaha, "Dynamic audio-visual biometric fusion for person recognition," *Computers Materials & Continua*, vol. 71, no. 1, pp. 1283–1311, 2022.
- [3] R. Srivastava, R. Tomar, A. Sharma, G. Dhiman, N. Chilamkurti *et al.*, "Real-time multimodal biometric authentication of human using face feature analysis," *Computers, Materials & Continua*, vol. 69, no. 1, pp. 1–19, 2021.
- [4] J. Xu and W. Chen, "Convolutional neural network-based identity recognition using ecg at different water temperatures during bathing," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1807–1819, 2022.
- [5] G. Gao and G. Jiang, "Bessel-fourier moment-based robust image zero-watermarking," *Multimedia Tools and Applications*, vol. 74, no. 3, pp. 841–858, 2015.
- [6] R. F. Nogueira, R. D. A. Lotufo and R. C. Machado, "Evaluating software based fingerprint liveness detection using convolutional networks and local binary patterns," in *IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications*. Rome, Italy, 22–29, 2015.
- [7] S. B. Nikam and S. Agarwal, "Texture and wavelet-based spoof fingerprint detection for fingerprint biometric systems," in *First Int. Conf. on Emerging Trends in Engineering and Technology*, Nagpur, Maharashtra, India, pp. 675–680, 2008.
- [8] D. Gragnaniello, G. Poggi, C. Sansone and L. Verdoliva, "Fingerprint liveness detection based on weber local image descriptor," in *IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications*. Naples, Italy, 46–50, 2013.
- [9] S. Memon, N. Manivannan, A. Noor, W. Balachadran and N. V. Boulgouris, "Fingerprint sensors: Liveness detection issue and hardware based solutions," *Sensors & Transducers*, vol. 136, no. 1, pp. 35, 2012.
- [10] P. V. Reddy, A. Kumar, S. M. K. Rahman and T. S. Mundra, "A new antispooofing approach for biometric devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 4, pp. 328–337, 2008.
- [11] R. F. Nogueira, R. D. A. Lotufo and R. C. Machado, "Fingerprint liveness detection using convolutional neural networks," *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 6, pp. 1206–1213, 2016.
- [12] Y. Zhang, S. Pan, X. Zhan, Z. Li, M. Gao *et al.*, "Fldnet: Light dense cnn for fingerprint liveness detection," *IEEE Access*, vol. 8, pp. 84141–84152, 2020.
- [13] W. Sun, G. C. Zhang, X. R. Zhang, X. Zhang and N. N. Ge, "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30803–30816, 2021.
- [14] J. Wang, C. Lan, C. Liu, Y. Ouyang, W. Zeng *et al.*, "Generalizing to unseen domains: A survey domain generalization," 2021. [Online]. Available: <https://arxiv.org/abs/2103.03097>.

- [15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.*, “Going deeper with convolutions,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 1–9, 2015.
- [16] K. He, X. Zhang, S. Ren and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [17] Z. Huang, H. Wang, E. P. Xing and D. Huang, “Self-challenging improves cross-domain generalization,” in *Computer Vision-ECCV 2020: 16th European Conf.*, Glasgow, UK, pp. 124–140, 2020.
- [18] J. Hu, L. Shen and G. Sun, “Squeeze-and-excitation networks,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7132–7141, 2018.
- [19] D. Yambay, L. Ghiani, P. Denti, G. L. Marcialis, F. Roli *et al.*, “LivDet 2011—fingerprint liveness detection competition 2011,” in *Int. Conf. on Biometrics (ICB)*, New Delhi, India, pp. 208–215, 2012.
- [20] L. Ghiani, D. Yambay, V. Mura, S. Tocco, G. L. Marcialis *et al.*, “Livdet 2013 fingerprint liveness detection competition 2013,” in *Int. Conf. on Biometrics (ICB)*, Phuket, Thailand, pp. 1–6, 2013.
- [21] D. Gragnaniello, G. Poggi, C. Sansone and L. Verdoliva, “Local contrast phase descriptor for fingerprint liveness detection,” *Pattern Recognition*, vol. 48, no. 4, pp. 1050–1058, 2015.
- [22] C. Yuan, Z. Xia, X. Sun and Q. M. J. Wu, “Deep residual network with adaptive learning framework for fingerprint liveness detection,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, no. 3, pp. 461–473, 2019.
- [23] C. Yuan, J. Chen, M. Chen and W. Gu, “A lightweight CNN using HSIC fine-tuning for fingerprint liveness detection,” in *Chinese Conf. on Biometric Recognition*, Shanghai, China, pp. 240–247, 2021.
- [24] R. K. Dubey, J. Goh and V. L. L. Thing, “Fingerprint liveness detection from single image using low-level features and shape analysis,” *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 7, pp. 1461–1475, 2016.
- [25] T. Chugh, C. Kai and A. K. Jain, “Fingerprint spoof buster: use of minutiae-centered patches,” *IEEE Transactions on Information Forensics & Security*, vol. 13, no. 9, pp. 2190–2202, 2018.
- [26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications 2017”<https://arxiv.org/abs/1704.04861>.
- [27] X. Zhang, X. Zhou, M. Lin and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 6848–6856, 2018.