

Received October 24, 2019, accepted December 11, 2019, date of publication December 17, 2019, date of current version December 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960411

GAN-Based Image Augmentation for Finger-Vein Biometric Recognition

JIANFENG ZHANG^{1,2}, ZHIYING LU¹, MIN LI¹, AND HAOPENG WU¹

¹School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

²Information Research Institute of Shandong Academy of Sciences, Qilu University of Technology, Jinan 250014, China

Corresponding author: Zhiying Lu (luzy@tju.edu.cn)

ABSTRACT Deep learning methods, and especially convolutional neural networks (CNNs), have made a considerable breakthrough in various fields of machine vision, basically by employing large-scale labeled databases. However, deep learning methods applied in finger-vein area are basically implemented on small-scale datasets, which are probably faced with challenges such as overfitting, susceptible to finger position, unstable performance on various datasets and son on. In this study, we present a lightweight and fully convolutional Generative Adversarial Network (GAN) architecture, which is named FCGAN, using preliminary batch normalization, and tightly-constrained loss function for implementing finger-vein image augmentation. In addition, we present a novel scheme FCGAN-CNN for finger-vein classification, which reveals that synthetic finger-vein images using FCGAN are capable of improving the property of CNN for finger-vein image classification. The experiment of sample augmentation shows that the training accuracy using FCGAN-augmented samples could go beyond 99%, which is higher than 96.34% obtained using only classic sample augmentation. Furthermore, the well-trained CNN is further evaluated on a totally different dataset, which indicates that the proposed scheme FCGAN-CNN is capable of improving the ability of CNN to extract deep features. We consider that the proposed method for sample augmentation could be extended to other biometric systems.

INDEX TERMS Sample augmentation, convolutional neural networks, generative adversarial networks, finger-vein classification.

I. INTRODUCTION

With increasing growth in the demand for security, biometric systems have attracted considerable attention. However, most biological patterns are susceptible to spoofing attacks [1], [2], more user-friendly, low-cost and highly secure biological patterns such as palm-vein [3], finger-vein [4] have received much attention, since they belong to the intrinsic modalities inside humans' body and are difficult to forge. Compared to palm-vein, finger-vein imaging device is more exquisite and easy to assemble, and it makes finger-vein recognition technology conducive to popularize. As shown in Fig. 1, near-infrared light is absorbed by the hemoglobin in blood vessels when it passes through the finger, and vein patterns ultimately appear as dark networks in the acquired image.

The associate editor coordinating the review of this manuscript and approving it for publication was Orazio Gambino¹.

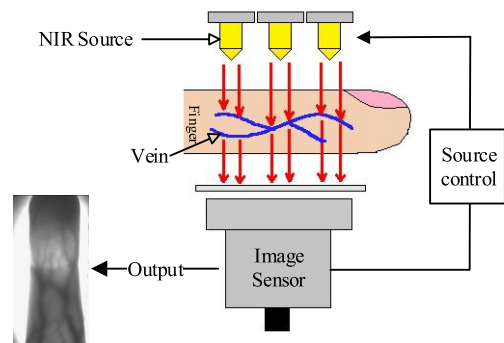


FIGURE 1. The structure of finger-vein image acquisition system.

A. RELATED WORK

Finger-vein recognition technology is a burgeoning research field, and meanwhile it faces enormous challenges. Since the acquisition process is inherently affected by various

factors such as uneven illumination [5], [6], light scattering inside finger tissues [7], [8] and ambient temperature [4], [9], a majority of finger-vein images inevitably contain blurred areas where venous and non-venous regions cannot be distinguished easily. Furthermore, since the acquisition system is mainly designed for non-contact, different finger postures can lead to displacement or deformation of acquired finger-vein images [10], [11]. Generally speaking, these uncontrolled factors mentioned above could ultimately degrade the performance of finger-vein recognition system. To overcome these issues, a huge number of approaches have been proposed to extract more robust features, which aim at improving the property of identification systems. Based on different descriptions of finger-vein features, they can be basically classified into two categories:

- 1) *Skeleton-based methods*. In this category, finger-vein patterns are considered as dark networks on the relatively bright background. In other words, they focus on extracting the whole vein network accurately from the complex background. Contrast-based methods [12]–[14] were proposed to enhance the contrast between the vein network and background. Since the cross section of vein networks shows valley forms, different valley detected methods [15]–[18] were presented to directly detect the entire vein network using repeated line and curvature. In addition, the atmospheric scattering model [19] was applied to obtain the restored finger-vein image and showed power of extracting the vein network from blurred images.
- 2) *Minutiae-based methods*. Since the recognition result of using vein networks extracted by algorithms is susceptible to finger shift and rotation, a number of minutiae-based methods, which are invariant to rotation, have been proposed to improve the recognition accuracy. Local binary pattern (LBP) [20], local line binary pattern (LLBP) [21] and pyramid histogram of double competitive pattern (PHDCP) [22] encoded the vein pattern using local statistical information. Relying on the bifurcation and termination of veins, various methods [23]–[26] were proposed to extract such key points with different feature descriptor. Furthermore, the scale invariant feature transform (SIFT) [27], [28] was employed to acquire minute features that are more robust to rotation and translation.

Most of the above approaches have achieved promising results in improving finger-vein recognition, whereas they still suffer from some drawbacks, mainly related to the way itself extracting features. For skeleton-based methods, they are hard to accurately acquire vein networks in a poor-quality finger-vein image, whose acquisition process may be affected by ambient lighting conditions and infrared light scattering inside fingers. Besides that, temperature variation could influence the thickness of veins, and it has a negative impact on the final matching accuracy. For minutiae-based methods, noise

inherently existing in the image could seriously affect the accuracy of extracting feature points.

Inspired by tremendous achievements made by deep neural networks recently in various fields of computer vision, a number of researchers have tried to solve problems existing in finger-vein recognition by using deep learning methods. By using a set of vein extraction algorithms, an approach [29] for automatically labeling the vein network was proposed to solve the problem of lacking annotated datasets. Then, CNN was trained for obtaining the vein network by itself. Fang *et al.* [30] proposed a lightweight finger-vein verification system based on two-stream convolutional network. They took two finger-vein images as input of the network and solve the finger posture issue by extracting the mini-ROI from the original image. Furthermore, inspired by the VGG-Net-16 model [31], different deep CNNs were presented [32]–[34] and achieved good performances on different finger-vein datasets. Most of methods above focused on the innovation of CNNs architecture to improve finger-vein recognition accuracy, and indeed they achieved inspiring results.

However, except optimizing the framework of models, data source also plays an important role in training procedure, since models are totally driven by data. For instance, GoogLeNet is often taken as a fine-tuned model, which is attributed to its pre-training by using a large number of various images. Besides that, insufficient training data could lead models to overfitting status, which needs to be avoided in real applications. Finger-vein recognition system is facing similar problems, which could be summarized as follows:

- Although deep learning methods have achieved good results in various finger-vein applications, they are still susceptible to training dataset and exhibit several problems, such as training on the small-scale dataset, unstable classification performance on different datasets and so on.
- Since the acquisition equipment is basically designed to be non-contact, offset and deformation which are caused by finger shift and rotation could degrade the robustness of deep learning methods and reduce the accuracy of finger-vein verification system in real-life scenario.

In order to overcome above issues, in this study an effective GAN-based architecture is presented to generate high-quality and diverse finger-vein images and accomplish the sample augmentation. In addition, the augmented sample is further employed for improving the classification accuracy of proposed CNN on finger-vein images.

The main contributions of this study are summarized as follows:

- 1) A fully convolutional GAN-based architecture named FCGAN is proposed to generate high-quality and diverse finger-vein images.
- 2) FCGAN is designed as a fully convolutional structure to solve the problem that the existing GAN-based

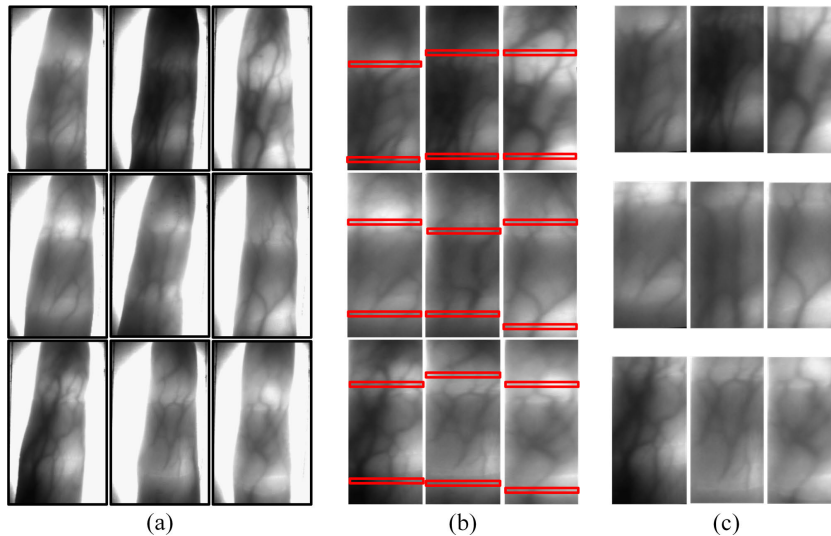


FIGURE 2. Finger-vein ROIs extracted using a gradient-based phalangeal joint detection approach. (a) Examples of HKPU dataset, (b) Phalangeal joints detection on corrected image, (c) ROIs with fixed size 160×80 .

methods are likely to produce grid effect due to the deconvolution operation.

- 3) A preliminary batch normalization is presented and used in FCGAN to improve the diversity of synthetic finger-vein images.
- 4) A tightly-constrained loss function with a penalty term is employed in FCGAN to relieve the unilateral dominance and unbalance between generator and discriminator during the training procedure of GANs.
- 5) We present a novel scheme FCGAN-CNN for finger-vein classification, which reveals that synthetic finger-vein images using FCGAN are capable of improving the property of CNN for finger-vein image classification.

II. CLASSICAL SAMPLE AUGMENTATION

As described above, deep learning methods applied in finger-vein applications are still susceptible to the training dataset and faced with many challenges. Thus, to expand the training dataset and improve the recognition performance, we augment the samples in two ways:

- Firstly, we increase samples using classic augmentation techniques which try to simulate the deformation phenomenon occurred in real scenarios.
- Using generated samples with classical augmentation techniques, we further augment samples utilizing the proposed FCGAN.

In this section we first describe the original finger-vein dataset and their specific properties. Then we introduce the preprocessing method and classical augmentation techniques in detail. In the next section, we will elaborate the structure of proposed FCGAN for synthesizing high-quality finger-vein images.

A. ORIGINAL DATA

The original data used in our work is obtained from Hong Kong Polytechnic University database, which is introduced briefly as follows:

The Hong Kong Polytechnic University finger-vein image database [13] consists of 3132 images acquired from 156 volunteers using an opening and contactless imaging device. The finger-vein images belong to 105 subjects were captured in two separate sessions, in which each subject provided 6 image samples from the index finger and middle finger respectively and the size of the original image is 513×256 pixels. The remaining 51 subjects only provided images of one single session.

B. PREPROCESSING

In our study, a gradient-based algorithm is used to extract finger-vein ROIs from original images. The ROI extraction process can be briefly summarized as three steps.

Firstly, finger-vein images are corrected in vertical direction based on the finger contour. Then a sub region, the maximal inscribed rectangle of the finger area, is segmented. Finally, two phalangeal joints are located in the sub region, which helps to obtain the whole ROI. The detailed ROI extraction process could be referred to our previous work in [35].

Some examples of original finger-vein images and extracted ROIs are shown in Fig. 2. Note that the extracted ROIs are totally normalized to fixed size of 160×80 .

There are two advantages by taking finger-vein ROIs as input for training CNNs:

- 1) The displacement of vein patterns is corrected by using ROIs extraction method [35], which could improve the classification stability of CNNs.

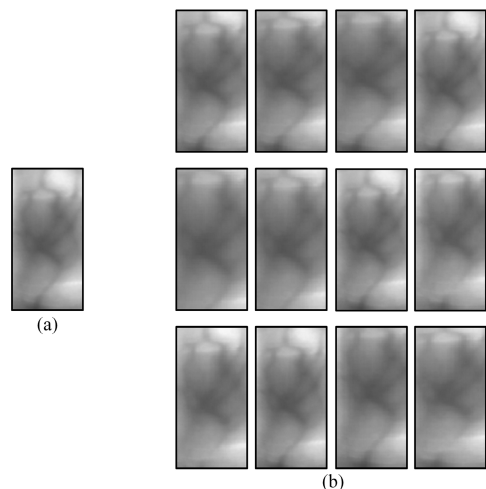


FIGURE 3. An example of finger-vein image augmentation. (a) Original finger-vein ROI, (b) Examples of image augmentation by using translation.

- 2) Most irrelevant areas are filtered and primary vein patterns are preserved, which could improve the training process effectively.

C. CLASSIC AUGMENTATION TECHNIQUES

As an effective solution to reduce overfitting, sample augmentation mainly includes geometric transformations such as rotation, scaling, flipping, cropping and translation [36], [37]. Since the inherent characteristic of finger-vein patterns and finger-vein images have been cropped into ROIS, some augmentation techniques like flipping and cropping are not suitable for finger-vein image augmentation. For instance, a finger-vein image will be changed into another category if it is flipped horizontally. Moreover, our purpose is to enlarge and enrich the dataset, which can better reflect different vein pattern deformations in real scenarios.

Thus, in our study we realize the image augmentation by using translation in different direction. The translation operation could be concretely represented as T_δ^θ , where $\theta = \{x|x = \frac{360^\circ}{N_\theta} \times i, 1 \leq i \leq N_\theta\}$ denotes the translation orientation which consists of N_θ directions, and $\delta = \{y|1 \leq y \leq N_\delta, y \in \mathbb{N}^+\}$ represents the translation distance which is measured by pixels. All these images translated by T_δ^θ are resized to fit a uniform size of 160×80 pixels using bicubic interpolation. As a result of above augmentation procedure, the total amount of augmentations is up to $N = N_\theta \times N_\delta$. An example of finger-vein image augmentations is shown in Fig. 3.

III. GANS FOR FINGER-VEIN IMAGE AUGMENTATION

GAN architecture was first proposed by [38], which aims at learning to represent an estimation of the sample distribution which includes samples $\{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$ drawn from a distribution P_{data} . Afterwards, a number of GAN-based augmentation methods have been presented to deal with problems in various pattern recognition tasks, such as palmprint

recognition [39], finger spelling recognition [40], person re-identification [41] and so on.

The nascent methods were suitable to be applied in small-scale datasets like MNIST and could not achieve a stable performance, until the Deep Convolutional GAN (DCGAN) [42] was proposed, which provided a fundamental framework for solving existing issues. Along with the requirement for generating images with high definition, ResNet [43] unit was integrated into GANs to increase depth and nonlinear capability of networks. As a typical ResNet-based GANs, the Self-Attention GANs (SAGAN) [44] achieved a remarkable performance on generating more exquisite facial images, by introducing the attention map and spectral normalization.

A. ARCHITECTURE OF PROPOSED FCGAN

Although DCGAN [42] achieved encouraging results in various vision tasks, it still faced several challenges. Firstly, it was likely to produce grid effects in synthetic images because of using deconvolutions with a large stride in the generator network [45].

Secondly, the batch normalization (BN) was efficient for accelerating the training procedure of discriminator network, but it simultaneously put side effect on the generator network, which is analyzed as follows:

Generally, a set of random noise which conforms to the prior probability distribution is taken as the input of generator network. However, the BN layer breaks this randomness and degrades the independence among synthetic samples, which eventually makes the generator network tend to synthesize samples with the same tone under the same batch of training data.

In addition, since the loss function totally depends on the label-data to implement weights learning, it could lead to unilateral dominance and unbalance between generator and discriminator during alternate training [46].

To deal with the above issues, we present a fully convolutional GAN architecture, which is named FCGAN, using tightly-constrained loss function for implementing finger-vein image augmentation. The main contributions of the proposed architecture FCGAN consist of three aspects:

- 1) A global pooling is integrated into discriminator network to extract representative features and support the construction of penalty term which is able to accelerate the convergence of proposed FCGAN.
- 2) The generator network is designed with a fully convolutional network to resist grid effect induced by the deconvolution operation.
- 3) A preliminary batch normalization (PBN) scheme is presented and used in the generator network to improve the diversity of synthetic samples.

The architecture of proposed FCGAN is shown in Fig. 4. Specifically, the generator network consists of 4 convolutional layers with a $[3 \times 3]$ kernel size, 4 up-sampling layers and 4 PBN layers. ReLU function is taken as the activation

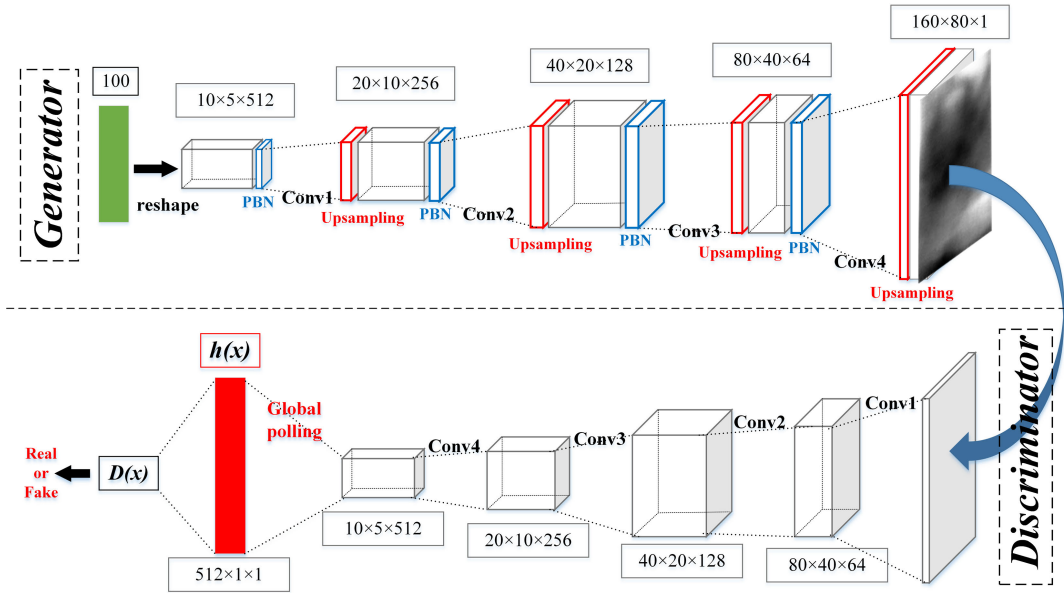


FIGURE 4. The architecture of proposed FCGAN.

function of each layer other than the final layer that employs the tanh function. In addition, the convolutional stride and padding are both set to 1.

The discriminator network consists of 4 convolutional layers with a $[3 \times 3]$ kernel size and one global polling layer with a $[10 \times 5]$ kernel size. ReLU function is taken as the activation function of each layer other than the final layer that employs the sigmoid function. In addition, the convolutional stride and padding are both set to 2.

The detailed description is given as follows.

1) PENALTY TERM

The primary concept of classical GANs is to establish a game between two players, which could be represented by

$$\min_G \max_D \mathbb{E}_{x \sim P_{data}} \log D(x) + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] \quad (1)$$

where x represents the real sample belonging to the distribution P_{data} , z denotes a random series belonging to the distribution P_z which obeys a normal distribution or uniform distribution. D and G represent the discriminator and generator respectively.

The ideal condition is that the generator and discriminator supervise and motivate each other to make synthetic samples $G(z)$ extremely approach to the distribution P_{data} . Nevertheless, sometimes the unilateral dominance and unbalance occurs in the real training process, which slows down the training procedure. It could be mainly attributed to the fact that back propagation excessively relies on label-data and lacks attention to the similarity between real and synthetic samples.

Therefore, this paper integrates a penalty term into the loss function to constrain the similarity between real and synthetic

samples and the specific definition is given as follows.

$$\|\mathbb{E}_{x \sim P_{data}} h(x) - \mathbb{E}_{z \sim P_z} h(G(z))\|_2^2 \quad (2)$$

Here a global polling layer is inserted in front of the discriminator network's output layer to extract representative features with 512 dimensions, as shown in Fig. 4. Then L2-norm is used to measure the distance between feature vector $h(x)$ and $h(G(z))$, where $h(x)$ is the output of global polling layer in the discriminator network. Finally, the loss function can be rewritten as

$$\min_G \max_D \mathbb{E}_{x \sim P_{data}} \log D(x) + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))] + \min \|\mathbb{E}_{x \sim P_{data}} h(x) - \mathbb{E}_{z \sim P_z} h(G(z))\|_2^2 \quad (3)$$

Note that in the training procedure of each mini-batch we calculate the average value of feature vectors, which means $h(x)$ is computed by

$$h(x) = \frac{1}{m} \sum_{i=1}^m h(x^{(i)}) \quad (4)$$

where m is the mini-batch size. Then Eq. 2 can be rewritten as

$$\mathcal{E} = \frac{1}{m} \sum_{i=1}^m \|h(x^{(i)}) - h(G(z^{(i)}))\|_2^2 \quad (5)$$

The discriminator and generator are trained alternately. For the training process of generator, the synthetic samples $G(z)$ is taken into the discriminator and the produced loss value $loss_G$ is transmitted back to the generator for updating weights using the back propagation algorithm, as shown in Fig. 5.

Here the stochastic gradient descent method is employed to learn the weights of each layer in the generator. The

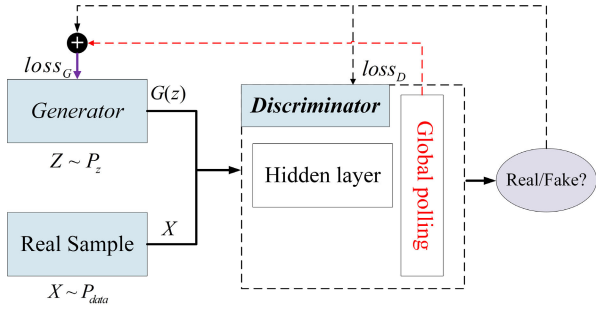


FIGURE 5. The loss transfer diagram in the training process of FCGAN.

computational formula is given as follows.

$$w^{(k)} = w^{(k)} - \eta \delta^{(k)} \frac{\partial \mathcal{F}^{(k)}(x)}{\partial x} x^{(k)} \quad (6)$$

where the superscript (k) denotes the k th layer of generator, w is the weight to be learned, x and δ represent the input and loss value separately, η is the learning rate which determines the descent stride per iteration, and $\mathcal{F}(x)$ denotes the activation function.

Note that the loss value $\delta^{(k)}$ in Eq. 6 is calculated by using $loss_G$ that is totally induced by the label-data of discriminator, which maybe lead to serval problems as discussed above. Thus, the penalty term which represents the loss value of similarity between real and fake samples is integrated into $loss_G$, as red dashed line shown in Fig. 5.

Finally, the gradient descent formula can be rewritten as follows.

$$w^{(k)} = w^{(k)} - \eta [\delta^{(k)} + \mathcal{E}] \frac{\partial \mathcal{F}^{(k)}(x)}{\partial x} x^{(k)} \quad (7)$$

It is obvious that the introduction of the penalty term is able to alleviate the issue of overdependence on label-data and lead the weight-learning to focus on the similarity between real and synthetic samples. The weight-learning is updated by using the above gradient descent algorithm until convergence.

2) FULLY CONVOLUTION

As discussed above, DCGAN [42] is likely to induce grid effect in synthetic images due to deconvolution operations with a large stride (stride > 1) in the generator network. Therefore, we design a fully convolutional network with a small stride (stride = 1) as the generator to relieve the grid issue, as shown in Fig. 4. In addition, an effective up-sampling method is conducted to ensure the feasibility of fully convolution. The specific design is summarized as follows.

- The up-sampling algorithm is employed to reshape the feature maps of each layer for the following convolution operation. Here we define the feature map as $\{\mathcal{H}(x, y) | x \in [1, w], y \in [1, h]\}$, then it is reshaped to $\{\mathcal{H}'(x, y) | x \in [1, \alpha * w], y \in [1, \beta * h]\}$, where w and h denotes the width and height of feature map, α and β are the scale factors along the horizontal and vertical direction separately.

For any coordinate (x_0, y_0) in $\mathcal{H}'(x, y)$, its corresponding coordinate mapped to $\mathcal{H}(x, y)$ can be located by the following formula:

$$\mathcal{H}'(x_0, y_0) \rightarrow \mathcal{H}(x_0/\alpha, y_0/\beta) = \mathcal{H}(i + u, j + v) \quad (8)$$

where i and u represent the integer and fractional parts of x_0/α separately. Similarly, j and v represent the integer and fractional parts of y_0/β .

Then, the value of each coordinate in $\mathcal{H}'(x, y)$ can be calculated according to the following interpolation algorithm:

$$\mathcal{H}'(x, y) = [1 - u \ u] \mathcal{Q} \begin{bmatrix} 1 - v \\ v \end{bmatrix} \quad (9)$$

where \mathcal{Q} is a matrix comprised of four coordinate points closest to $(x_0/\alpha, y_0/\beta)$, which is defined as follows.

$$\mathcal{Q} = \begin{bmatrix} \mathcal{H}(i, j) & \mathcal{H}(i, j + 1) \\ \mathcal{H}(i + 1, j) & \mathcal{H}(i + 1, j + 1) \end{bmatrix} \quad (10)$$

In the design of FCGAN, the scale factors α and β are both set to 2.

- As shown in Fig. 4, convolutional layers are used to deal with each output of up-sampling layer. Here the convolutional kernel and stride are both set to small size ($[3 \times 3]$ and 1), which is beneficial for decreasing the complexity of the generator and keeping details of feature maps (compared with large convolutional kernel and stride).

3) PRELIMINARY BATCH NORMALIZATION

The classical batch normalization is calculated by the following formula:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (11)$$

where

$$\mu_B = \frac{1}{n} \sum_{i=1}^n x_i, \quad \sigma_B^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_B)^2 \quad (12)$$

\mathcal{B} denotes a mini-batch of samples with size n , x_i is the output of neurons in the i th hidden layer, \hat{x}_i is the updated value of x_i using batch normalization, μ_B and σ_B are the mean value and standard deviation of \mathcal{B} in the i th hidden layer respectively, and ϵ is an extremely small constant.

From Eq. 11 and Eq. 12, it can be seen that the batch normalization makes the distribution of each layer's output tend to a normal distribution $\mathcal{N}(\mu_B, \sigma_B^2)$, which is capable of alleviating the vanishing gradient issue and improve the training process of networks effectively.

However, for a mini-batch of samples, it employs the same mean value and standard deviation (μ_B, σ_B^2), which is likely to make the generator network tend to synthesize samples with the same tone and degrade the diversity of synthetic samples. The specific examples and discussions will be elaborated in Section V.

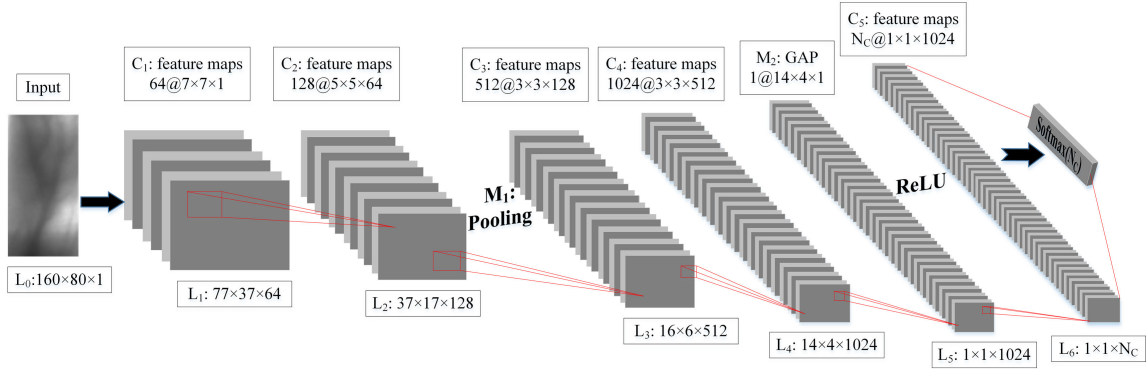


FIGURE 6. The architecture of our employed CNN for finger-vein image classification.

Here a preliminary batch normalization (PBN) is presented to solve the issue induced by batch normalization in the generator network, as shown in Fig. 4. The specific scheme of PBN is introduced in detail as follows.

- Before the formal training process begins, a pre-training is conducted using a batch of samples. Then the mean value and standard deviation of each hidden layer in the generator, which are defined as $\bar{\mu}$ and $\bar{\sigma}$ separately, could be statistically figured out according to Eq. 12. Note that $\bar{\mu}$ and $\bar{\sigma}$ will never be altered in the following formal training process.
- The obtained $\bar{\mu}$ and $\bar{\sigma}$ are taken as dynamic variables to adjust the distribution of each hidden layer's output in the formal training procedure. Thus, the Eq. 11 can be rewritten as

$$\hat{x}_i = \frac{x_i - (\mu_B + \lambda \bar{\mu})}{\sqrt{(\sigma_B + \lambda \bar{\sigma})^2 + \varepsilon}} \quad (13)$$

where λ is a real number ranged to $[-1, 1]$. Note that λ will be randomly initialized with regard to each mini-batch of training samples, which means that λ varies in the training process of each mini-batch samples.

Obviously, the distribution of each hidden layer's output is extended to the normal distribution $\mathcal{N}(\hat{\mu}_B, \hat{\sigma}_B^2)$, where $\hat{\mu}_B \in [\mu_B - \bar{\mu}, \mu_B + \bar{\mu}]$ and $\hat{\sigma}_B^2 \in [(\sigma_B - \bar{\sigma})^2, (\sigma_B + \bar{\sigma})^2]$. That is, the distribution of each hidden layer's output varies in a certain range through Eq. 13, which is beneficial for improving the diversity of synthetic samples by generator.

4) TRAINING PROCESS

The proposed FCGAN is trained to generate finger-vein ROIs for each category respectively and the training procedure is conducted alternately for the generator and discriminator. The mini-batch is set to 60 and weights are initialized by using a standardized normal distribution with a standard deviation of 0.01. We use stochastic gradient descent with the Adam optimizer [47] controlled by parameters $\beta_1 = 0.5$ and

$\beta_2 = 0.99$ separately, and adopt a learning rate of 0.0002 for 30 epochs.

IV. CNN FRAMEWORK

The proposed CNN architecture of finger-vein classification system is shown in Fig. 6. The CNN network consists of 5 convolutional layers, 2 pooling layers, 1 ReLU, and a soft-max layer. The elaboration of CNN topology and training procedure is introduced as follows.

Convolutional layers: In the first two convolutional layers, larger convolutional kernel size ($[7 \times 7 \times 1]$ and $[5 \times 5 \times 1]$) is used for acquiring a broader vision, which is capable of decreasing the false rejection rate caused by finger-vein image deformation or displacement. Besides that, larger stride is utilized to serve as the substitute of pooling layers, which is able to reduce the dimension of the input image and prevent overfitting. In the latter convolutional layers, small convolutional kernel size ($[3 \times 3 \times 1]$) is adopted to obtain more representative features, which could improve classification accuracy of the network.

Pooling layers: As depth of the network increases, the kernel number of every convolutional layer grows fast and corresponding parameters increase dramatically. To relieve the negative effect caused by overlarge parameters, a global pooling [48] with size $[14 \times 4 \times 1]$ is introduced in L_5 layer to immediately obtain the feature vector of 1024 dimensions. Then, the 1024-dimensional feature vector is fed into a classifier which is composed by a fully connected layer- L_6 . Finally, the network ends with a softmax layer to predict the probability that the input finger-vein image belongs to a specific category.

The detailed configuration of proposed CNN is shown in Table 1.

Training process: Input finger-vein images are totally cropped into ROIs (in Section II-B) and fixed into size 160×80 . In the training procedure, the mini-batch and learning rate are separately set to 64 and 0.0001 for 30 epochs. Furthermore, we adopt ReLU as the activation function and stochastic gradient descent optimized by using the Adam optimizer in which parameters β_1 and β_2 are set to 0.8 and

TABLE 1. The detailed parameters of employed CNN.

Layer	Type	Number of filters	Kernel size	Stride/Padding	Output Size
L_0	Input layer	-	-	-/-	$160 \times 80 \times 1$
C_1L_1	Convolutional layer	64	7×7	2/0	$77 \times 37 \times 64$
C_2L_2	Convolutional layer	128	5×5	2/0	$37 \times 17 \times 128$
M_1	Max-Pooling layer	1	2×2	2/0	$18 \times 8 \times 128$
C_3L_3	Convolutional layer	512	3×3	1/0	$16 \times 6 \times 512$
C_4L_4	Convolutional layer	1024	3×3	1/0	$14 \times 4 \times 1024$
M_2L_5	Global Average Pooling layer	1	14×4	1/0	$1 \times 1 \times 1024$
R_1	ReLU layer	-	-	-/-	-
C_5L_6	Convolutional layer	N_c (Number of classes)	1×1	1/0	$1 \times 1 \times N_c$
	Softmax layer	-	-	-/-	-

0.99 respectively. Besides that, to further lower the overfitting risks, during the training process we introduce the dropout technique [49] with a probability of 0.75 into the fully connected layer- L_6 .

V. EXPERIMENTS

In this section, a series of experiments are conducted to evaluate the performance of the proposed FCGAN on enhancing the quality of synthetic finger-vein images and improving the recognition results of the presented CNN architecture which is introduced in Section IV.

In addition, to evaluate the generalization ability of presented CNN well trained by FCGAN-augmented finger-images, a totally different database is taken for further verification. Note that in our experiments all the finger-vein images are preprocessed into ROIs utilizing the method elaborated in Section II-B.

The architectures of GANs and CNN employed in this paper are entirely implemented in tensorflow framework. All the experiments have been performed using i5, 3.20GHz CPU and an NVIDIA GeForce GTX 1080 Ti GPU. The detailed experiments and discussions are described as follows.

A. EVALUATION OF FCGAN AUGMENTATION

In this section, DCGAN [42] and SAGAN [44] are taken to compare with the proposed FCGAN on the following two aspects:

- Comparison on the quality of synthetic finger-vein images.
- Comparison on improving the recognition accuracy of finger-vein images.

Since the experiments of above two points depend on each other and more importantly, a progressive increase method is presented to evaluate the performance of various GAN-based methods on improving the recognition accuracy of finger-vein images. For clarity, the test on recognition accuracy is first conducted and in the last part a detail analysis and discussion about the quality of synthetic finger-vein images using different GAN-based methods is elaborated.

1) EVALUATION OF CLASSIFICATION PERFORMANCE

To exhibit the impact of FCGAN-augmentation on the classification performance of CNN more clearly, the presented CNN architecture is trained with a progressive increase way. It means that in every period samples are increased once, and a classification result can be obtained. Note that the same amount of samples is added into every finger-vein category in one period, which makes sure the balance of training samples among different finger-vein categories. The progressive increase method is elaborated in the following.

Classical sample augmentation: As mentioned in Section II-A, the HKPU dataset includes 210 categories of finger-vein images, which are captured from 105 subjects' index and middle fingers in two sessions. Taking into account the effect in different acquisition periods, we take finger-vein images of the first session as training samples and the second session as testing samples. We utilize classical sample augmentation techniques described in Section II-C to augment every finger-vein ROIs in each category. In this study, N_θ and N_δ are set to 8 and 10 respectively, which produces $N = 80$ augmented images for each finger-vein ROI and almost 500 images per category. We randomly divide the 500 images of each category into 10 mutually-exclusive groups equally and each group includes 50 images. In order to quantify easily, we define each group as $G_{class}^{(i)}$, where $i \in [1, 10]$.

GAN-based augmentation: We take the augmented images which are generated with classical sample augmentation above as the training data for generating synthetic images with GAN-based methods. Note that we utilize the GAN-based architecture to train each finger-vein category respectively.

Here DCGAN [42], SAGAN [44] and the proposed FCGAN are taken to produce synthetic images and demonstrate their impact on finger-vein classification accuracy using presented CNN, respectively. As implementing in classical sample augmentation, 500 synthesis images per category are generated by using GAN-based methods and randomly divide 500 images into 10 mutually-exclusive groups equally. Similarly, each group is denoted as $G_{gans}^{(j)}$,

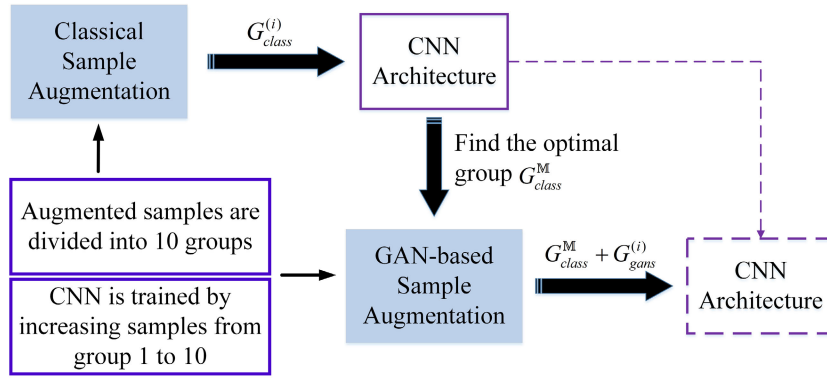


FIGURE 7. The Schematic diagram of training CNN with the progressive increase method.

Algorithm 1 Algorithm of Training the Employed CNN With a Progressive Increase Way

- 1: **Classical Augmentation:**
- 2: **for** each $i \in [1, 10]$ **do**
- 3: Take the data group $\{G_{class}^{(m)}, 1 \leq m \leq i\}$ per finger-vein category as training data for training the employed CNN \mathbb{M} ;
- 4: Update \mathbb{M} by stochastic gradient descent until training process is finished;
- 5: **end for**
- 6: Find out the optimal status $\mathbb{M}^{(o)}$ which gets the best classification performance while i equals to o , and the corresponding data group of each category $G_{class}^{\mathbb{M}} = \{G_{class}^{(m)}, 1 \leq m \leq o\}$.
- 7: **GAN-based Augmentation:**
- 8: **for** each $j \in [1, 10]$ **do**
- 9: Take the data group $\{G_{class}^{\mathbb{M}}, G_{gans}^{(n)}, 1 \leq n \leq j\}$ per finger-vein category into $\mathbb{M}^{(o)}$ for training;
- 10: Update $\mathbb{M}^{(o)}$ by stochastic gradient descent until training process is finished;
- 11: **end for**

where $j \in [1, 10]$. For simplicity, classification schemes using different augmentation methods are separately denominated as CLA-CNN, SAGAN-CNN, DCGAN-CNN and FCGAN-CNN.

Separately utilizing the classical and GAN-based sample augmentation methods, the training dataset is totally augmented to 2.1×10^5 images. In addition, the second session in HKPU database is taken as the testing dataset which includes 1260 images. Since the augmented samples of each category are randomly divided into 10 groups, so the training procedure is carried out using 10-fold cross-validation to evaluate our experimental performance.

Training CNN by a progressive increase method: The employed CNN is trained by two stages. Firstly, $G_{class}^{(i)}$ of each finger-vein category is put into CNN regularly from $i = 1$ to $i = 10$ for training, then the optimal point $\mathbb{M}^{(o)}$, achieving first-rank classification result, could be determined. Sec-

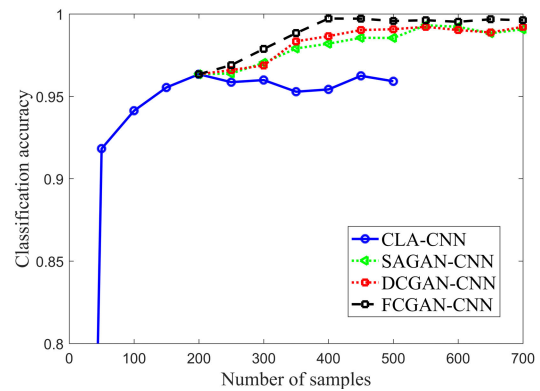


FIGURE 8. The classification accuracy for finger-vein classification with progressive increase method. The blue curve represents the classification accuracy trained by using classical augmented data which is increased from $G_{class}^{(1)}$ to $G_{class}^{(10)}$. The other three lines represent the classification accuracy trained by using GAN-based augmented data which is increased from $G_{gans}^{(1)}$ to $G_{gans}^{(10)}$.

ondly, $\mathbb{M}^{(o)}$ is trained sequentially, like the way described in stage 1, using GAN-based augmented sample $G_{gans}^{(j)}$. The detailed training scheme is elaborated in Algorithm 1 and Fig. 7.

2) EVALUATION RESULTS

Fig. 8 shows classification results using progressive increase method elaborated in Algorithm 1. From Fig. 8, it can be seen that while the number of finger-vein images in each category equals to 200 ($\{G_{class}^{(m)}, m \in [1, 4]\}$), the employed CNN reaches to optimal point $\mathbb{M}^{(o)}$ and the classification accuracy gets to 96.34%. After the optimal point $\mathbb{M}^{(o)}$, the network tends to be saturated and continuing to increase training data cannot improve the classification accuracy efficiently. However, when GAN-based augmented data is integrated into $\mathbb{M}^{(o)}$ increasingly, the classification accuracy has been improved greatly. The best classification performance of CNN using different GAN-based methods and corresponding sample size are summarized in Table 2.

TABLE 2. The best classification performance and corresponding sample size (each category).

Methods	Number of samples			Accuracy
	Classics	GANs	Totals	
SAGAN-CNN	200	350	550	98.71%
DCGAN-CNN	200	250	450	98.74%
FCGAN-CNN	200	200	400	99.67%

Algorithm 2 Algorithm for Computing the Coordinate Points (FPR, TPR) of ROC Curves

Require: τ -a threshold, different threshold corresponds to different coordinate point (FPR, TPR).

- 1: **for** each $i \in [1, 210]$ **do**
- 2: Treat images of i -th category as positive examples and images of other categories as negative examples;
- 3: Referring to binary classification, compare the predicted probability per image to the threshold τ and obtain the confusion matrices statistically;
- 4: Calculate $FPR^{(i)}$ and $TPR^{(i)}$ using confusion matrices;
- 5: **end for**
- 6: $FPR = \sum_{i=1}^{210} FPR^{(i)} / 210$, $TPR = \sum_{i=1}^{210} TPR^{(i)} / 210$.
- 7: **return** Coordinate point (FPR, TPR).

TABLE 3. The classification accuracy and EER of different methods.

Methods	TPR (FPR=0.1%)	TPR (FPR=1%)	EER
CLA-CNN	95.08%	98.11%	1.77%
SAGAN-CNN	97.31%	99.13%	0.93%
DCGAN-CNN	97.36%	99.17%	0.91%
FCGAN-CNN	98.77%	99.63%	0.52%

It is obvious that compared with SAGAN [44] and DCGAN [42], the proposed FCGAN achieves a better performance on improving the classification accuracy of CNN. In addition, the presented CNN converges faster while it is trained by utilizing FCGAN-augmented samples. It means that FCGAN is able to generate more high-quality and diverse finger-vein images, which is beneficial for improving the training procedure of CNN. The visualization of synthetic finger-vein images using different GAN-based methods will be shown at the end of this section.

In order to further demonstrate the reliability of integrating FCGAN-augmented data into training process, we validate accuracy of the three well-trained models (as the curves shown in Fig. 8) on samples of second session consisting of 1260 samples from 210 categories. We calculate false positive rate (FPR) and true positive rate (TPR) using Algorithm 2 to draw the receiver operating characteristic (ROC) curves for three models, as shown in Fig. 9. Table 3 gives the classification accuracy and equal error rate (EER) of the three

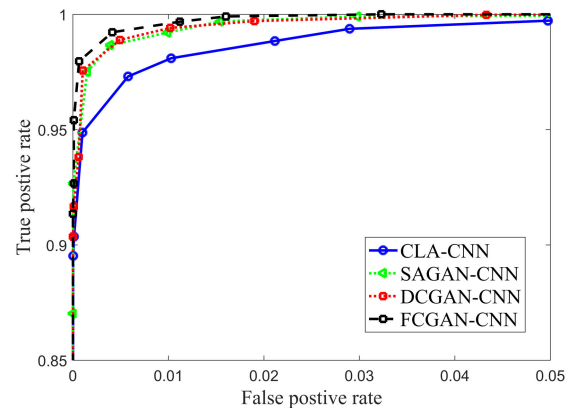


FIGURE 9. Receiver operating characteristic curves of different augmentation approaches.

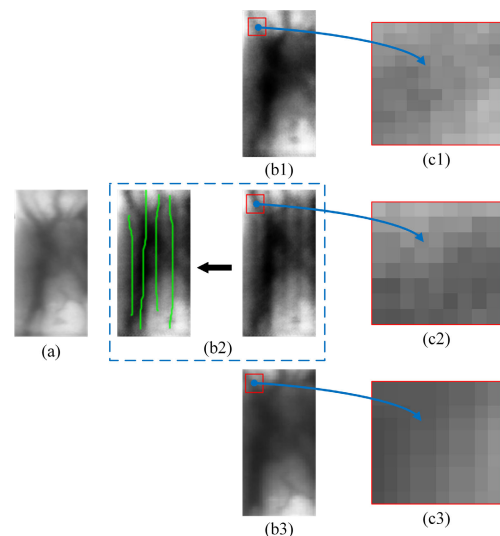


FIGURE 10. Examples of grid effect using different GAN-based methods. (a) Original finger-vein image, (b1)-(b3) Synthetic finger-vein images using SAGAN, DCGAN and FCGAN separately, (c1)-(c3) Grid effect of (b1)-(b3).

models on 1260 test samples, respectively. As can be seen from Fig. 9 and Table 3, the proposed FCGAN has maintained a highest classification accuracy and lowest EER, which are both superior to other augmentation schemes. Accordingly, it further illustrates that generating finger-vein images using FCGAN is indeed beneficial for improving finger-vein recognition accuracy and it is capable of producing finger-vein images which are more diverse and high-quality than other augmentation methods.

3) ANALYSIS ON IMAGE QUALITY

In this section, finger-vein images which are generated using different GAN-based methods are given to visually reveal the performance of generative models. Here the evaluation index is focused on the following two points that the inhibition of grid effect and the diversity of synthetic samples. The detailed discussion is given as follows.

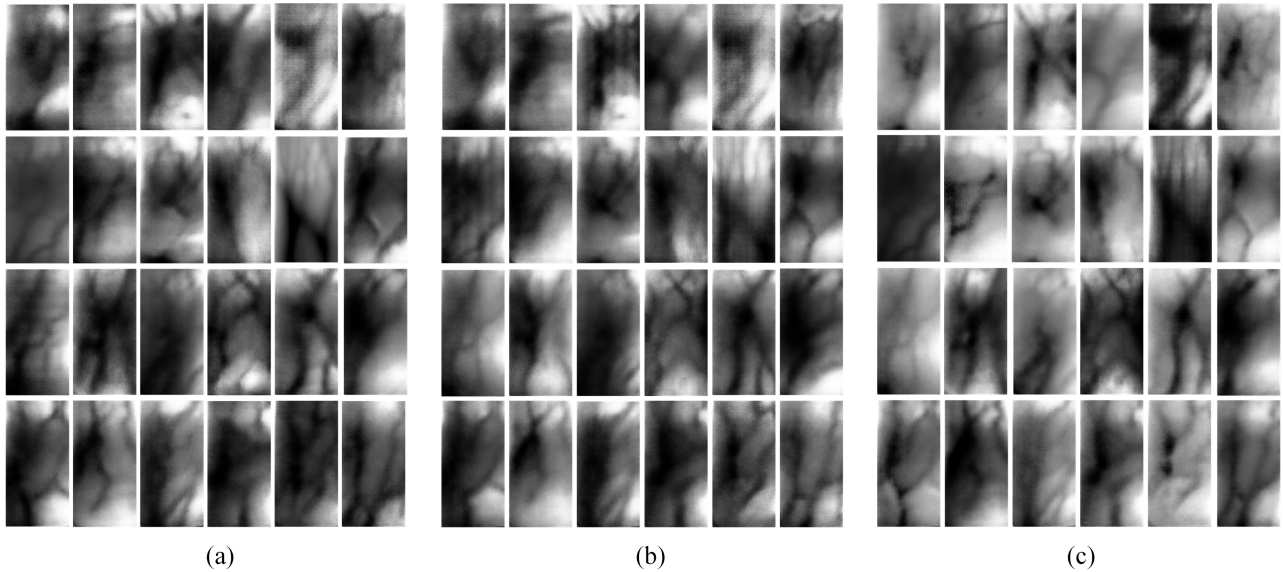


FIGURE 11. Finger-vein images generated by using various GAN-based methods. (a)-(c) Examples of synthetic finger-vein images using SAGAN, DCGAN and FCGAN respectively.

a: GRID EFFECT

Since DCGAN [42] excessively depends on the label-data to update weights and lose sight of the similarity between real and synthetic samples, unilateral dominance and unbalance maybe occurs in the training process. As shown in Fig. 10(b2), strip-shaped creases (areas marked by green lines) unexpectedly appear in the synthetic finger-vein image, which indicates that the generator surpasses and cheats the discriminator and finally dominates the training process to achieve convergence. The above situation could take side effect on the training of classification networks and degrade the classification accuracy. However, the strip-shaped creases do not appear in the finger-vein image generated by FCGAN, as shown in Fig. 10(b3). It uncovers that the penalty term integrating into the loss function is able to tighten the criterion of discriminator and lead the feature of synthetic samples closer to that of real samples.

In addition, due to the deconvolution operation, the finger-vein image generated by SAGAN [44] and DCGAN [42] shows obvious grid effect, as shown in Fig. 10(c1) and Fig. 10(c2). The grid effect is also a negative factor for the training of classification networks. By contrast, the finger-vein image generated by FCGAN is relatively smooth and the grid effect is well suppressed, as shown in Fig. 10(c3). It reveals that the proposed fully convolutional architecture used in FCGAN is able to effectively resist the grid effect induced by deconvolution operation and improve the quality of synthetic finger-vein images.

b: SAMPLE DIVERSITY

It is well known that the sample diversity is of great significance to improve the robustness of classification models. Therefore, it is also an important index to evaluate the generative networks. As shown in Fig. 11(b), the synthetic finger-vein images using DCGAN [42] tend to the

TABLE 4. The time consumption results of different methods during training procedure.

Methods	Computational time (s)		Total time (s)
	GANs	CNNs	
SAGAN-CNN	1673.65	688.78	2362.43
DCGAN-CNN	1068.78	688.78	1757.56
FCGAN-CNN	998.57	688.78	1687.35
FCGAN-AlexNet	998.57	983.41	1981.98

similar tone which shows dark visually. Although strip-shaped crease does not appear in synthetic finger-vein images using SAGAN [44], the generated images also tend to the similar tone, as shown in Fig. 11(a). As discussed above, it is attributed to the fact that batch normalization degrades the individual independence among generated samples, although it is able to accelerate the convergence of networks.

On the contrary, the synthetic finger-vein images generated by FCGAN show more variable tones, which reveals that the proposed PBN used in FCGAN is capable of improving the diversity of synthetic samples by dynamically changing the distribution of each hidden layer's output in the generator.

In summary, the proposed FCGAN is able to generate more diverse and high-quality finger-vein images, which simultaneously uncovers the reason why FCGAN-augmented finger-vein images are capable of improving the classification accuracy better.

4) ANALYSIS ON TIME CONSUMPTION

To fully demonstrate the advantages of the proposed scheme FCGAN-CNN in computational time during the training process, Table 4 shows the time consumption of different schemes which are all conducted with 1×10^5 training samples, 64 batches and 10 epochs.

TABLE 5. The encoding forms of various feature extraction methods.

Methods	Encoding type	Encoding length per minutiae	Total length of feature vector	Distance measure
Fei <i>et al.</i> [24]	decimal and binary code	76 dimensions	$76 \times N$ dimensions	Euclidean and Hamming distance
Matsuda <i>et al.</i> [25]	decimal code	45 dimensions	$45 \times N$ dimensions	Euclidean distance
Wang <i>et al.</i> [26]	binary code	-	128 dimensions	Hamming distance
CLA-CNN	decimal code	-	1024 dimensions	Euclidean distance
FCGAN-CNN	decimal code	-	1024 dimensions	Euclidean distance

Compared with other baseline methods, the proposed scheme FCGAN-CNN takes a lower time consumption, which is attributed to following aspects:

- Compared with SAGAN [44] which owns large network depth and parameter size, FCGAN belongs to the lightweight network.
- The penalty term is able to improve the balance between generator and discriminator and accelerate the training process of FCGAN.
- Both FCGAN and CNN used in this paper adopt the global pooling layer to replace the dense layer, which is capable of decreasing the parameter size greatly and reducing the risk of overfitting.

B. EVALUATION OF GENERALIZATION ABILITY

In this section, we adopt a totally different dataset SDUMLA to evaluate the generalization ability of FCGAN-CNN, and reveal the prominent ability of FCGAN-CNN to extract effective features by comparing with different existing minutiae-based methods. The detailed validation scheme and experimental results are introduced as follows.

1) TEST DATASET AND SCHEME

The test dataset used in this experiment is acquired from SDUMLA finger-vein image database, which is introduced briefly as follows:

The SDUMLA finger-vein image database [50] consists of 636 fingers captured from 106 subjects, which were collected by Shandong University of China. Each of the subjects provided 6 image samples from the index, middle and ring fingers of left and right hands. All the finger-vein images are stored as BMP format with a fix size of 320×240 pixels.

In our experiment, the finger-vein images are totally normalized to fixed size of 160×80 using the preprocess procedure described in Section II-B.

We obtain feature vectors from finger-vein images of SDUMLA database using FCGAN-CNN, and evaluate the performance of networks utilizing One-to-many matching approach. The detailed scheme is given as follows.

a: EXTRACTING FEATURE VECTOR

Firstly, we take each finger-vein image of SDUMLA database into the well-trained FCGAN-CNN. Then, through the

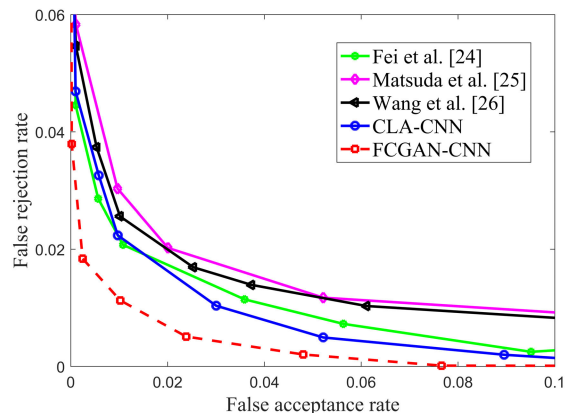


FIGURE 12. ROC curves of various feature extraction methods.

output of L_5 layer we could obtain a feature vector with 1024 dimensions $V^{(i)} = [p_1^{(i)}, p_2^{(i)}, \dots, p_{1024}^{(i)}]^T$ corresponding to each finger-vein image $I^{(i)}$.

b: ONE-TO-MANY MATCHING

We utilize the feature vector $V^{(i)}$ to match 636 finger-vein images (one-to-many comparison) with six images stored per finger. Therefore, the matching number of same category is 19080 ($636 \times C_6^1 \times C_5^1$), and the matching number of different categories is 384780 ($636 \times (636 - 1) - 636 \times C_6^1 \times C_5^1$).

2) TEST RESULTS

To further demonstrate the performance of FCGAN-CNN to extract effective features, various latest minutiae-based methods are taken in this experiment for comparison. The encoding forms of different methods are elaborated in Table 5, where N is the number of minutiae. The feature vector extracted by Fei *et al.* [24] includes two parts, the spatial information encoded in decimal and the LEBP descriptor encoded in binary. Hence, Fei *et al.* utilized Euclidean and Hamming distance to evaluate the similarities of feature vector, respectively. In addition, Wang *et al.* [26] divided a finger-vein image into 96 patches and each patch encoded as a 19-dimensional decimal vector, then the whole 1824-dimensional decimal feature was compacted and encoded as a 128-dimensional binary feature. Based on the one-to-many matching scheme above, we calculate the false acceptance rate (FAR) and the false rejection rate (FRR),

TABLE 6. The classification accuracy and EER of three different methods (GAR=1-FRR).

Methods	GAR (FAR=0.1%)	GAR (FAR=1%)	EER
Fei et al. [24]	95.54%	97.83%	1.93%
Matsuda et al. [25]	93.87%	96.98%	2.14%
Wang et al. [26]	94.54%	97.44%	2.03%
CLA-CNN	95.31%	97.76%	1.92%
FCGAN-CNN	97.13%	99.15%	0.87%

and by adjusting the recognition threshold we draw the ROC curves of various approaches, as shown in Fig. 12. Table 6 illustrates the EERs, and genuine acceptance rates (GARs) of existing methods under different FARs.

The experimental results show that compared with latest feature extraction methods the FCGAN-CNN reaches a higher GAR and lower EER, which reveals that using FCGAN to generate finger-vein images for sample augmentation is beneficial for improving the ability of networks to extract deep features. Moreover, the FCGAN-CNN gets better results than CLA-CNN, which further confirms that integrating GAN-based sample augmentation into the training procedure could break through the limitation of classical augmentation in improving the classification result of networks.

VI. CONCLUSION

The existing finger-vein identification systems, especially deep learning based methods, are mainly implemented on small-scale dataset. It is hard to ensure a high tolerance to the deformation caused by finger position, or the stable recognition accuracy on different finger-vein dataset. Hence, this paper proposes a lightweight GAN-based architecture named FCGAN, using fully convolutional networks and tightly-constrained loss function for implementing finger-vein image augmentation. In addition, we present a novel scheme FCGAN-CNN for finger-vein classification, which reveals that synthetic finger-vein images using FCGAN are capable of improving the property of CNN for finger-vein image classification.

Firstly, to weaken the side impact of finger position on classification accuracy, a robust gradient-based approach is applied to detect phalangeal joints accurately and achieve well-corrected finger-vein ROIs. Furthermore, large-stride convolution layers are applied in our employed CNN to acquire a broader vision and further decrease the false rejection rate caused by finger-vein position.

Secondly, the proposed FCGAN is designed as a fully convolutional network with a small stride to relieve the grid effect caused by the deconvolution operation. In addition, a novel preliminary batch normalization is employed in FCGAN to improve the diversity of synthetic samples. To further improve the quality of synthetic samples and relieve the unilateral dominance and unbalance during the training

procedure of GANs, the tightly-constrained loss function with a penalty term is employed in FCGAN.

Finally, we validate the effectiveness of our proposed FCGAN-CNN scheme through two groups of experiments. In the first experiment, we verify that sample augmentation using FCGAN is able to generate more diverse and high-quality finger-vein images than existing augmentation methods. In addition, it can break through the limitation of classical augmentation and achieve a better performance on improving the classification accuracy of employed CNN. In the second experiment, we evaluate the generalization ability of FCGAN-CNN scheme on a totally different dataset and reveal its reliability on new datasets.

VII. FUTURE WORK

As the proposed system has not yet been perfected, there are still many challenges in our future work.

A. NETWORK VISUALIZATION

The visualization of neural networks is an important research field and we can obtain output results of each hidden layer visually using this technology. It is worthwhile to investigate related approaches of implementing the visualization of neural networks and make the experimental results more clear and convincing.

B. TRAINING COMPLEXITY

The training procedure of FCGAN is complicated and now we train separate FCGAN for each finger-vein category, which increases the training complexity greatly. Hence, it is worthwhile to investigate GAN architectures and find out the solution to generate multi-class samples together.

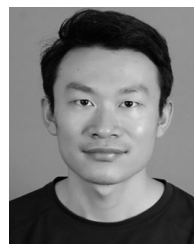
C. EXPANSION PLAN

The two groups of experiments validate that our proposed FCGAN-CNN scheme could effectively obtain the deep features from a finger-vein image. Therefore, the proposed scheme could be also extended to other biometric domains, such as finger print and palm vein, which can benefit from synthetic images augmented with GAN-based methods for improving the training procedure.

REFERENCES

- [1] S. Marcel, M. S. Nixon, and S. Z. Li, *Handbook of Biometric Anti-Spoofing*, vol. 1. Cham, Switzerland: Springer, 2014.
- [2] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcão, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 864–879, Apr. 2015.
- [3] Y. Zhou and A. Kumar, "Human identification using palm-vein images," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1259–1274, Dec. 2011.
- [4] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Mach. Vis. Appl.*, vol. 15, no. 4, pp. 194–203, 2004.
- [5] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li, "Finger-vein authentication based on wide line detector and pattern normalization," in *Proc. Int. Conf. Pattern Recognit.*, 2010, pp. 1269–1272.
- [6] L. Lei, F. Xi, and S. Chen, "Finger-vein image enhancement based on pulse coupled neural network," *IEEE Access*, vol. 7, pp. 57226–57237, 2019.

- [7] J. Yang and Y. Shi, "Towards finger-vein image restoration and enhancement for finger-vein recognition," *Inf. Sci.*, vol. 268, no. 6, pp. 33–52, Jun. 2014.
- [8] L. Yang, G. Yang, X. Xi, X. Meng, C. Zhang, and Y. Yin, "Tri-branch vein structure assisted finger vein recognition," *IEEE Access*, vol. 5, pp. 21020–21028, 2017.
- [9] S. Shazeeda and B. A. Rosdi, "Nearest centroid neighbor based sparse representation classification for finger vein recognition," *IEEE Access*, vol. 7, pp. 5874–5885, 2019.
- [10] X. Meng, X. Xi, G. Yang, and Y. Yin, "Finger vein recognition based on deformation information," *Sci. China Inf. Sci.*, vol. 61, no. 5, p. 052103, 2018.
- [11] L. Yang, G. Yang, K. Wang, H. Liu, X. Xi, and Y. Yin, "Point grouping method for finger vein recognition," *IEEE Access*, vol. 7, pp. 28185–28195, 2019.
- [12] J. Yang, J. Yang, and Y. Shi, "Finger-vein segmentation based on multi-channel even-symmetric Gabor filters," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst. (ICIS)*, vol. 4, Nov. 2009, pp. 500–503.
- [13] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012.
- [14] Z. Lu, S. Ding, and J. Yin, "Finger vein recognition based on finger crease location," *J. Electron. Imag.*, vol. 25, no. 4, p. 043004, 2016.
- [15] T. Liu, J. B. Xie, W. Yan, P. Q. Li, and H. Z. Lu, "An algorithm for finger-vein segmentation based on modified repeated line tracking," *J. Photograph. Sci.*, vol. 61, no. 6, pp. 491–502, 2013.
- [16] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Trans. Inf. Syst.*, vol. 90, no. 8, pp. 1185–1194, 2007.
- [17] W. Song, T. Kim, H. Kim, J. Choi, H. Kong, and S. Lee, "A finger-vein verification system using mean curvature," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1541–1547, 2011.
- [18] H. Qin, X. He, X. Yao, and H. Li, "Finger-vein verification based on the curvature in Radon space," *Expert Syst. Appl.*, vol. 82, pp. 151–161, Oct. 2017.
- [19] J. Yang, B. Zhang, and Y. Shi, "Scattering removal for finger-vein image restoration," *Sensors*, vol. 12, no. 3, pp. 3627–3640, 2012.
- [20] E. C. Lee, H. Jung, and D. Kim, "New finger biometric method using near infrared imaging," *Sensors*, vol. 11, no. 3, pp. 2319–2333, 2011.
- [21] B. A. Rosdi, W. S. Chai, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, pp. 11357–11371, 2011.
- [22] Y. Lu, S. Yoon, S. Wu, and D. S. Park, "Pyramid histogram of double competitive pattern for finger vein recognition," *IEEE Access*, vol. 6, pp. 56445–56456, 2018.
- [23] H. Liu, L. Yang, G. Yang, and Y. Yin, "Discriminative binary descriptor for finger vein recognition," *IEEE Access*, vol. 6, pp. 5795–5804, 2018.
- [24] F. Liu, G. Yang, Y. Yin, and S. Wang, "Singular value decomposition based minutiae matching method for finger vein recognition," *Neurocomputing*, vol. 145, no. 145, pp. 75–89, Dec. 2014.
- [25] Y. Matsuda, N. Miura, A. Nagasaka, H. Kiyomizu, and T. Miyatake, "Finger-vein authentication based on deformation-tolerant feature-point matching," *Mach. Vis. Appl.*, vol. 27, no. 2, pp. 237–250, 2016.
- [26] K. Wang, L. Yang, G. Yang, and Y. Yin, "Integration of discriminative features and similarity-preserving encoding for finger vein image retrieval," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3525–3529.
- [27] J. Peng, N. Wang, A. A. A. El-Latif, Q. Li, and X. Niu, "Finger-vein verification using Gabor filter and sift feature matching," in *Proc. 8th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, 2012, pp. 45–48.
- [28] H. Qin, L. Qin, L. Xue, X. He, C. Yu, and X. Liang, "Finger-vein verification based on multi-features fusion," *Sensors*, vol. 13, no. 11, pp. 15048–15067, 2013.
- [29] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017.
- [30] Y. Fang, Q. Wu, and W. Kang, "A novel finger vein verification system based on two-stream convolutional network learning," *Neurocomputing*, vol. 290, pp. 100–107, May 2018.
- [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [32] Y. Lu, S. Xie, and S. Wu, "Exploring competitive features using deep convolutional neural network for finger vein recognition," *IEEE Access*, vol. 7, pp. 35113–35123, 2019.
- [33] J. M. Song, W. Kim, and K. R. Park, "Finger-vein recognition based on deep densenet using composite image," *IEEE Access*, vol. 7, pp. 66845–66863, 2019.
- [34] R. Das, E. Piciucco, E. Maiorana, and P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 360–373, Feb. 2019.
- [35] J. Zhang, Z. Lu, and M. Li, "Finger-vein recognition based on gradient distribution and self-adaptive recovery model," *J. Electron. Imag.*, vol. 27, no. 5, p. 053022, 2018.
- [36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [37] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1170–1181, May 2016.
- [38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [39] G. Wang, W. Kang, Q. Wu, Z. Wang, and J. Gao, "Generative adversarial network (GAN) based data augmentation for palmprint recognition," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, 2018, pp. 1–7.
- [40] B. Kwolek, "GAN-based data augmentation for visual finger spelling recognition," in *Proc. 11th Int. Conf. Mach. Vis. (ICMV)*, vol. 11041, 2019, p. 110411U.
- [41] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer GAN to bridge domain gap for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 79–88.
- [42] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: <https://arxiv.org/abs/1511.06434>
- [43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [44] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*. [Online]. Available: <https://arxiv.org/abs/1805.08318>
- [45] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.
- [46] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
- [47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [48] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: <https://arxiv.org/abs/1312.4400>
- [49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [50] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," in *Proc. Chin. Conf. Biometric Recognit.* Berlin, Germany: Springer, 2011, pp. 260–268.



JIANFENG ZHANG received the M.Sc. degree in automation from the Nanjing University of Science and Technology, Nanjing, China, in 2010. He is currently pursuing the Ph.D. degree with Tianjin University, Tianjin, China. From 2010 to 2015, he was employed in the Shandong Academy of Sciences and committed to the research of information security and machine learning. His research interests include biometrics recognition and deep learning.



ZHIYING LU received the B.Sc. and M.Sc. degrees from the School of Electrical and Information Engineering, Tianjin University, Tianjin, China, in 1987 and 1992, respectively. She is currently a Professor with the School of Electrical and Information Engineering, Tianjin University. She has over 20 years of academic research and has presided over a number of national and provincial projects. Her research interests include image processing, biometrics recognition, and power systems.



HAOPENG WU received the M.Sc. degree from the Inner Mongolia University of Technology, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Electrical and Information Engineering, Tianjin University, China. His research interests include face recognition and facial expression recognition.

• • •



MIN LI received the B.Sc. degree from Shandong University, China, in 2015. She is currently pursuing the degree with the School of Electrical and Information Engineering, Tianjin University, Tianjin, China. Her research interests include information security and image processing.