Adversarial learning-based data augmentation for palm-vein identification

Huafeng Qin, Haofei Xi, Yantao Li, Mounim El-Yacoubi, Jun Wang, Xinbo Gao

Adversarial Learning-based Data Augmentation for Palm-vein Identification

Huafeng Qin, Haofei Xi, Yantao Li, Mounim A. El-Yacoubi, Jun Wang and Xinbo Gao

***Abstract*—Palm-vein identification is a highly secure pattern biometrics that has become an active research area in recent years. Despite the recent progress in deep neural networks (DNNs) for vein identification, existing solutions for feature representation continue to lack robustness due to the limited training samples. To address this limitation, data augmentation approaches, including Generative Adversarial Networks (GANs), have been investigated, but these schemes suffer from the following issues. First, it is practically unfeasible to use all the generated samples for classifier training due to the limited storage space and computation resources. Further, some of these generated samples may be non-representative or ineffective, seriously compromising models' generalization capabilities. Second, the augmented dataset is fed to the target classifier repeatedly, resulting in overfitting after substantial training epochs. To tackle the above problems, we propose Advein$\mathbb{AU}$, an Adversarial vein AUtomatic AUgmentation approach that generates challenging samples to train a more robust vein classifier for palm-vein identification by alternatively optimizing the vein classifier and a set of latent variables. First, we consider a conditional deep convolution generative adversarial net (cDCGAN) to learn the distribution of real data and the generated data, and then a latent variable from the latent variable space is mapped to the sample space. Second, we combine the trained generator with the vein classifier to constitute Advein$\mathbb{AU}$, where the input sets of the generator and the classifier are alternatively updated by adversarial training. Specifically, a latent variable set is learned to increase the training loss of a target network through generating adversarial samples, while the classifier learns more robust features from harder examples to improve the generalization. To avoid collapsing inherent meanings of images, an exponential moving average (EMA) teacher and $cosine$ similarity are employed for regularization to reduce the search space. Unlike previous works where GANs synthesize new realistic images, our model aims to search a latent variable set, based on which the generator can produce challenging samples along with the training process to improve the classifier's performance. Finally, we conduct extensive experiments on three public palm-vein datasets to evaluate the performance of Advein$\mathbb{AU}$, and the experimental results demonstrate that the proposed Advein$\mathbb{AU}$ is capable of generating harder samples to improve the performance of the vein classifier.**

***Index Terms*—Data augmentation, Adversarial learning, Palm-vein identification, Deep Learning, Conditional generative adversarial networks**

I. Introduction

With the large spectrum of Internet applications and increasing privacy awareness concerns, translational authentication technology cannot meet the requirements of users with respect to security and convenience. For instance, passwords are prone to forgetfulness, keys are easy to copy and forge, and ID cards are vulnerable to physical damage and destruction. To overcome these issues, biometrics-based recognition technology that automatically identifies individuals by their physiological characteristics (e.g., face [1], fingerprint [2], iris [3], and vein [4]) or behavioral characteristics (e.g., gait [5] and voice [6]) has been investigated in recent decades, and some commercial biometrical recognition systems have been developed and applied for immigration clearance, financial payments, and access control. Commonly used biometrical traits include face, fingerprint, and iris. Applications of such traits, however, result in security and privacy issues. For example, it is not difficult to obtain a human face copy version without user awareness, and this has been successfully employed to fool commercial face recognition systems, even with 3D face recognition [7]. Likewise, fingerprint samples are easy to copy from the fingerprint residues on a sensor. By contrast, the vein traits are hidden beneath the human skin, which results in the following advantages [8]. 1) High stability: Vein vessels are located beneath the human finger skin, and the authentication results are, therefore, not prone to degradation owing to skin humidity. Generally, they are not easily damaged and contaminated by external factors. 2) High security and privacy: As veins are difficult to observe in visible light, it is much harder to obtain vein patterns without user consent. Hence, it is much more difficult to forge vein patterns to attack systems based on vein imaging. 3) Liveness detection: the human skin includes outermost epidermis, dermis, and the subcutaneous layer. The subcutaneous layers contain subcutaneous veins and arteries [9]. Different skin layers have different responses to light with different wavelengths. The

H. Qin, and H. Xi are with the Chongqing Key Laboratory of Intelligence Perception and Blockchain technology, Chongqing Engineering Laboratory of Detection Control and Integrated System, School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China (e-mail: qinhuafengfeng@163.com).

H. Qin and X. Gao are with the Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: qinhuafengfeng@163.com, gaoxb@cqupt.edu.cn).

Y. Li is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: yantaoli@cqu.edu.cn).

M. A. El-Yacoubi is with SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France (e-mail: mounim.el_yacoubi@telecom-sudparis.eu).

J. Wang is with the College of Computer Science, China University of Mining and Technology, Jiangsu 221116, China (e-mail: WJ999LX@163.com).

Manuscript received April XX, 2022; revised July XX, 202X. This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61976030, 62072061, and U20A20176), the Scientific Innovation 2030 Major Project for New Generation of AI (Grant No. 2020AAA0107300), the Science Fund for Creative Research Groups of the Chongqing University (Grant No. CXQT21034), the Chongqing Talent Program (Grant No. CQYC201903246), the Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJQN201900848), and the National Key R&D Program of China (Grant Nos. 2022YFC3801700 and 2020YFB1805400). (Corresponding author: Yantao Li.)

blood that returns to the heart through the veins contains deoxygenated hemoglobin, which absorbs light in the near-infrared part of the spectrum so that the vein patterns can be recorded based on a near-infrared camera. As deoxidized hemoglobin only exists in the lively body, the vein pattern is a biometrics modality with inherent liveness detection [10], [11]. Thanks to these advantages, research on vein recognition has attracted increasing attention in recent years.

A. Related work

The blood vessels are concealed inside our body and show good connectivity. Generally, the vein patterns are difficult to observe in visible light, but they can be collected by infrared light with a wavelength of about 850 nm. Vein patterns have been shown to have a unique structure for each individual in some medical research works [12], [13]. Vein recognition, however, is still challenging because collected vein images may be affected by many factors, such as light scattering [14], [15], environment temperature [13], [16], and user behavior [13], [16], [17]. As a result, the capturing process may generate some low-quality images where the vein region and the background region are hard to distinguish, which may degrade recognition accuracy. To tackle this problem, various approaches have been proposed to extract the vein patterns. They can be broadly split into three categories.

1) Handcrafted approaches: Handcrafted approaches are usually mathematics-based models that are designed to overcome specific issues by human $prior$ knowledge [18]. Handcrafted approaches typically include valley detection-based approaches [9], [16], [17], [19]–[22], line-like detection-based approaches [13], [23]–[25], and local descriptor-based approaches [26]–[29]. Given that the cross-sectional profile created by vein pixels shows a valley shape, some researchers employ descriptors, such as the curvature and Gaussian function, to detect such a valley for vein feature extraction. For example, Miura et al. [17] proposed repeated line tracking to detect vein pixels located in the valley. Subsequently, they investigated an approach to compute the local maximum curvatures in the cross-sectional profiles of a vein image [16]. For the same purpose, vein patterns are extracted by computing the difference of curvature [30] and the mean of curvature [19] in the valley. Recently, Yang et al. [31] proposed an orientation map-guided curvature-based approach to extract the vein pattern. Others have assumed that the vein patterns appear like a line segment in a predefined region and have proposed, accordingly, some models to extract the line-like features. The Gabor filter, a classical detector, has also been applied for vein extraction [13], [25]. Moreover, its improved versions [32], [33] were developed to enhance the vein patterns. Besides, local descriptors have been explored to extract invariant features for vein recognition. For instance, LBP [26]–[28] has been employed for feature extraction from the raw image, and some researchers have proposed various improved approaches, such as local line binary pattern (LLBP) [27], [34], efficient local binary pattern (ELBP) [29], and discriminative binary code (DBC) [35]. It is also worth mentioning that the improved watershed transformation proposed by Lin et al. [36] extracts the points of the vein patterns (FPVPs).

2) Traditional machine learning-based approaches: Most traditional machine learning methods deal with the data in shallow structures [37]. These structural models have only one or two layers of nonlinear feature transformation at most. Different from handcrafted descriptors in the first category, traditional machine learning is to learn laws from historical data through related algorithms and make predictions or judgments on new sample data. For example, the work [38] employed principal component analysis (PCA), kernel principal component analysis (KPCA) [39], and kernel entropy components for vein recognition [40]. Similarly, Yang et al. [41] combined $(2D)^2PCA$ with metric learning to extract compact features. To improve verification performance, linear discrimination analysis (LDA) [42] was employed to extract the discriminating features, which then were input to the nearest neighbor classifier. To extract more reliable and accurate vein patterns, Kapoor et al. [43] proposed a grey wolf optimization-based SVM (GWO-SVM) to optimize SVM parameters. In [44], the vein patterns are firstly enhanced, and then a sparse representation (SR) was introduced to learn a robust feature representation from the enhanced images. Besides, the work [45] incorporated a decision rule into SR to improve vein classification performance. Recently, Yang et al. [46] employed some extra information, such as class labels and local geometric structure, as the constraints of low-rank coefficients to propose a low-rank representation-based model for extracting noiseless discriminative information from vein images.

3) Deep learning-based approaches: Different from traditional shallow learning, deep learning emphasizes the depth of the model structure and the importance of feature learning, which can automatically learn features from the data in an end-to-end way [37]. In recent years, deep neural networks, such as CNN and Transformer, have shown robust feature representation capacity and achieved state-of-the-art performance in computer vision tasks. Inspired by their success, many researchers brought them into vein classification tasks [47]–[66]. For example, CNN has been employed for vein identification [47], [48], quality assessment [50], vein segmentation [4], template protection [67], and AntiSpoofing attacks [55]. Besides, generative adversarial networks (GANs) are proposed to learn the joint distribution of finger-vein images and pattern maps for vein texture extraction [52]. Motivated by the success of vision Transformers for image classification, the transformer is explored for vein recognition tasks. Unlike CNN, which aims to learn local features, the transformer is capable of learning the global dependencies among the tokens from an image. For example, Lu et al. [64] proposed a vision Transformer (ViT) to extract vein features. To achieve a robust classification, four modules, namely conditional positional, embedding expansion token embedding, expansion-less mechanism, and local information-enhanced FFN [66], were incorporated into ViT for vein feature representation. To learn robust features for multi-view finger-vein recognition, Zhao et al. [68] investigated a deep neural network named Hierarchical Content-Aware Network (HCAN) by combining a CNN and a

recurrent neural network (RNN).

B. Motivation

The handcrafted approaches extract the vein patterns based on some assumptions that the distributions of vein patterns show a valley or a line-like shape. However, the vein pixels may generate more complex distributions, compromising recognition performance. In addition, the handcrafted-based approaches usually extract some image processing-based features that might discard relevant information about vein recognition. In other words, the vein features extracted from handcrafted approaches may be incomplete for recognition. On the contrary, traditional learning based approaches such as SVM and LDA automatically learn robust features for the representation of an image. Nevertheless, the recognition performance is still not robust because of their limited feature representation capacity. Different from these approaches in the first two categories, deep learning-based approaches are able to automatically learn robust feature representations from the raw images without any $prior$ assumptions by inferring rich information from large training data. However, as described in the conclusion of work [59], some classical convolutional neural networks (CNNs) achieve lower recognition accuracy than some traditional methods on challenging datasets. One of the important reasons is that they generally require large training sample sizes to train a large number of network parameters. Unfortunately, it is hard to acquire a large number of samples from each class in practical applications because of limited storage and privacy policy. Generally, there are no more than twenty images for each class in existing 2D vein datasets [9], [13], [16], [19], [23], [30]. Therefore, the learning capacity of these models is not sufficiently exploited due to the very limited training samples. To solve this problem, some researchers introduced data augmentation techniques, such as GAN, to augment the set of training samples, thereby improving the performance of deep learning-based approaches. For example, the GAN has been employed to enlarge training image sets for palm-vein recognition [8], [69], palmprint recognition [70], and finger-vein recognition [71], [72]. These existing GAN models for data augmentation share the same learning paradigm: a GAN to learn the real data distribution, an augmented training dataset based on additional synthetic images generated from GAN, and a deep learning-based classifier for vein classification. As a result, they suffer from the following problems. In these works [8], [69]–[72], after training robust GANs, a certain amount of synthesized samples are randomly generated for the target network training. However, the image generation scheme is not directly related to the target task, i.e., classification. Moreover, it is impossible to generate all the possible instances from the learning distribution space because of limited storage space and computation resources. Consequently, the works [8], [69]–[72] employ the trained generator to generate synthesized samples for data augmentation randomly. However, the resulting synthesized samples may not be closely relevant for the classification task, which degrades classifier generalization. Besides, these generated samples will be input to the target network repeatedly, which results in an inevitable overfitting in a long multi-epoch training.

C. Our work

To address the drawbacks mentioned above, we propose Advein$\mathbb{AU}$, an Adversarial vein AUtomatic AUgmentation approach to train a robust classifier for vein identification by simultaneously optimizing vein classifier training and a latent variable set search, as shown in Fig. 1. First, we investigate a cDCGAN consisting of a generator and a discriminator to learn the real data distribution for data augmentation. To improve the performance, we employ two time-scale update rules (TTUR) and a one-sided label smoothing for training cDCGANs with stochastic gradient descent. Moreover, the Wasserstein generative adversarial networks with Gradient Penalty (WGAN-GP)) loss is used to improve stability. After training, taking different latent variables and conditional information as the inputs of the generator, the latter generates the samples for data augmentation. Second, we combine the trained generator with a vein classifier to obtain an adversarial network, consisting of two adversarial players. One is a vein classifier that predicts the probability of input samples belonging to a class. The other is a collection of latent variables that are taken as the input of the trained generator to create adversarial samples, thereby increasing the classifier loss. The two players are alternately updated during the training process. We conduct rigorous experiments on three public palm-vein datasets, and the experimental results show that the proposed Advein$\mathbb{AU}$ can improve the identification accuracy of the existing vein classifiers and outperform the existing GAN-based vein augmentation approaches.

The main contributions of this work are summarized as follows:

1) We present an adversarial learning-based data augmentation approach, Advein$\mathbb{AU}$, to control the quality of vein images generated by GAN models, thereby improving the performance of vein classifiers.

2) To perform palm-vein data augmentation, we explore a cDCGAN to learn the real sample distribution. Then, an adversarial framework is proposed to jointly optimize the target classifier training and the latent variable search.

3) We carry out extensive experiments to estimate the performance of Advein$\mathbb{AU}$ on three public palm-vein datasets. The experimental results show that the proposed Advein$\mathbb{AU}$ improves the feature representation capacity of state-of-the-art classifiers and outperforms GAN-based data augmentation approaches. In addition, we conduct ablation experiments to further verify Advein$\mathbb{AU}$, and the results show that the $cosine$ distance and teacher model enhance the performance.

The rest of this work is organized as follows: We detail the proposed approach in Section II and conduct experiments to evaluate the performance of our approach in Section III. Section IV concludes this work.

II. The proposed approach

Currently, many approaches [31], [36], [50], [68] have been proposed for vein recognition. Compared to existing works, we intend to propose an adversarial learning model for palm-vein identification, which exhibits key differences: 1) Unlike these works proposing deep learning-based classifiers or handcrafted-based descriptors to improve recognition

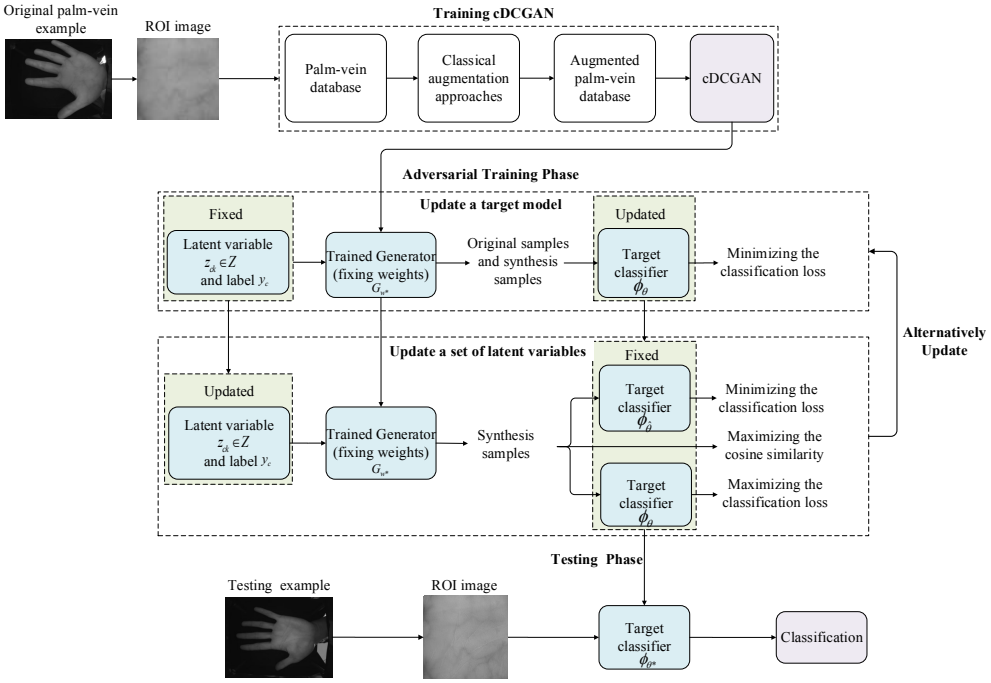

Fig. 1: Architecture of Advein$\mathbb{AU}$ palm-vein identification system

performance, our goal is to solve the problem of scarce vein data. Specifically, we focus on generating harder vein image samples to challenge any vein classifier to improve recognition accuracy. 2) Instead of optimizing a single classification model, we develop an adversarial framework to optimize the target classifier training and latent variable search jointly. After iterative training, a more robust target classifier is obtained for vein classification. Different from traditional GAN-based vein data augmentation models, we aim to build a set of latent variables that can continuously track the change of the learned representation rather than synthesized images. In this section, we present our adversarial learning-based data augmentation approach, Advein$\mathbb{AU}$, as illustrated in Fig. 1. As shown in Fig. 1, for the training phase, classic augmentation techniques are introduced to enlarge the training dataset, based on which the cDCGAN is trained to learn the sample distribution. Then, the generator of the trained cDCGAN is combined with the target classifier to obtain the Advein$\mathbb{AU}$ network, which is trained by alternatively updating the target classifier and the set of latent variables. During the training process, the set of latent variables is searched to generate harder samples to optimize the loss of the target classifier, while the target classifier learns a more robust representation based on the generated harder samples. For the testing phase, the trained target classifier is employed for classification.

A. Classic augmentation techniques

Directly training cDCGANs is prone to over-fitting due to limited vein data. To overcome this problem, we augment the training data using classical approaches, such as geometric transformations (rotation, scaling, flipping, cropping, and translation) and color-enhancing transformations (contrast, brightness, posterize, equalize, and saturation). However, some geometric transformation-based augmentation techniques are unsuitable for vein image augmentation because the geometric change of vein images can disturb the original data distribution. For example, cropped images may differ from the originals, while rotated or translated images may result in more black regions. To enlarge the training data effectively, we perform image augmentation using operators such as contrast, brightness, posterize, and saturation in different directions. In our work, we generate about 50 images for each class and then add them to the original images to obtain a new data set for cDCGAN training.

B. GAN-based augmentation

GAN is an effective data augmentation technology capable of learning the distribution of vein images. This enables it to generate realistic images with good consistency and diversity. Recently, GAN has been employed for data augmentation and has achieved a promising performance in several computer vision tasks. Some works [8], [69]–[72] introduced GANs to enlarge the training data for palm-vein, palmprint, and finger-vein recognition. The standard GAN, however, cannot control the data generation process and thus requires training a model for each class to produce samples, which increases the computation and storage cost. In contrast, conditional deep convolutional generative adversarial networks (cDCGANs) can produce training samples for each class conditioned on additional information. Therefore, we explore cDCGANs to accomplish vein data augmentation in our work. The image generation process is conditioned on additional information, namely the corresponding class labels. The generator takes a random latent variable associated with the class label as its

input and outputs a feature map. The resulting map or a real image associated with the class label is fed to the discriminator to predict its probability of being fake or real. In addition, the TTUR is employed for training cDCGANs to improve the model stability. To further improve the performance, the mechanism of the one-sided label smoothing is incorporated into cDCGANs.

1) Generator: The generator consists of 5 fractionally-stridden convolutional modules, as shown in Table I, which are employed to gradually increase the feature map resolution, thereby satisfying the target resolution of $W \times H$ (training images). For image training, a random latent variable $z \in R^{128\times 1}$ and its class label $y \in R^{num\times 1}$ are combined as the input of the generator, where num is the class number. Then, we compute their summation, and the resulting vector with the length of $(128 + num) \times 1 \times 1$ is fed to a CNN to obtain a map with the size of $8C \times W \times H$. For the first four layers, the fractionally-stridden convolutions can be treated as a pixel upsampling block. The resolution of the input feature map increases two times while the embedded dimension decreases to half in each layer. In the last layer, the resolution of the output is the same as that of the target samples, and then the embedding dimension is transformed to 1. Finally, the generator outputs a synthesized image x' with a size of $W \times H \times 1$.

2) Discriminator: The discriminator includes 5 convolutional layers. We combine class label y with the synthesized image x' or the real image x as the input of the discriminator, which outputs a probability that the input is a fake image or a real one. First, class label y is fed to a fully connected layer to obtain a vector with a size of $WH \times 1$. The resulting vector is reshaped to a 2D feature map with a size of $W \times H$. Second, we pack the resulting 2D feature map and the image (x or x') to obtain a map $X \in R^{W\times H\times 2}$. The resulting map is fed to each convolutional layer, and the resolution of the input feature map is reduced to half while the embedding dimension increases by two times. In the last layer, the output is a probability value. In existing work [73], the one-sided label smoothing scheme refers to replacing 0 and 1 target for a classifier with smoothed values and is employed to reduce the vulnerability of the discriminator to adversarial examples. Inspired by [73], we replace positive classification targets with μ and negative targets with υ to obtain the discriminator in Eq. (1):

$$D(x) = \frac{\mu p_{data}(x) + \upsilon p_{model}(x)}{p_{data}(x) + p_{model}(x)}, \tag{1}$$

where $p_{data}(x)$ and $p_{model}(x)$ denote the data distribution and model distribution, respectively. Similar to work [73], μ and υ are set to 0.9 and 0.1, respectively.

3) Adversarial loss: Let z be the latent variable and y be the conditional information, i.e., class label. The discriminator and generator are represented by D and G, respectively. The generator takes z and y as its input and outputs a synthesized image $x' = G(z|y)$. The discriminator receives either a real image or a synthesized image from the generator as the input and then classifies it as fake or real. The objective function for the conditional GAN is defined in Eq. (2):

$$\begin{aligned} L_{adv}(D,G) =& \mathbb{E}_{x\sim p_{data}(x)}[logD(x|y)] \\ &+ \mathbb{E}_{x\sim p_z(z)}[log(1 - D(G(z|y)))]. \end{aligned} \tag{2}$$

The classic GAN is difficult to train because the divergences that it typically minimizes are potentially not continuous with respect to the generator's parameters. To accelerate the convergence, we employ the WGAN-GP loss [74] as the adversarial loss, which is defined as Eq. (3):

$$\begin{aligned} L_{adv} =& \mathbb{E}_{\widetilde{x}\sim\mathbb{P}_g}[D(\widetilde{x}|y)] - \mathbb{E}_{z\sim\mathbb{P}_r}[(D(z|y))] \\ &+ \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\Delta_{\hat{x}}D(\hat{x})\|_2 - 1)^2], \end{aligned} \tag{3}$$

where $\mathbb{P}_r$ denotes the data distribution, $\mathbb{P}_g$ indicates the generator distribution implicitly defined by $\widetilde{x} = G(z|y)$, and $z \sim p(z)$ (z is sampled from some simple noise distribution). $\mathbb{P}_{\hat{x}}$ is defined to sample uniformly along straight lines between pairs of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$, and λ is the penalty coefficient of the unit gradient norm constraint.

4) GAN training and testing: In the training process, discriminator D is trained to distinguish samples from the generator or from real data, and then generator G is alternately trained to fool the discriminator. To improve the GAN stability, a TTUR is employed to update the weights of the discriminator and generator. We consider discriminator $D_{w'}$ with parameter vector w' and generator G_w with parameter vector w. Let $\mathcal{L}_D$ and $\mathcal{L}_G$ be the loss functions of discriminator D and generator G, respectively. The gradients $g(w, w')$ and $h(w, w')$ are computed as $\nabla_G\mathcal{L}_G$ and $\nabla_D\mathcal{L}_D$, respectively. Then, the approximations of the true gradients are computed as $\hat{g}(w', w) = g(w, w') + V^w$ and $\hat{h}(w', w) = h(w, w') + V^{w'}$, by incorporating random variables V^w and $V^{w'}$. Moreover, the approximations of the gradients are stochastic because they are computed based on the mini-batches of m real-world samples x and synthetic samples x', which are randomly selected. The resulting two time-scale stochastic approximates, namely TTUR [75], are employed to train our GAN. In TTUR, the learning rates $b(n)$ and $a(n)$ are used to update the discriminator and the generator by Eq. (4) and Eq. (5), respectively.

$$w_{n+1} = w_n + a(n)(g(w, w') + V^w), \tag{4}$$

$$w'_{n+1} = w'_n + b(n)(h(w, w') + V^{w'}). \tag{5}$$

Assume that $\mathbb{Z} = \{z_{ck}|c = 1, 2, ..., C; k = 1, 2, ..., K\}$ is a set of latent variables and $\mathbb{Y} = [y_1, y_2, ..., y_C]$ is a class label set, where K is the number of generated samples for each class, and C is the number of class labels. After training the GAN, we sample a latent variable $z \in \mathbb{Z}$ from the latent variable distribution space. The resulting vector z and the corresponding label $y \in \mathbb{Y}$ are taken as the input of the trained generator G_{w*} to generate K synthesized samples x_{cl} by Eq. (6):

$$x' = G_{w*}(z, y). \tag{6}$$

After the training, the generator learns the real data distribution, and is, therefore, capable of producing realistic samples for data augmentation.

TABLE I: Architecture of the generator

Operator	Output	# Kernel	KSize	Stride	Padding
Input noise and label	$(128 + num) \times 1 \times 1$	-	-	-	-
Conv-ReLU-BatchNorm	$1024 \times 4 \times 4$	1024	4	1	0
Conv-ReLU-BatchNorm	$512 \times 8 \times 8$	512	4	2	1
Conv-ReLU-BatchNorm	$256 \times 16 \times 16$	256	4	2	1
Conv-ReLU-BatchNorm	$128 \times 32 \times 32$	128	4	2	1
Conv-ReLU-BatchNorm	$1 \times 64 \times 64$	1	4	2	1

C. Deep learning based classifiers

Various deep-learning classifiers have been recently investigated for vein recognition. Assume $\mathbb{X} = \{x_{cl}|c = 1, 2, ..., C; l = 1, 2, ..., L\}$ is a training set, where x_{cl} denotes the lth image from the cth class, and L is the number of samples for each class. Let ϕ_θ be any feature extraction model, e.g., ResNet [76], VGG [77], FV-CNN [49], PV-CNN [8], FVRAS-Net [55], Lightweight CNN [60], and Vit [64], where θ is a trainable weight. The classifier maps example $x \in \mathbb{X}$ into label $y \in \mathbb{Y}$. A deep learning-based classifier is implemented by the combination of ϕ_θ parameters and a softmax regression layer that predicts the posterior class probability by Eq. (7):

$$P_{Y|X}(y_c|x) = \frac{e^{w_c^T \phi_\theta(x)}}{\sum_i e^{w_i^T \phi_\theta(x)}}, \tag{7}$$

where w_c is the vector of classification parameters of the cth class. Given a training set $\{x_c, y_c\}$, parameter θ and $W = \{w_c\}_{c=1}^C$ are learned by minimizing Eq. (8):

$$L_{ce}(x, y) = -\,l\,og\frac{e^{w_c^T \phi_\theta(x)}}{\sum_i e^{w_i^T \phi_\theta(x)}}. \tag{8}$$

D. Adversarial augmentation

The trained generator in Eq. (6) can generate vein images, based on which the deep learning-based classifier is effectively trained for vein recognition. However, the generator is capable of generating vein samples based on different input latent variable z, but it is impossible to exploit all synthesized vein samples in the data distribution space for classifier training. Generally, finite samples, e.g., about 500 samples for each class in work [8], are generated for classifier training. Moreover, the generated samples are repeatedly input to the classifier for training, which likely results in over-fitting. Such finite samples hardly guarantee that the target classifier is effectively trained. To overcome this problem, an adversarial framework is proposed to jointly optimize the classifier training and augmentation data set generation. As described in Section II-B4, $\mathbb{Z} = \{z_{ck}|c = 1, 2, ..., C; k = 1, 2, ..., K\}$ is a latent variable set, where z_{ck} is the kth latent variable for the cth class, and $\mathbb{X} = \{x_{cl}|c = 1, 2, ..., C; l = 1, 2, ..., L\}$ is a training set associated with a label set $\mathbb{Y} = [y_1, y_2, ..., y_C]$. Based on the generator G_w, we can obtain the synthesized image set $\mathbb{X}' = \{G_w(z_{ck}, y_c)|k = 1, 2, ..., K\} = \{x'_{ck}|c = 1, 2, ..., C; k = 1, 2, ..., K\}$. Then, we combine the two sets $\mathbb{X}$ and $\mathbb{X}'$ to constitute an augmented data set Ω. In this dataset, there are $(K+L)$ images for each class, which results in $(K + L) \times C$ images. The target classifier ϕ_θ is trained to minimize the loss in Eq. (8) based on the images from the augmented data set Ω. The generator $G_w(\cdot)$ attempts to produce a synthesized image set X' by searching a latent variable set Z as its inputs to increase the training loss of the target network ϕ_θ through adversarial learning. Finally, an equilibrium can be reached where the learned representation achieves maximized performance.

Consider the target classifier $\phi_\theta(\cdot)$ with a loss function $L_{ce}(\cdot)$. The trained generator $G_{w*}(\cdot)$ transforms the latent variable space into the sample data space. The optimizing process of θ can be defined as the following minimization problem, as shown in Eq. (9):

$$\begin{aligned}\theta^* =& arg\,\min_\theta[\mathop{\mathbb{E}}_{z_{ck}\in\mathbb{Z}}[L_{ce}(\phi_\theta(G_{w*}(z_{ck}, y_c)), y_c) \\ &- \lambda_1 L_{ce}(\phi_{\hat{\theta}}(G_{w*}(z_{ck}, y_c)), y_c)) \\ &+ \lambda_2 cosine(G_{w*}(z_{ck}, y_c), x_c)] \\ &+ \mathop{\mathbb{E}}_{x_{cl}\in\mathbb{X}} L_{ce}(\phi_\theta(x_{cl}), y_c)],\end{aligned} \tag{9}$$

where the prediction label of each query in the current mini-batch is a function of classifier weights, and thus we optimize θ over them. Similarly, the latent variables in $\mathbb{Z}$ are treated as free variables of composite function $\phi_\theta(G_{w*}(\cdot, y_c), y_c)$, which can be optimized for a given θ. During the training process, the synthesized images and original images are mixed to feed the target classifier for training. To facilitate the description, we split it into two terms: the first term $L_{ce}(\phi_\theta(G_{w*}(z_{ck}, y_c)), y_c)$ is the loss computed based on synthesized images, and the second $L_{ce}(\phi_\theta(x_{cl}), y_c)$ is the loss computed based on original images, as shown in Eq. (9). The $cosine(\cdot)$ is the function of $cosine$ similarity and $\phi_{\hat{\theta}}$ is a teacher model whose weights are updated as an exponential moving average of the target model's weights, i.e., $\hat{\theta} \leftarrow \xi\hat{\theta} + (1 - \xi)\theta$ and ξ=0.9 [78], which are employed as two regularization terms to control the selection of the latent variables in set $\mathbb{Z}$. In our experiments, $\lambda_1 = \lambda_2 = 0.5$.

The problem in Eq. (9) is usually solved by vanilla SGD with a learning rate of α and a batch size of N, and the training procedure for each batch can be expressed by Eq. (10):

$$\begin{aligned}\theta(t+1) =& \theta(t) - \alpha\frac{1}{N}[\sum_{n=1}^{N_1} \nabla_\theta[L_{ce}(\phi_\theta(G_{w*}(z_{ck}, y_c)), y_c) \\ &- \lambda_1 L_{ce}(\phi_{\hat{\theta}}(G_{w*}(z_{ck}, y_c)), y_c)) \\ &+ \lambda_2 cosine(G_{w*}(z_{ck}, y_c), x_c)] \\ &+ \sum_{n=1}^{N_2} \nabla_\theta L_{ce}(\phi_\theta(x_{cl}), y_c)],\end{aligned} \tag{10}$$

where N_1 and N_2 are respectively the numbers of the generated and the original samples in a batch, and $N = N_1 + N_2$. As

the $cosine$ similarity and the teacher model are independent of variable θ, Eq. (10) can be changed to Eq. (11):

$$\begin{aligned}\theta(t+1) =&\theta(t) - \alpha\frac{1}{N}[\sum_{n=1}^{N_1}\nabla_\theta L(\phi_\theta(G_{w*}(z_{ck}, y_c)), y_c) \\ &+ \sum_{n=1}^{N_2}\nabla_\theta L_{ce}(\phi_\theta(x_{cl}), y_c)].\end{aligned} \tag{11}$$

Note that the training procedure employs an average over N instances of gradient computation to reduce gradient variance, which results in a faster convergence of the target network. However, it is also prone to over-fitting owing to the limited training data. To overcome this issue, the set of training samples generated by the GAN-based augmentation network will act as adversaries to the target network, resulting in a min-max problem to self-train the network. Such a self-supervisory objective may be sufficiently challenging to prevent the learned representation from overfitting the objective. Therefore, we can mathematically define the objective as the following maximization problem in Eq. (12):

$$\begin{aligned}\mathbb{Z}^* =&arg\,\max_{\mathbb{Z}}[\mathop{\mathbb{E}}_{z_{ck}\in\mathbb{Z}}[L_{ce}(\phi_\theta(G_{w*}(z_{ck}, y_c)), y_c) \\ &- \lambda_1 L_{ce}(\phi_{\hat{\theta}}(G_{w*}(z_{ck}, y_c)), y_c)) \\ &+ \lambda_2 cosine(G_{w*}(z_{ck}, y_c), x_c)] \\ &+ \mathop{\mathbb{E}}_{x\in\mathbb{X}} L_{ce}(\phi_\theta(x_{cl}), y_c)].\end{aligned} \tag{12}$$

To solve the above problem, a gradient ascent is applied for the update with a learning rate of β, and the parameter update rule is defined in Eq. (13):

$$\begin{aligned}z_{ck}(t+1) =&z_{ck}(t) + \beta\frac{1}{N}\sum_{n=1}^{N}\nabla_{z_{ck}}[L_{ce}(\phi_\theta(G_{w*}(z_{ck}, y_c)), y_c) \\ &- \lambda_1 L_{ce}(\phi_{\hat{\theta}}(G_{w*}(z_{ck}, y_c)), y_c)) \\ &+ \lambda_2 cosine(G_{w*}(z_{ck}, y_c), x_{cl})].\end{aligned} \tag{13}$$

Intuitively, the optimization of Eq. (13) can be explained by noticing that the loss of the classifier (Eq. (8)) is maximized when $L_{ce}(\phi_\theta(G_{w*}(z_{ck}, y_c)), y_c)$ is maximized and $L_{ce}(\phi_{\hat{\theta}}(G_{w*}(z_{ck}, y_c)), y_c) - cosine(G_{w*}(z_{ck}, y_c), x_{cl})$ is minimized for the adversarial training of $\mathbb{Z}$. In other words, this tends to push the synthesized samples far away from the real samples in the same class while ensuring that such synthesized samples are recognizable for a teacher model and kept, at the same time, within a constrained distance from the original images. This scheme ensures the generating challenging samples closely tracking the updates of the classifier. The adversarial learning of target network training and the latent variable set to search for augmentation data are summarized in Algorithm 1.

1) Improvement techniques: The training procedure for our model is similar to GAN's. To improve performance, non-saturating loss and label smoothing are introduced into the adversarial model.

Algorithm 1 Joint Training of Target Network and a latent variable set to search for augmentation data.

Input: The set of label $\mathbb{Y} = [y_1, y_2, ..., y_C]$; The initial set of latent variables, $\mathbb{Z} = [z_{11}, z_{12}, ..., z_{CK}]$; Original training sample set $\mathbb{X} = [x_{11}, x_{12}, ..., x_{CL}]$; The classifier, ϕ_θ; The trained generator, G_{w*};
Output: θ^*, $\mathbb{Z}^*$;

```
For 1 ≤ e ≤ epoch;
    Input latent variable set ℤ to trained generator G_{w*} to generate K × C training samples 𝕏′. Then we combine them with original samples 𝕏 to obtain the augmentation data set Ω; We randomly select N samples from Ω to construct mini-batches.
    For 1 ≤ t_1 ≤ T_1;
        Update θ(t+1) according to Eq. (13);
    end
    For 1 ≤ t_2 ≤ T_2;
        Update z_ck(t+1) via Eq. (15), k = 1, 2, ..., K;
        Update the set ℤ = [z_11(t+1), z_12(t+1), ..., z_CK(t+1)] by latent variable z_ck(t+1);
    end
end
```

Non-saturating loss. Most of the classification tasks employ the following cross-entropy loss as the loss function in Eq. (14):

$$L_{ce}(z_{ck}, y_c) = \sum_{c=1}^{C} -y_c log(\phi_\theta(G_{w*}(z_{ck}, y_c), y_c)). \tag{14}$$

Based on such a loss, the gradient of the first term in Eq. (9) often saturates in the maximization problem w.r.t $\mathbb{Z}$ when the target model's predictions are very confident. To overcome this problem, instead of using Eq. (14), a non-saturating loss defined in Eq. (15) is employed to train the augmentation model.

$$L_{ce}(z_{ck}, y_c) = \sum_{c=1}^{C} y_c log(1 - \phi_\theta(G_{w*}(z_{ck}, y_c), y_c)). \tag{15}$$

Label smoothing technique. Label smoothing is a technique that has been applied to various classification tasks, which is defined as Eq. (16):

$$\hat{y_c} = (1-\omega)y_c + \omega/C, \tag{16}$$

where $\omega \in [0, 1)$ is a smoothing parameter. The one hot label y_c can be transformed to smoothing label $\hat{y_c}$ based on Eq. (16). The label smoothing technique prevents exploding gradients when the classifier decision is very confident under the non-saturating loss, especially for easy tasks or strong target models. In our experiments, to mitigate exploding gradients, the smoothed label is introduced into the classifier (the first term in Eq. (9)) during the parameter updating of the latent variable set. To achieve a fair comparison, it is not applied when updating the classifier.

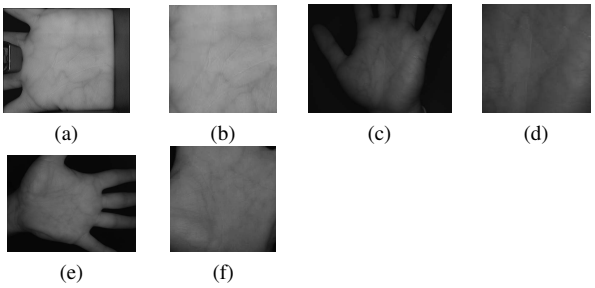

Fig. 2: Preprocessing results. (a) Original image from database A; (b) ROI from (a); (c) Original image from database B; (d) ROI from (c); (e) Original image from database C; (f) ROI from (e).

III. Experiments

In this section, we conduct experiments on three public palm-vein datasets to evaluate the performance of the proposed approach. To test the identification performance of Advein$\mathbb{AU}$, we employ various classifiers, namely ResNet [76], VGG [77], FV-CNN [49], PV-CNN [8], FVRAS-Net [55], Lightweight CNN [60], and Vit [64] as baselines, which are incorporated into Advein$\mathbb{AU}$ to improve classification performance. In addition, we compare Advein$\mathbb{AU}$ with existing vein data augmentation approaches, namely classical data augmentation techniques, FV-GAN [71] and Palm-GAN [70], in terms of improving the classifiers' classification accuracy. All experiments are conducted on PyTorch with NVIDIA GeForce GTX 3090 GPUs.

A. Databases

1) Database A: The PolyU multispectral palmprint dataset [79] includes 6,000 (250 subjects $\times$ 2 palms $\times$ 6 images $\times$ 2 sessions) palm-vein images collected by near-infrared (NIR) light. The images are captured from 250 volunteers, with two separate sessions with an average time interval of 9 days. Each volunteer provides 24 (2 palms $\times$ 6 images $\times$ 2 sessions) images from two hands, each providing 6 images at a session.

2) Database B: The images in Tongji University palmprint dataset [80] are collected from 300 subjects in a contactless way at two sessions with an average time interval of about two months. Each subject provides 40 images (10 images $\times$ 2 sessions $\times$ 2 palms), with each palm providing 10 images at one session. As a result, there are 12,000 palm images (300 objects $\times$ 2 palms $\times$ 10 images$\times$ 2 sessions) from 300 subjects in total.

3) Database C: The VERA PalmVein [81] dataset is built by the Idiap group. There are 2,200 images (110 objects $\times$ 2 palms $\times$ 5 images $\times$ 2 sessions) collected from 110 volunteers at two independent sessions. Each volunteer provides two palms (left and right palms), and 5 images are captured from each palm in one session. As a result, 20 images are collected from each volunteer at two sessions.

As the original images in the three palm-vein datasets include the background regions, which do not provide discriminating information, we employ the approach in [82] to extract the region of interest (ROI) images, which are then normalized to 100×100. Fig. 2 shows the ROI images and original palm-vein images from the three datasets. To facilitate the deep learning model training, the class label is generally transformed to an one-hot label vector. For database A, there are 500 palms in total. If each palm is treated as a class, there are 500 classes. The one-hot vector is denoted as $l \in R^{1\times 500}$. For the first class, the label l_1 is [1 0 0 $\cdots$ 0]. Similarly, the label vectors for databases B and C are denoted as $l' \in R^{1\times 600}$ and $l'' \in R^{1\times 220}$, respectively.

B. Parameter selection

In our approach, the cosine similarity and teacher model are incorporated into Eq. (9) as regularization terms to control the quality of the generated images, with λ_1 and λ_2 as the corresponding regularization coefficients. To investigate the upper bound and lower bound of the proposed model, we list the recognition accuracy of the classic classifier, i.e., Resnet, with different λ_1 and λ_2 values. As shown in Table II, when λ_1 and λ_2 are equal to 0, Resnet, trained on the synthesized images, achieves the lowest accuracy on the three databases. The best identification accuracy is achieved for Resnet when $\lambda_1 = 1$ and $\lambda_2 = 1$. When λ_1 and λ_2 increase to 1,000, respectively, the recognition accuracy decreases. From the experimental results, we observe, therefore, that quality control can further improve classifier accuracy. The best performance and worst performance of our model are obtained for different λ_1 and λ_2.

C. Experimental setting

To assess Advein$\mathbb{AU}$, we split each dataset into two subsets. For dataset A, images from different palms are treated as different classes, and we obtain, as a result, 500 classes. We select images from the first session for training and images from the second session for testing. In this way, there are 3,000 (500 hands $\times$ 6 images from the first session) images in training set A1 and 3,000 (500 hands $\times$ 6 images from the second session) images in testing set A2. Similarly, dataset B is divided into two datasets: the training dataset B1 with 6,000 (600 hands $\times$ 10 images from the first session) images, and testing set B2 with 6,000 (600 hands $\times$ 10 images from the second session) images. For dataset C, there are 1,100 (220 hands $\times$ 5 images from the first session) images in training set C1 and 1,100 (220 hands $\times$ 5 images from the second session) images in testing C2. The training sets, i.e., A1, B1, and C1, are used to train cDCGAN (as shown in Fig. 1), respectively, in order to learn the image distribution. After training, the resulting generator is combined with the existing classifiers, i.e., ResNet, VGG, FV-CNN, PV-CNN, FVRAS-Net, Lightweight CNN, and Vit for adversarial learning, as shown in Fig. 1 and Algorithm 1. During the training of Advein$\mathbb{AU}$, we randomly generate 50 latent variables to construct a set for each class. The latent variables and class labels are taken as input of the generator to produce 50 images for each class. Combining the generated images and the original ones, we obtain an augmented training set. There are 28,000 images (500 hands $\times$ (6 images from the first session + 50 synthesized images)) for dataset A, 36,000 images (600 hands $\times$ (10 images from the first session + 50

TABLE II: Identification accuracy of the Resnet classifier with different λ_1 and λ_2.

Accuracy (%)	λ_1=0, λ_2=0	λ_1=0.001, λ_2=0.001	λ_1=1, R2=1	λ_1=100, λ_2=100	λ_1=1000, λ_2=1000
Database A	98.01	98.04	98.78	98.55	98.10
Database B	93.08	93.14	94.08	93.81	93.15
Database C	93.66	94.00	95.16	94.77	94.06

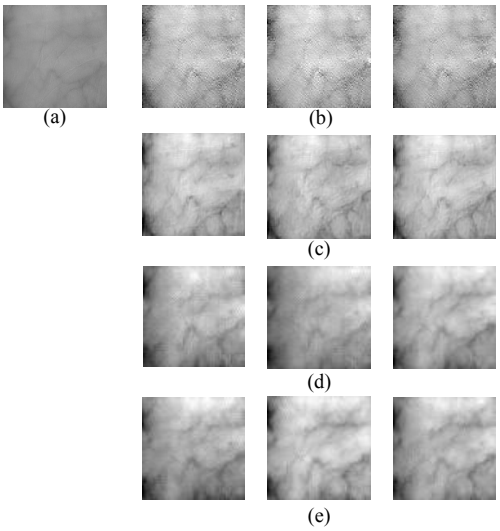

Fig. 3: Generated images. (a) Original palm-vein images; (b) images generated by FVGAN; (c) images generated by DCGAN; (d) images generated by cDCGAN; (e) images generated by cDCGAN by Advein$\mathbb{AU}$.

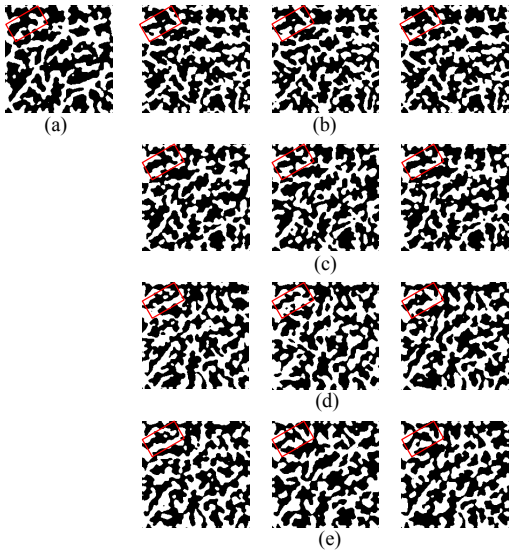

Fig. 4: Vein patterns exacted from (a) Original binary palmvein images; (b) images generated by FV-GAN; (c) images generated by DCGAN; (d) images generated by cDCGAN; (e) images generated by cDCGAN by Advein$\mathbb{AU}$.

synthesized images)) for dataset B, and 6,050 images (110 hands $\times$ (5 images from the first session + 50 synthesized images)) for dataset C. Subsequently, the classifier is trained based on such augmented datasets. After several iteration steps, we fix the weights of the trained classifier to update the latent variable set for each class. In this way, the classifier weights and the latent variable set are alternatively updated until convergence. Finally, the resulting classifier is employed for vein identification. In our experiments, the identification accuracy of the various classifiers on the testing set A2 from dataset A, the testing set B2 from dataset B, and the testing set C2 from dataset C is reported to evaluate the performance of our approach. Moreover, the classical data augmentation method and GAN-based data augmentation methods, i.e., FCGAN [71] and DCGAN [70], are also considered to verify the efficacy of Advein$\mathbb{AU}$.

D. Visual assessment

In this section, we visually assess the quality of the synthesized images to investigate the performance of the considered approaches. In the experiments, the augmentation approaches, namely FV-GAN, DCGAN, cDCGAN, and Advein$\mathbb{AU}$, are trained to produce images. The resulting synthesized images on dataset A are depicted in Fig. 3. Fig. 3(a) shows an original palm-vein image, while Fig. 3(b), Fig. 3(c), Fig. 3(d), and Fig. 3(e) show synthesized images generated by FV-GAN, DCGAN, cDCGAN, and Advein$\mathbb{AU}$, respectively. To facilitate comparison, we extract the vein patterns from synthesized gray-scale images by an existing segmentation model [19], and the resulting binary images are shown in Fig. 4. Furthermore, we compute the average of three generated images for each approach and obtain an averaged binary image by a threshold of 0.5. Then, we compute the difference between each binary image and the resulting averaged binary image. The difference images for FV-GAN, DCGAN, cDCGAN, and Advein$\mathbb{AU}$ are shown in Fig. 5. Note that the difference pixels are labeled in cyan.

1) Consistence: Comparing the synthesized images (Fig. 3(b), Fig. 3(c), Fig. 3(d), and Fig. 3(e)) with the original image (Fig. 3(a)), we observe that images generated by DCGAN, cDCGAN, and Advein$\mathbb{AU}$ show realistic structures and configurations of vein patterns while preserving similar visual content with the original image. For example, the generated samples display similar smoothness, global structures of veins, and detailed texture distribution to the original samples (Fig. 3(a)). By contrast, there is a significant distribution difference between the generated samples from FV-GAN and the original samples. For instance, there are abnormal textures or noise in the generated images by FV-GAN (Fig. 3(b)), which may result in the degradation of identification accuracy.

2) Diversity: In practical applications, the acquired palm-vein images include various variations, such as translation, rotation, scale, and distribution in local and global regions, especially for images collected contactless. In general, the variations in a global region can be normalized by image

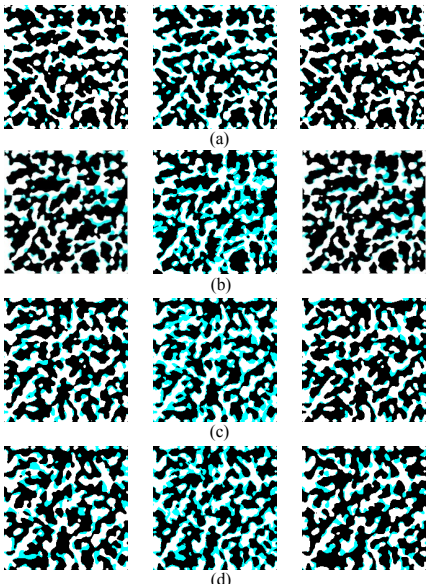

Fig. 5: Difference images from (a) images generated by FV-GAN; (b) images generated by DCGAN; (c) images generated by cDCGAN; (d) images generated by cDCGAN by Advein$\mathbb{AU}$.

prepossessing-based approaches [13], [80], [82]. However, it is difficult to reduce the local variations, such as local rotation, local translation, local scale, and local distortion, that usually exist in contactless hand-vein identification. Therefore, the diversity of synthesized images is helpful to improve the performance of classifiers.

Comparing the experimental results from the four approaches with the original palm-vein images in Fig. 3, Fig. 4, and Fig. 5, we observe that Advein$\mathbb{AU}$ and cDCGAN outperform the other two approaches, as new structures and configurations of vein patterns emerge in the generated images. For instance, as shown in the red rectangle of Fig. 4, the locations and directions of some vein patterns are changed between the generated images (Fig. 4(c) and Fig. 4(d)) by Advein$\mathbb{AU}$ and cDCGAN and the original images (Fig. 4(a)), but the pattern of the region (hole) formed by vein patterns remains. Interestingly, we observe from Fig. 4(d) and Fig. 4(e) that the proposed approach employs the trained generator of cDCGAN to synthesize more diverse images than cDCGAN. Fig. 5 shows the consistent trend that there is a significant difference among images generated by Advein$\mathbb{AU}$, which implies that our approach is capable of producing more diverse images. Such results are attributed to the following facts: 1) cDCGAN aims to synthesize images, so the random latent variables are input to the trained generator for data augmentation. 2) Instead, Advein$\mathbb{AU}$ focuses on generating challenging images to improve classifiers' robustness so the latent variables are updated by maximizing their loss. In other words, during training, the latent variables are perturbed to generate adversarial samples that increase the vein classifier loss. This is why the generated adversarial samples actually seek a compromise between image quality and fidelity to the original training images and diversity w.r.t the latter so as to increase classifier loss. This is different from traditional GANs that overlook the increased loss-diversity aspect because the generated images are too similar to the original ones. From the vein patterns in Figs. 5(a), 5(b), and 5(c), we observe a similar vein pattern distribution among the samples generated from the two baselines (FV-GAN and DCGAN), which results in the degradation of diversity. The reason is that, similar to cDCGAN, the baselines focus on image generation instead of classifier performance improvement. By contrast, we observe that there are larger differences among the three images generated by our approach as more cyan pixels show up in each image. In general, the diversity of samples may be related to the classification performance. In other words, the images with more variations may allow more effective training of the classifiers, thereby leading to high classification accuracy.

TABLE III: Average variance of original images and synthesized images on three databases.

Method	Database A	Database B	Database C
Original images	0.0022	0.0016	0.0015
Classic approach	0.0049	0.0015	0.0045
FV-GAN	0.0088	0.0092	0.0093
DCGAN	0.0126	0.0159	0.0124
cDCGAN	0.0153	0.0217	0.0178
AdveinAU	0.0177	0.0223	0.0181

E. Quantitative assessment

For quantitative assessment, we randomly select original samples and generated samples from one class in the three databases, respectively, and compute the average variance for comparison. For each database, we compute the variance of the original images and synthesized images for each class according to different approaches. Then, the average variance of all the classes is calculated to verify the performance of our approach. The experimental results on the three databases are illustrated in Table III. From these results, we observe that the images generated by our approach show a larger variance, which implies that our approach outperforms the other approaches in terms of producing diverse images.

In addition, the Inception Score (IS) [83] and Fréchet inception distance (FID) [84] are two important parameters that have been widely employed to quantitatively assess the quality of the generated images by GANs in the existing works [85]–[90]. IS measures the quality and diversity of the generated images. It is calculated using a pre-trained inception model to classify the generated images and then compute the average of the predicted class probabilities for each image. A higher IS indicates that the generated images have high quality and diversity. FID measures the similarity between the distribution of the real images and that of the generated images. It uses the activation of an intermediate layer of a pre-trained inception model to calculate the Fréchet distance between the two distributions. A lower FID indicates that the generated images are closer to the real images in terms of distribution. In our experiments, we compute IS and FID of the synthesized images from FV-GAN, DCGAN, cDCGAN, and Advein$\mathbb{AU}$. Table IV lists the IS and FID of the images generated by the various approaches on the three databases.

TABLE IV: Image quality evaluation of various methods on three databases

Approach	Database A		Database B		Database C	
	IS ↑	FID ↓	IS ↑	FID ↓	IS ↑	FID ↓
FV-GAN	1.8496	31.8260	2.0425	61.6108	1.8523	52.4167
DCGAN	1.7949	47.5943	1.9025	85.0079	1.5941	86.0344
cDCGAN	1.9209	31.4126	2.0425	51.0132	2.0321	36.4344
AdveinAU	1.9846	28.5481	2.3393	32.0308	2.1798	35.1698

The experimental results in Table IV show that our approach achieves the lowest IS and the highest FID among the existing approaches on the three databases, which demonstrate that our approach can produce consistent and diverse vein images. This conclusion is also supported by the visual assessment experiments in Fig. 4 and Fig. 5.

F. Identification results

We have conducted rigorous experiments on three public palm-vein datasets to evaluate the efficiency of Advein$\mathbb{AU}$. We report the identification accuracy results on the datasets collected at two separate sessions, which is the realistic practical scenario. As described in Section III-C, for dataset A, there are 3,000 images from the first session in training set A1, based on which the augmentation approaches, i.e., the classic approach, FCGAN, DCGAN, and cDCGAN, are trained to generate 50 images for each class. As a result, the augmented dataset A3 includes 28,000 images (500 hands $\times$ (6 images from the first session + 50 synthesized images)). Based on A3, seven existing classifiers, namely ResNet, VGG, FV-CNN, PV-CNN, FVRAS-Net, Lightweight CNN, and Vit, are trained for vein identification. Each classifier is trained five times, and the average identification accuracy on testing set A2 is reported for comparison. For Advein$\mathbb{AU}$, the trained generator of cDCGAN is combined with each classifier for adversarial training, and the resulting classifier is used for classification. The identification results on testing A2 are illustrated to verify the efficacy of our approach. Table V lists the identification results of seven classifiers under different data augmentation approaches for dataset A. For dataset B, three data augmentation approaches, i.e., FCGAN, DCGAN, and cDCGAN, are trained based on dataset B2 to produce 50 images for each class, which results in an augmented dataset B3 with 36,000 images (600 hands $\times$ (10 images from the first session+ 50 synthesized images)). Similarly, the augmented dataset C3 consists of 6,050 images (110 hands $\times$ (5 images from the first session + 50 synthesized images)) for dataset C. We train the seven classifiers on augmented datasets B3 and C3 and show the average identification accuracy on testing sets B2 and C2, respectively. In addition, we incorporate the seven classifiers with the trained generator of cDCGAN for training, respectively; the identification results are reported for comparable analysis. Table VI and Table VII list the recognition results of the various approaches for dataset B and dataset C, respectively. The experimental results (Table V, Table VI, and Table VII) show that Advein$\mathbb{AU}$ significantly outperforms the existing data augmentation approaches, i.e., the classic approach, FCGAN, DCGAN, and cDCGAN and achieves the highest identification accuracy, i.e., 99.24% on dataset A, 95.87% dataset B, and 96.20% on dataset C. As described in the three tables, the seven classifiers achieve the lowest identification results when only original images from dataset A, dataset B, and dataset C are employed for training. After training on the enlarged data set produced by the data augmentation approaches (classic approach, FCGAN, DCGAN, and cDCGAN), the seven classifiers achieve an improvement in the identification accuracy, which implies that the vein classifier benefits from the data augmentation scheme for vein recognition tasks. Compared to the deep learning-based data augmentation approaches, the classical data augmentation approach achieves the lowest improvement on the three datasets. Such poor performance can be explained by the following facts: 1) The classic data augmentation approach is based on the $apriori$ assumption that there are variations such as translation, rotation, and scale in the captured images. In fact, as the vein image capture process is affected by many factors, the collected images may include more complex variations, such as local distortion and local rotation in unseen samples. 2) The pixels in an image can produce different distribution variations, so it is impossible to model all variations. Most of the classic data augmentation approaches employ variation attributes which are easily observed and modeled, e.g., global translation or global rotation. 3) It is hard to build a mathematical model to describe such variations in images. In general, we employ some simple affine transformation functions, such as translation and rotation, to model them. On the contrary, the deep learning-based data augmentation approaches, such as FV-GAN, DCGAN, and cDCGAN, have shown a highly robust feature representation capacity and are capable of learning the real data distribution by the adversarial learning to produce realistic vein images (Fig. 3 and Fig. 4). However, the existing GAN based data augmentation models, i.e., FV-GAN, DAGAN, and cDCGAN, aim to synthesize realistic images instead of improving the classifier performance. In principle, if the real data distribution is successfully learned, all images are sampled from such a distribution to enlarge the training data, which results in the improvement of classification accuracy. Nevertheless, it is impossible to generate all possible relevant samples from such a distribution for classifier training because of limited storage space and computation resources. As a result, limited sample sizes, e.g., hundreds of images for each class are randomly produced to train the classifiers. The resulting images may include redundant information, thereby lacking diversity, which comprises the classification performance. Moreover, such generated images will be input to a classifier repeatedly during the whole training process, which may result in an inevitable overfitting in long epoch training. By contrast, Advein$\mathbb{AU}$ focuses on searching an optimal set of latent variables to generate the images along with the training process rather than synthesizing new images in an offline way. As the image generation process is guided by maximizing the loss of the classifier, challenging images with good consistency and diversity (Fig. 3(d) and Fig. 4(d)) may be produced and gradually updated to train a more robust classifier during the adversarial training process. This is the reason why Advein$\mathbb{AU}$ significantly outperforms cDCGAN and achieves better performance even if they share

TABLE V: Identification accuracy of various classifiers improved by different data augmentation approaches on dataset A.

Classifiers	Without augmentation	Classic approach	FV-GAN	DCGAN	cDCGAN	Adverin$\mathbb{AU}$
ResNet	97.10	97.37	97.60	97.57	97.67	98.80
VGG	96.44	97.17	97.73	97.40	97.74	98.90
FV-CNN	98.30	98.53	98.86	98.83	98.87	99.24
PV-CNN	96.24	96.99	97.80	98.00	97.97	98.84
FVRAS-Net	96.48	96.80	97.14	97.10	97.21	98.47
Lightweight CNN	93.02	94.54	95.77	95.24	95.81	97.84
Vit	95.14	96.84	98.30	97.54	98.40	99.03

TABLE VI: Identification accuracy of various classifiers improved by different data augmentation approaches on dataset B.

Classifiers	Without augmentation	Classic approach	FV-GAN	DCGAN	cDCGAN	Adverin$\mathbb{AU}$
ResNet	86.42	87.07	90.80	90.47	92.47	94.15
VGG	87.50	89.00	91.26	90.60	91.28	94.13
FV-CNN	92.25	92.70	93.13	92.95	93.14	95.87
PV-CNN	89.10	90.97	92.48	91.88	92.48	94.83
FVRAS-Net	89.05	84.40	85.32	85.29	85.39	87.13
Lightweight CNN	93.02	90.37	91.35	91.18	91.85	93.77
Vit	84.65	85.02	85.45	85.25	85.46	87.60

TABLE VII: Identification accuracy of various classifiers improved by different data augmentation approaches on dataset C.

Classifiers	Without augmentation	Classic approach	FV-GAN	DCGAN	cDCGAN	Adverin$\mathbb{AU}$
ResNet	92.03	92.74	93.31	92.84	93.32	95.29
VGG	91.49	92.33	92.75	92.45	92.75	96.20
FV-CNN	90.67	91.13	92.27	91.87	92.30	94.11
PV-CNN	93.81	94.26	94.61	94.65	94.66	96.20
FVRAS-Net	89.86	90.40	91.84	91.77	92.01	93.32
Lightweight CNN	87.62	88.04	88.63	88.37	88.63	91.73
Vit	90.31	91.07	91.70	91.88	91.94	93.39

the same generator. Overall, most vein classifiers achieve a higher recognition rate on database A compared to other databases. Consistent with this trend, our approach obtains the highest accuracy of 99.24% on database A. To verify whether our approach suffers from overfitting, we conduct the cross-validation experiment on database A, and the average accuracy on four random data sets is 99.28%. Therefore, such good performance can be explained by the fact that there are fewer intra-class variations among palm-vein images rather than by overfitting.

In fact, Adverin$\mathbb{AU}$ brings a bridge between the GAN and the classifier. Certainly, we can replace the cDCGAN in Fig. 1 with other image generation models, such as DCGAN, FV-GAN, or a classical data augmentation approach, to build an adversarial model for classifier training. Moreover, other existing classifiers can be incorporated into our framework to improve performance. In addition, Adverin$\mathbb{AU}$ is trained in two steps. Concretely, cDCGAN is firstly trained to obtain a fixed generator, which is combined with the classifier for adversarial learning. Factually, we also directly train the whole model (Fig. 1) in an end-to-end way to obtain a robust classifier. However, the optimization problem with additional loss functions may be more challenging.

To further estimate the performance of our approach, we show the confusion matrices of various approaches to evaluate type I and type II errors. As described in the previous section, each class includes 56 samples (6 original images and 50 synthesized images) for database A, 60 samples (10 original images and 50 synthesis images) for database B, and 55 samples (5 original images and 50 synthesized images) for database C. Based on these samples, we can train various vein classifiers for recognition. However, there are lots of classes for each database, which results in lots of confusion matrices. Therefore, we only compute the confusion matrices of ten classes based on the prediction scores by the state-of-the-art vein classifier, i.e., FV-CNN. The resulting confusion matrices for FCGAN, DCGAN, cDCGAN, and Adverin$\mathbb{AU}$ on the three databases are illustrated in 6. From 6, we can observe that the classifier trained on the images generated by our approach achieves the best performance, which implies that our model generates high-quality samples for data augmentation.

As an effective data augmentation approach, GAN has been employed to produce images so as to improve the performance of deep learning-based approaches. For the same purpose, some researchers introduced it into vein recognition tasks for data augmentation. Generally, vein training data are limited and thus insufficient for training a GAN, so some classic data augmentation approaches, such as rotation and translation, are employed to generate additional samples. For vein classification tasks, data augmentation aims to improve the performance of vein classifiers. The quality of synthesized images is, therefore, related to vein classifiers instead of human vision. In our work, we mainly focus on searching high-quality samples from the synthesized vein images to train a robust vein classifier instead of improving the capacity of GAN-based models. If the GAN is capable of learning the distribution space of vein images, our approach can obtain samples with high quality to train a more robust classifier. By contrast, if it is not, our approach can search samples with better quality from such a space to improve vein classifier performance by

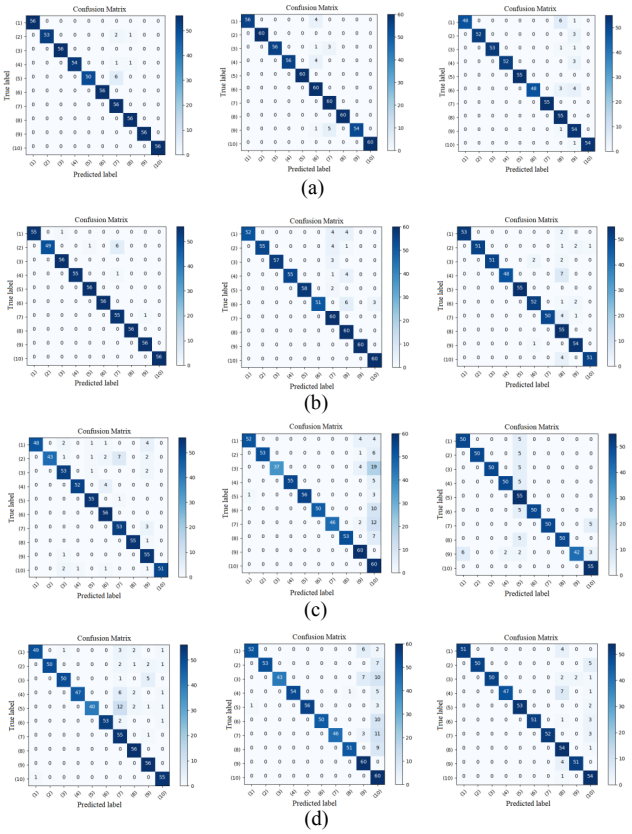

Fig. 6: The confused matrix based on (a) images generated by AdveinAU; (b) images generated by cDCGAN; (c) images generated by DCGAN; (d) images generated by FVGAN.

the adversarial learning. In other words, our approach aims to find better samples from distribution space learning by a GAN to improve classifier performance. The quality of the generated images is hence estimated or controlled based on classification performance. From our experimental results (Tables V, VI, and VII), we can see that, based on the generated images searching from the cDCGAN by our approach, various vein classifiers achieve higher recognition rates than the ones based on randomly generated from cDCGAN, which implies that our approach can control sample quality to train more robust vein classifiers.

It is worth noting that if our model is effectively trained on a sufficiently large dataset, it will learn the feature distribution space as well as the noise distribution in the samples. Such a model would be able to generate diverse samples with similar noise, based on which the training classifier would become robust to noise in recognition systems. By contrast, if such a large dataset is not available, the noise distribution may be overlooked in the generated images. In practical applications, images from different physical devices have different noise distributions. If we aim to improve the generalization of our approach to different physical devices, domain transfer learning may be a more effective solution.

Generally, there is a difference between the quality of generated images by a GAN and the performance of the GAN. In

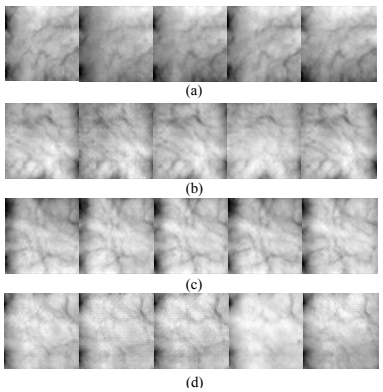

Fig. 7: The images generated by AdveinAU for four classes.

the data augmentation schemes, high-quality images generated by a GAN are usually associated with the ones with good consistency and large diversity. In other words, the generated images with high quality should preserve similar visual content to the original images but include large variations for vein patterns, such as local rotation, local translation, local scale, and local distortion, illustrating, thereby, that the GAN can model practical variations among vein patterns in images during the capturing process. By contrast, low-quality images show a significant distribution difference from the original images, or there is no difference among the vein patterns in the generated images. Differently, the performance of the GAN is related to the performance of downstream tasks. For example, a good GAN is capable of generating images that can improve the performance of downstream classifiers. By contrast, the GAN with poor performance implies that the performance of the classifier cannot be improved by leveraging the generated images for classifier training. Usually, the image generation is not directly related to downstream tasks, such as classification. Thus, it is possible that the classifier trained on the synthesized images with high quality cannot improve the performance for classification tasks.

For biometric recognition, it is important to preserve identity information for the synthesized biometric images. However, the adversarial loss in traditional GAN only makes the generator produce samples according to the target data distribution without any assurance that these generated samples will retain identity information. Different from traditional GANs, conditional DCGANs (cDCGANs) can produce training samples for each class conditioned on additional information, such as class labels, images, or texts. Conditional DCGANs (cDCGANs) have been recently widely applied in computer vision tasks [91]–[94] to generate images with conditional variables. Similarly, our AdveinAU generates the images based on such cDCGAN, where the class label is taken as the input of the generator and discriminator to preserve identity information for the synthesized vein images. In addition, to preserve identity information within the generated vein images, the cosine distance and a teacher model are incorporated into the objective loss of our model. For example, if the generated samples are far from the original samples for a given class, the loss of the objective function will increase, which enables

TABLE VIII: Results of ablation experiments on dataset A.

Classifier	Basic Advein$\mathbb{AU}$	Basic Advein$\mathbb{AU}$ + *cosine* similarity	Basic Advein$\mathbb{AU}$ + EMA	Basic Advein$\mathbb{AU}$ + *cosine* similarity + EMA
ResNet	98.01	98.17	98.47	98.80
VGG	97.87	98.04	98.17	98.90
FV-CNN	98.94	99.04	99.00	99.24
PV-CNN	98.57	98.74	98.84	98.84
FVRAS-Net	97.34	97.17	97.77	97.47
Lightweight CNN	96.97	97.17	97.77	97.84
Vit	98.77	98.97	98.90	99.03

TABLE IX: Results of ablation experiments on dataset B.

Classifier	Basic Advein$\mathbb{AU}$	Basic Advein$\mathbb{AU}$ + *cosine* similarity	Basic Advein$\mathbb{AU}$ + EMA	Basic Advein$\mathbb{AU}$ + *cosine* similarity + EMA
ResNet	93.08	93.62	93.90	94.15
VGG	91.90	92.63	93.85	94.13
FV-CNN	93.63	94.37	94.71	95.87
PV-CNN	93.90	94.37	94.71	94.83
FVRAS-Net	85.57	85.85	86.58	87.13
Lightweight CNN	92.05	93.65	92.78	93.77
Vit	85.55	85.70	86.12	87.60

TABLE X: Results of ablation experiments on dataset C.

Classifier	Basic Advein$\mathbb{AU}$	Basic Advein$\mathbb{AU}$ + *cosine* similarity	Basic Advein$\mathbb{AU}$ + EMA	Basic Advein$\mathbb{AU}$ + *cosine* similarity + EMA
ResNet	93.66	94.57	95.11	95.29
VGG	93.21	94.57	95.02	96.20
FV-CNN	92.84	93.84	94.11	94.11
PV-CNN	95.11	95.74	96.01	96.20
FVRAS-Net	92.22	92.41	92.50	93.32
Lightweight CNN	89.19	89.50	90.68	91.73
Vit	92.03	92.57	92.93	93.39

the model to reduce the difference by optimization so as to preserve the identity information of the generated images. Fig. 7 shows the images of four classes generated by our approach. The experimental results demonstrate that our approach is capable of producing high-quality images for different classes. In other words, the generated images by our approach preserve the identity information of each class. Furthermore, the identification results (Table V, Table VI, and Table VII) support such a conclusion.

G. Ablation experiments

In Advein$\mathbb{AU}$, two regularization terms, namely *cosine* similarity and EMA model, are employed to avoid collapsing the inherent meaning of images. In this section, we highlight the experiments conducted to evaluate the performance of each teacher with respect to the classifier performance improvement. To facilitate the description, we remove the *cosine* similarity and EMA model from Advein$\mathbb{AU}$, and the resulting approach is denoted as basic Advein$\mathbb{AU}$. Then, we incorporate the *cosine* similarity into the basic Advein$\mathbb{AU}$, which is denoted as basic Advein$\mathbb{AU}$ + *cosine* similarity. Likewise, the basic Advein$\mathbb{AU}$ with EMA model are presented as basic Advein$\mathbb{AU}$ + EMA model, and the basic Advein$\mathbb{AU}$ with EMA model and *cosine* similarity is denoted as basic Advein$\mathbb{AU}$ + EMA model + *cosine* similarity, namely Advein$\mathbb{AU}$. The classification accuracy of these various approaches on dataset A, dataset B, and dataset C is illustrated in Tables VIII, IX, and X, respectively. The experimental results on the three datasets show that the combination of the *cosine* similarity and the EMA model allows to achieve higher identification accuracy compared to a single regularization term.

In our model, cosine similarity is employed as a regularization term to control the quality of synthesized images, which may be important to keep the identity for each class for a conditional generation model. To further investigate this aspect, we select the original images and synthesized images of five classes on three databases to train our model. Then, we compute the average cosine similarity between the original images and synthesized ones for each class. From the experimental results listed in Table XI, we can see that the cosine scores on the three databases are very small, which implies that, for each class, the vein patterns in the synthesized images have similar distribution with the ones in the original images. In other words, the synthesized images for each class show good fidelity to the class identity for the class conditional variable.

TABLE XI: *Cosine* similarity of our approach on three datasets.

Class Number	Dataset A	Dataset B	Dataset C
1	0.0080	0.0478	0.0117
2	0.0069	0.0462	0.0058
3	0.0062	0.0453	0.0090
4	0.0068	0.0298	0.0068
5	0.0069	0.0269	0.0087

IV. Conclusion

This paper proposes a novel adversarial learning-based data augmentation approach, named Advein$\mathbb{AU}$, that is able to optimize target palm vein classifiers and the latent variable search loss. First, a cDCGAN is trained to learn the real data distribution. Then, the resulting generator is combined with the target task to obtain adversarial Advein$\mathbb{AU}$, which is trained in an adversarial way. The sample search policy attempts to increase the training loss of a classifier by generating adversarial samples while the target network learns more robust features from harder examples to improve classifier performance. The resulting classifier is employed for vein classification. The experimental results on three public datasets show that the proposed approach generates more realistic and diverse adversarial palm-vein samples and significantly improves the classifier identification accuracy, thereby achieving state-of-the-art level performance.

In our work, we perturb the latent variables instead of the vein images to generate hard samples for vein classifier training. Some research works have demonstrated that deep learning algorithms were susceptible to attacks by adding crafted adversarial perturbations to legitimate samples as input. Such perturbations are often small in magnitude and are imperceptible to human eyes, but they can lead to severe misclassification of legitimate samples at inference time. In future work, we will investigate a network to defend against adversarial perturbation attacks.

References

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *CVPR*, pp. 586–591, 1991.

[2] A. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 302–314, 1997.

[3] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[4] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1816–1829, 2017.

[5] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.

[6] T. K. Perrachione, S. N. Del Tufo, and J. D. Gabrieli, "Human voice recognition depends on language ability," *Science*, vol. 333, no. 6042, pp. 595–595, 2011.

[7] D. Menotti, G. Chiachia, A. Pinto, W. Robson Schwartz, H. Pedrini, A. Xavier Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 864–879, 2015.

[8] H. Qin, M. A. El-Yacoubi, Y. Li, and C. Liu, "Multi-scale and multi-direction gan for cnn-based single palm-vein identification," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2652–2666, 2021.

[9] Y. Zhou and A. Kumar, "Human identification using palm-vein images," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1259–1274, 2011.

[10] P. Harsha, R. Kanimozhi, and C. Subashini, "A real time embedded system of vein used for authentication in teller machine," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, pp. 400–405, 2013.

[11] R. R. Tallam, S. S. Temgire, and R. M. Zirange, "Finger vein recognition system using image processing," *International Journal of Electrical, Electronics and Data Communication*, vol. 2, pp. 64–68, 2014.

[12] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[13] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2228–2244, 2012.

[14] J. Yang and Y. Shi, "Towards finger-vein image restoration and enhancement for finger-vein recognition," *Information Sciences*, vol. 268, pp. 33–52, 2014.

[15] E. C. Lee and K. R. Park, "Image restoration of skin scattering and optical blurring for finger vein recognition," *Optics and Lasers in Engineering*, vol. 49, no. 7, pp. 816–828, 2011.

[16] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 8, pp. 1185–1194, 2007.

[17] ——, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Machine Vision and Applications*, vol. 15, no. 4, pp. 194–203, 2004.

[18] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision classification," *Pattern Recognition*, vol. 71, pp. 158–172, 2017.

[19] W. Song, T. Kim, H. C. Kim, J. H. Choi, H.-J. Kong, and S.-R. Lee, "A finger-vein verification system using mean curvature," *Pattern Recognition Letters*, vol. 32, no. 11, pp. 1541–1547, 2011.

[20] P. Gupta and P. Gupta, "An accurate finger vein based verification system," *Digital Signal Processing*, vol. 38, pp. 43–52, 2015.

[21] H. Qin, L. Qin, and C. Yu, "Region growth–based feature extraction method for finger-vein recognition," *Optical Engineering*, vol. 50, no. 5, pp. 057 208–057 208, 2011.

[22] T. Liu, J. Xie, W. Yan, P. Li, and H. Lu, "An algorithm for finger-vein segmentation based on modified repeated line tracking," *The Imaging Science Journal*, vol. 61, no. 6, pp. 491–502, 2013.

[23] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li, "Finger-vein authentication based on wide line detector and pattern normalization," *ICPR*, pp. 1269–1272, 2010.

[24] Z. Zhang, S. Ma, and X. Han, "Multiscale feature extraction of finger-vein patterns based on curvelets and local interconnection structure neural network," *ICPR*, vol. 4, pp. 145–148, 2006.

[25] C.-B. Yu, H.-F. Qin, Y.-Z. Cui, and X.-Q. Hu, "Finger-vein image recognition combining modified hausdorff distance with minutiae feature matching," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 1, no. 4, pp. 280–289, 2009.

[26] W. Kang and Q. Wu, "Contactless palm vein recognition using a mutual foreground-based local binary pattern," *IEEE Transactions on Information Forensics & Security*, vol. 9, no. 11, pp. 1974–1985, 2014.

[27] B. A. Rosdi, h. W. Shing, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, p. 11357– 1137, 2011.

[28] H. C. Lee, B. J. Kang, E. C. Lee, and R. P. Kang, "Finger vein recognition using weighted local binary pattern code based on a support vector machine," *Journal of Zhejiang University Science C*, vol. 11, no. 7, pp. 514–524, 2010.

[29] K. Y. H. Liu C, "An efficient finger-vein extraction algorithm based on random forest regression with efficient local binary patterns," *IEEE International Conference on Image Processing*, p. 3141–3145, 2015.

[30] 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. 15 048–15 067, 2013.

[31] L. Yang, G. Yang, Y. Yin, and X. Xi, "Finger vein recognition with anatomy structure analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[32] Y. Zhang, W. Li, L. Zhang, X. Ning, L. Sun, and Y. Lu, "Adaptive learning gabor filter for finger-vein recognition," *IEEE Access*, vol. 7, pp. 159 821–159 830, 2019.

[33] H. Wang, M. Du, J. Zhou, and L. Tao, "Weber local descriptors with variable curvature gabor filter for finger vein recognition," *IEEE Access*, vol. 7, pp. 108 261–108 277, 2019.

[34] G. Yang, R. Xiao, and Y. Yin, "Finger vein recognition based on personalized weight maps," *Neurocomputing*, vol. 13, no. 9, p. 12093–12112, 2013.

[35] X. Xi, L. Yang, and Y. Yin, "Learning discriminative binary codes for finger vein recognition," *Pattern Recognition*, vol. 66, pp. 26–33, 2017.

[36] C.-L. Lin and K.-C. Fan, "Biometric verification using thermal images of palm-dorsa vein patterns," *IEEE Transactions on Circuits and systems for Video Technology*, vol. 14, no. 2, pp. 199–213, 2004.

[37] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognition Letters*, vol. 141, pp. 61–67, 2021.

[38] J. D. Wu and C. T. Liu, "Finger-vein pattern identification using principal component analysis and the neural network technique," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5423–5427, 2011.

[39] Z. Gao, J. Cai, Y. Shi, L. Hong, F. Yan, and M. Zhang, "Integration of two-dimensional kernel principal component analysis plus twodimensional linear discriminant analysis with convolutional neural network for finger vein recognition." *Traitement du Signal*, vol. 38, no. 4, 2021.

[40] S. Damavandinejadmonfared, "Finger vein recognition using linear kernel entropy component analysis," in *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*. IEEE, 2012, pp. 249–252.

[41] G. Yang, X. Xi, and Y. Yin, "Finger vein recognition based on (2d)2 pca and metric learning," *Journal of Biomedicine and Biotechnology*, vol. 2012, no. 2, p. 324249, 2012.

[42] S. Elnasir and S. M. Shamsuddin, "Proposed scheme for palm vein recognition based on linear discrimination analysis and nearest neighbour classifier," in *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*. IEEE, 2014, pp. 67–72.

[43] K. Kapoor, S. Rani, M. Kumar, V. Chopra, and G. S. Brar, "Hybrid local phase quantization and grey wolf optimization based svm for finger vein recognition," *Multimedia Tools and Applications*, vol. 80, no. 10, pp. 15 233–15 271, 2021.

[44] Y. Xin, Z. Liu, H. Zhang, and H. Zhang, "Finger vein verification system based on sparse representation," *Applied optics*, vol. 51, no. 25, pp. 6252–6258, 2012.

[45] S. Shazeeda and B. A. Rosdi, "Finger vein recognition using mutual sparse representation classification," *IET Biometrics*, vol. 8, no. 1, pp. 49–58, 2019.

[46] L. Yang, G. Yang, K. Wang, F. Hao, and Y. Yin, "Finger vein recognition via sparse reconstruction error constrained low-rank representation," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4869–4881, 2021.

[47] K. Itqan, A. Syafeeza, F. Gong, N. Mustafa, Y. Wong, and M. Ibrahim, "User identification system based on finger-vein patterns using convolutional neural network," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 5, pp. 3316–3319, 2016.

[48] J. Wang, G. Wang, and M. Zhou, "Bimodal vein data mining via cross-selected-domain knowledge transfer," *IEEE Transactions on Information Forensics & Security*, vol. PP, no. 99, pp. 1–1, 2017.

[49] R. Das, E. Piciucco, E. Maiorana, and P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 360–373, 2018.

[50] H. Qin and M. A. El-Yacoubi, "Deep representation for finger-vein image-quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1677–1693, 2017.

[51] W. Kim, J. M. Song, and K. R. Park, "Multimodal biometric recognition based on convolutional neural network by the fusion of finger-vein and finger shape using near-infrared (nir) camera sensor," *Sensors*, vol. 18, no. 7, p. 2296, 2018.

[52] W. Yang, C. Hui, Z. Chen, J.-H. Xue, and Q. Liao, "Fv-gan: Finger vein representation using generative adversarial networks," *IEEE Transactions on Information Forensics & Security*, vol. 14, no. 9, pp. 2512–2524, 2019.

[53] G. Wang, C. Sun, and A. Sowmya, "Multi-weighted co-occurrence descriptor encoding for vein recognition," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 375–390, 2019.

[54] W. Kang, H. Liu, W. Luo, and F. Deng, "Study of a full-view 3d finger vein verification technique," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1175–1189, 2019.

[55] W. Yang, W. Luo, W. Kang, Z. Huang, and Q. Wu, "Fvras-net: An embedded finger-vein recognition and antispoofing system using a unified cnn," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 11, pp. 8690–8701, 2020.

[56] K. J. Noh, J. Choi, J. S. Hong, and K. R. Park, "Finger-vein recognition based on densely connected convolutional network using score-level fusion with shape and texture images," *IEEE Access*, vol. 8, pp. 96 748–96 766, 2020.

[57] Z. Pan, J. Wang, G. Wang, and J. Zhu, "Multi-scale deep representation aggregation for vein recognition," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1–15, 2020.

[58] R. S. Kuzu, E. Piciucco, E. Maiorana, and P. Campisi, "On-the-fly finger-vein-based biometric recognition using deep neural networks," *IEEE Transactions on information Forensics and Security*, vol. 15, pp. 2641–2654, 2020.

[59] W. Jia, J. Gao, W. Xia, Y. Zhao, H. Min, and J.-T. Lu, "A performance evaluation of classic convolutional neural networks for 2d and 3d palmprint and palm vein recognition," *International Journal of Automation and Computing*, vol. 18, no. 1, pp. 18–44, 2021.

[60] J. Shen, N. Liu, C. Xu, H. Sun, Y. Xiao, D. Li, and Y. Zhang, "Finger vein recognition algorithm based on lightweight deep convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2021.

[61] J. Huang, M. Tu, W. Yang, and W. Kang, "Joint attention network for finger vein authentication," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[62] Y.-Y. Fanjiang, C.-C. Lee, Y.-T. Du, and S.-J. Horng, "Palm vein recognition based on convolutional neural network," *Informatica*, vol. 32, no. 4, pp. 687–708, 2021.

[63] F. O. Babalola, Y. Bitirim, and Ö. Toygar, "Palm vein recognition through fusion of texture-based and cnn-based methods," *Signal, Image and Video Processing*, vol. 15, no. 3, pp. 459–466, 2021.

[64] H. Lu, Y. Li, C. Zhao, W. Liu, Y. Li, and N. Ma, "A novel finger-vein recognition approach based on vision transformer," in *International Conference on Frontiers of Electronics, Information and Computation Technologies*, 2021, pp. 1–6.

[65] Y. Zhang, W. Li, L. Zhang, X. Ning, L. Sun, and Y. Lu, "Agcnn: adaptive gabor convolutional neural networks with receptive fields for vein biometric recognition," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 12, p. e5697, 2022.

[66] J. Huang, W. Luo, W. Yang, A. Zheng, F. Lian, and W. Kang, "Fvt: Finger vein transformer for authentication," *IEEE Transactions on Instrumentation and Measurement*, 2022.

[67] H. Ren, L. Sun, J. Guo, C. Han, and F. Wu, "Finger vein recognition system with template protection based on convolutional neural network," *Knowledge-Based Systems*, vol. 227, p. 107159, 2021.

[68] P. Zhao, S. Zhao, L. Chen, W. Yang, and Q. Liao, "Exploiting multi-perspective driven hierarchical content-aware network for finger vein verification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[69] W.-F. Ou, L.-M. Po, C. Zhou, P.-F. Xian, and J.-J. Xiong, "Gan-based inter-class sample generation for contrastive learning of vein image representations," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 2, pp. 249–262, 2022.

[70] G. Wang, W. Kang, Q. Wu, Z. Wang, and J. Gao, "Generative adversarial network (gan) based data augmentation for palmprint recognition," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, 2018, pp. 1–7.

[71] J. Zhang, Z. Lu, M. Li, and H. Wu, "Gan-based image augmentation for finger-vein biometric recognition," *IEEE Access*, vol. 7, pp. 183 118–183 132, 2019.

[72] B. Hou and R. Yan, "Triplet-classifier gan for finger-vein verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[73] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[74] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 5767–5777.

[75] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[77] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[78] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[79] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An online system of multispectral palmprint verification," *IEEE transactions on instrumentation and measurement*, vol. 59, no. 2, pp. 480–490, 2009.

[80] Z. Lin, Z. Cheng, Y. Shen, and D. Wang, "On the vulnerability of palm vein recognition and a new large-scale contactless palmvein dataset," *Symmetry*, vol. 10, no. 4, p. 78, 2018.

[81] P. Tome and S. Marcel, "On the vulnerability of palm vein recognition to spoofing attacks," in *2015 International Conference on Biometrics (ICB)*, 2015, pp. 319–325.
[82] H. Qin, M. A. El Yacoubi, J. Lin, and B. Liu, "An iterative deep neural network for hand-vein verification," *IEEE Access*, vol. 7, pp. 34 823–34 837, 2019.
[83] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.
[84] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.
[85] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.
[86] A. Obukhov and M. Krasnyanskiy, "Quality assessment method for gan based on modified metrics inception score and fréchet inception distance," in *Software Engineering Perspectives in Intelligent Systems: Proceedings of 4th Computational Methods in Systems and Software 2020, Vol. 1 4*. Springer, 2020, pp. 102–114.
[87] M. J. Chong and D. Forsyth, "Effectively unbiased fid and inception score and where to find them," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6070–6079.
[88] Y. Yu, W. Zhang, and Y. Deng, "Frechet inception distance (fid) for evaluating gans," *China University of Mining Technology Beijing Graduate School: Beijing, China*, 2021.
[89] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.
[90] J. Lee and M. Lee, "Fidgan: A generative adversarial network with an inception distance," in *2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2023, pp. 397–400.
[91] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5907–5915.
[92] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 1857–1865.
[93] K. Sricharan, R. Bala, M. Shreve, H. Ding, K. Saketh, and J. Sun, "Semi-supervised conditional gans," *arXiv preprint arXiv:1708.05789*, 2017.
[94] B. Dai, S. Fidler, R. Urtasun, and D. Lin, "Towards diverse and natural image descriptions via a conditional gan," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2970–2979.