

LSFM: Light Style and Feature Matching for Efficient Cross-Domain Palmprint Recognition

Song Ruan[✉], Yantao Li[✉], Senior Member, IEEE, and Huafeng Qin[✉]

Abstract—The exceptional feature extraction capabilities of deep neural networks (DNNs) have significantly advanced palmprint recognition. However, DNNs typically require training and testing data originate from the same distribution, which limits their practical applications. Moreover, existing unsupervised domain adaptation methods struggle to achieve high accuracy with efficiency. To address these challenges, we propose LSFM, an efficient Light Style and Feature Matching method that enhances palmprint recognition performance in cross-domain scenarios with fewer resources. Specifically, we develop an efficient style transfer model to mitigate domain shifts at the pixel level. We then align features across multiple task-specific layers in high dimensional space to reduce domain discrepancies, further improving cross-domain performance. Finally, we evaluate the effectiveness of the proposed LSFM through extensive experiments on two public multi-domain palmprint databases. The experimental results demonstrate that LSFM achieves superior performance with significantly reduced resource consumption, improving average accuracy to 94.87% and lowering the average equal error rate to 1.46%, while saving over 80% of resources.

Index Terms—Palmprint recognition, unsupervised domain adaptation, light style, feature matching.

I. INTRODUCTION

BIOMETRIC features are widely used for establishing individual identity, with applications in security systems, mobile payments, and more. Among these, palmprint recognition has garnered significant attention in recent years due to its convenience of acquisition and high accuracy [1]. Additionally, it offers hygienic advantages and is difficult to forge, since image capture can occur without physical contact. However, key steps in palmprint recognition, including image capture, pre-processing, and recognition, pose significant challenges for machines. Various factors, such as illumination, capturing

Received 18 April 2024; revised 22 August 2024; accepted 3 October 2024. Date of publication 9 October 2024; date of current version 17 October 2024. This work was supported in part by the National Science and Technology Major Project 2020AAA0107300, in part by the National Natural Science Foundation of China under Grant 62072061 and Grant 61976030, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZDK202100104, and in part by the National Natural Science Foundation of Chongqing under Grant CSTB2024NSCQ-MSX118. The associate editor coordinating the review of this article and approving it for publication was Dr. Fernando Alonso-Fernandez. (Corresponding authors: Yantao Li; Huafeng Qin.)

Song Ruan and Yantao Li are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: rsmung@cqu.edu.cn; yantaoli@cqu.edu.cn).

Huafeng Qin is with the National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China (e-mail: qinhufengfeng@163.com).

Digital Object Identifier 10.1109/TIFS.2024.3476978

devices, and user behavior, can affect palmprint images. As a result, achieving robust and reliable personal identification using palmprint images has become a critical area of research. The rapid advancement of deep learning technologies, driven by the evolution of high-performance devices and the increasing availability of data, has further propelled this field. Due to the superior feature extraction capabilities of deep learning, numerous palmprint recognition methods have been proposed, yielding impressive results [1]. However, pre-trained deep learning models often suffer from performance degradation when applied to new, related domains, known as target domains [2]. This is common due to the dynamic nature of data capture in natural environments, leading to inevitable domain discrepancies. These differences between the training domain (source domain) and the testing domain (target domain) are commonly referred to as domain shifts [3]. In addition, labeling data is both time-consuming and costly, making it impractical to collect a sufficient number of labeled samples from the target domain for model fine-tuning.

To address these issues, researchers have introduced various unsupervised domain adaptation (UDA) methods, aiming to transfer knowledge from a labeled source domain to an unlabeled target domain. These UDA methods can be broadly categorized into three groups: divergence-based methods, adversarial learning-based methods, and pseudo-labeling algorithm-based methods. Divergence-based methods [3], [4] utilize a divergence measure to quantify discrepancies among features from different domains. By minimizing this divergence, these methods encourage the generation of domain-invariant features, thereby enhancing model performance in the target domain. Adversarial learning-based methods [3], [5] leverage the training framework of generative adversarial networks (GANs) [6] to align features across domains, though they face challenge related to training stability. Pseudo-labeling algorithm-based methods [3], [7] iteratively retrain models using confident pseudo-labels to reduce the domain gap, despite the potential noise in these labels. Specifically for palmprint recognition, the authors in [8] employ adversarial learning and divergence metrics to alleviate domain shifts. However, these methods primarily focus aligning feature distributions, thereby limiting overall performance. In [9], the authors utilize style-transfer models to bridge the gap between different domains at the pixel level. Furthermore, the authors of [10] and [11] adapt features at both the pixel and feature levels. Nonetheless, these methods often overlook resource constraints, which are particularly relevant in palmprint recognition applications, hindering practical deployment.

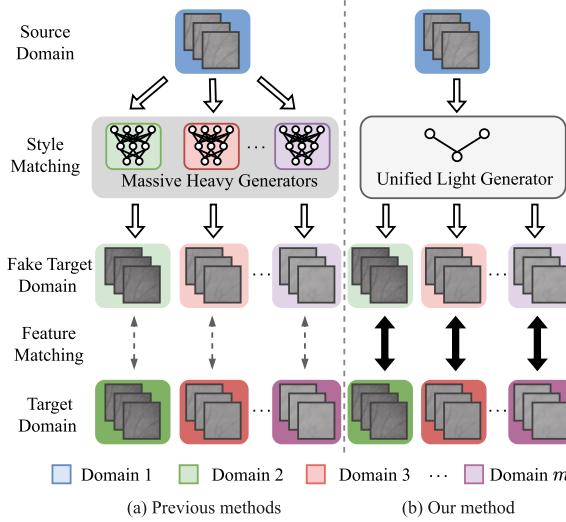


Fig. 1. Comparison between previous methods and our method. (a) previous optimal unsupervised domain adaptation methods for palmprint recognition include a style-matching stage that relies on large, resource-intensive generators and a less effective feature-matching stage. (b) Our method leverages a unified light generator for style alignment and a more robust feature alignment strategy, significantly improving cross-domain palmprint recognition performance.

We aim to develop a lightweight method for mitigating domain shifts in palmprint recognition, which poses two key challenges. *The first challenge is how to achieve pixel-level alignment using a lightweight style-transfer model.* Current state-of-the-art style-transfer models [12], [13], [14] are designed to learn relationships between two distinct domains in a single training process, thereby limiting their compatibility in scenarios involving multiple diverse domains. For example, when transforming between m palmprint domains, each domain requires $m - 1$ generators, as illustrated in Fig. 1(a). As a result, $m(m - 1)$ generators must be trained and deployed. Furthermore, models such as [15], [16], and [17] can generate images with various styles but are not specifically designed for image-to-image translation tasks. While StarGAN [18], [19] can transform source domain images into various target domain images using a single model, the ResNet blocks [20] in its generators are too computationally intensive for resource-constrained environments. Meanwhile, TinyStarGAN [21] reduces StarGAN's computational load but introduces additional complexity through a knowledge-transfer procedure, and its use of the MobileNet block [22] in the generator is not the most advanced lightweight structure, limiting its performance. Consequently, there is a strong need for further research into transforming palmprint images across different domains using lightweight models. *The second challenge is how to effectively learn domain-invariant features.* The authors in [8] and [11] employ adversarial learning to minimize feature gaps, but achieving an optimal model within this framework is challenging, limiting performance. The authors of [10] and [23] focus only on aligning the distribution of the last layer in domain-specific feature extractors, hindering overall model improvement. Since the final output distributions for different domains are inherently diverse, forcing them to be similar can lead to negative adaptation effects.

To address these challenges, we present LSFm, an efficient Light Style and Feature Matching method for cross-domain palmprint recognition in resource-constrained scenarios, as illustrated in Fig. 1(b). Specifically, LSFm is able to effectively reduce domain gaps using a lightweight style-transfer technique, LightStarGAN, which significantly lowers resource costs during both training and deployment stages. With LightStarGAN, image-to-image translation tasks can be performed using a single unified model. By transforming source domain images into pseudo-target images, domain gaps are minimized, making alignment easier. Moreover, to achieve more robust feature matching across domains, we address the challenges posed by the difficulty of achieving optimal results with adversarial learning and the limitations of single-layer alignment. Instead, we focus on minimizing distribution divergences across multiple task-specific layers.

The main contributions of this work are summarized as follows:

- 1) We propose LSFm to achieve efficient cross-domain palmprint recognition in resource-constrained scenarios. Adapting source models to target domains with minimal costs is challenging but essential for practical applications.
- 2) We introduce a lightweight, unified style-transfer model, LightStarGAN, to minimize domain discrepancies at the pixel level by transforming the style of images across multiple domains. Moreover, we align the features of pseudo-target images and target images across multiple task-specific layers, thereby significantly enhancing the performance of the palmprint classifier.
- 3) Extensive experiments on two public multi-domain palmprint databases validate the superior performance of our light cross-domain palmprint recognition method, demonstrating that it outperforms competitors, including vanilla UDA methods and those specifically designed for palmprint recognition.

The remainder of this work is organized as follows: Section II introduces the related work. Section III details the proposed LSFm. The experimental results and corresponding analyses are present in Section IV. Finally, we conclude this work in Section V.

II. RELATED WORK

A. Palmprint Recognition

Palmprint identification is a challenging task that involves several intricate processes, including palmprint acquisition, feature extraction, and personal identification. Currently, there are three types of palmprint images: low-resolution palmprint images, high-resolution palmprint images, and 3D (three-dimensional) palmprint images [24]. Feature extraction and identity recognition methods can be broadly categorized into two types: traditional methods and DNNs-based methods.

Traditional methods for palmprint recognition primarily rely on visible features such as principal lines, wrinkles, and texture. Several studies focus on low-resolution palmprint images. Zhang et al. [25] introduced a two-dimensional Gabor phase encoding method to extract texture features from contact-based palmprint images. To fully utilize the orientation information of palmprint lines, Kong and Zhang [26] proposed a competitive coding scheme along with an angular

matching method. Huang et al. [27] developed a modified finite Radon transform to extract principal lines more explicitly and efficiently. In recent years, Fei et al. [28] employed a local discriminant direction binary pattern (LDDBP) to comprehensively represent directional features. Palma et al. [29] presented a positive linear dynamical system for palmprint recognition with principal lines. Xu et al. [30] enhanced the discriminative power and robustness of competitive coding by weighting the orientation information of neighboring areas. Zhao and Zhang [31] proposed a least square regression model to extract salient palmprint features. Additionally, they combined the complete local direction feature (CLDF) with the salient convolution difference feature (SCDF) to achieve more discriminative feature extraction [32]. Yang et al. [33] explored previously overlooked second-order textures, achieving excellent results by combining them with first-order textures.

Some works focus on high-resolution images, which are typically used in high-security forensic applications. Jain and Feng [34] leveraged a fixed-length minutiae descriptor to extract minutiae features when matching latent palmprints, often recovered from crime scenes with full palmprint images. Considering the abundance of minutiae features in high-resolution palmprints, Liu et al. [35] developed a coarse-to-fine matching method based on minutiae clustering. Dai et al. [36], using quantitative statistics of palmprint features, proposed a segment-based matching and fusion method, which demonstrated excellent performance on their high-resolution palmprint database. Additionally, a quality-based adaptive orientation field estimation method was proposed for high-resolution palmprint recognition based on multiple palmprint features [37]. Hussein et al. [38] employed a statistical gray-level co-occurrence matrix (GLCM) to extract features from high-resolution palmprint images.

Several studies focus on 3D palmprint recognition. Fei et al. [39] introduced a compact surface type (CST) to capture the 3D structure features of palmprints. Iula et al. leveraged principal line depth information for palmprint recognition based on a 3D ultrasound system [40], and later, they generated a 3D template by analyzing 2D features at different under-skin depths [41].

DNNs-based methods have achieved outstanding results in palmprint recognition. We begin by reviewing low-resolution palmprint recognition. Dian and Dongmei [42] explored an eight-layer convolutional neural network (CNN) to extract palmprint features. Genovese et al. [43] used a CNN to extract discriminative palmprint descriptors based on Gabor responses and principal component analysis (PCA). Zhong and Zhu [44] applied a centralized large margin cosine loss to train a CNN, achieving superior results. They also developed a deep hashing network (DHN) for palmprint recognition [45]. Zhou et al. [46] proposed a biologically inspired method for robust palmprint recognition using convolutional layers. Liu and Kumar [47] introduced a faster R-CNN for palmprint region detection and used a CNN with a soft-shifted triplet loss function for feature extraction. Zhao et al. [48] designed a discriminative CNN to extract high-level features from limited palmprint data. Zhao et al. [49] proposed a joint constrained least-square regression (JCLSR) framework to address the undersampling

problem with the aid of a CNN. More recently, Liang et al. [50] utilized 3D convolutions and a modified Gabor filter to extract contactless palmprint features. Grosz et al. [51] developed a mobile-based end-to-end palmprint recognition system that used a CNN and a vision transformer to extract local and global features. Shao et al. [52] proposed a few-shot palmprint recognition method using a graph neural network, and later designed a meta-siamese network (MSN) for small-sample palmprint recognition [53].

Fewer works specialized in high-resolution and 3D palmprint recognition. Liu et al. [54] were the first to introduce DNNs for minutiae extraction from high-resolution palmprint images. Korichi et al. [55] utilized a series of pre-trained DNNs and employed transfer learning to build a 2D/3D palmprint recognition system.

By leveraging supervised learning, DNNs excel in feature extraction and recognition. However, current deep learning methods often overlook the associated resource costs, and their performance tends to diminish when testing images differ slightly from the training images.

B. Unsupervised Domain Adaptation

Given the dynamic nature of real-world environments, it is essential to explore how pre-trained models can be adapted from source training data to target testing data. Existing unsupervised domain adaptation (UDA) methods have been developed to address this challenge and can be classified into four main categories: discrepancy-based methods, adversarial learning-based methods, pseudo labeling algorithm-based methods, and UDA methods for palmprint recognition.

Discrepancy-based methods focus on minimizing discrepancies between different domains using specific metrics. For example, the DDC model [56] introduced an adaptation layer that utilized maximum mean discrepancy to learn domain-invariant features. Extensions of this approach included DAN [4], which altered the alignment position, and JAN [57], which proposed a joint maximum mean discrepancy criterion.

With the advancement of generative adversarial networks (GANs) [6], *adversarial learning-based methods* emerged as a means to extract domain-invariant features. Typically, a discriminator distinguishes features from different domains, while a feature extractor generates domain-invariant features to confuse the discriminator [5], [58]. Ganin et al. [58] introduced the gradient reversal layer (GRL) to implicitly implement adversarial learning. Tzeng et al. [5] presented ADDA, which split the optimization process and introduced the untied weight sharing. Other methods, like MCD and CGDM, use two diverse classifiers to challenge the feature extractor, with MCD [59] aiming to generate dissimilar predictions and CGDM [60] focusing on minimizing the gradient discrepancy between source and target data. However, adversarial learning-based methods often struggle with achieving balanced and stable training.

Pseudo labeling algorithm-based methods address the lack of labels in the target domain. Xie et al. [61] introduced pseudo labels for target data to mitigate this issue, aligning the centroid of each class in both the source and target

domains to correct biased pseudo labels. Zhang et al. [62] utilized pseudo-labeled target data to re-train and enhance their collaborative and adversarial network (CAN), achieving improved performance. Zou et al. [7] designed the confidence regularized self-training (CRST) method to avoid assigning overconfident labels to incorrect classes during pseudo-label generation. Although these methods employ various pseudolabeling algorithms, the presence of inevitable noise in the pseudo-labels tends to limit their performance.

UDA methods for palmprint recognition [8], [9], [10] combine elements from the previous three categories to achieve significant results. For example, R-ADAH [8] integrated discrepancy-based and adversarial learning methods to learn domain invariant features. PalmGAN [9] utilized a style transfer model to generate labeled fake target palmprint images, reducing domain gaps and fine-tuning the pre-trained source model. However, the advanced JPFA method [10] required large models for multiple cross-domain scenarios, which was impractical for resource-constrained environments.

Existing methods struggle to balance resource costs with performance. Although [63] addresses resource-saving and fast adaptive inference, it only realizes adaptation at the feature level, limiting its effectiveness in enhancing cross-domain palmprint recognition. In contrast, our approach is specifically designed to address and mitigate these limitations.

III. METHOD

A. Notation and Problem Definition

The UDA problem in palmprint recognition involves two main steps: 1) training the source model on labeled source data, and 2) adapting the model to unlabeled target data. First, we train the source model using labeled source data, denoted as $D_S = \{(p_i^S, y_i^S)\}_{i=0}^{N_S}$. The source data consists of N_S palmprints from the source domain, represented as $p_i^S \sim P_S$. Each palmprint p_i^S is associated with a corresponding label $y_i^S \sim Y_S$. The source model, comprising a feature extractor F_{ES} and a classifier H_S , learns a function $F_S : P_S \rightarrow Y_S$ to classify the source palmprints into K classes, using the loss function shown in Eq. (1):

$$\begin{aligned} \min_{\theta_{FES}, \theta_{HS}} & \mathcal{L}_{cls}(P_S, Y_S) \\ &= \frac{1}{N_S} \sum_{i=1}^{N_S} \mathcal{L}_{CE}\left(H_S\left(F_{ES}\left(p_i^S\right)\right), y_i^S\right), \end{aligned} \quad (1)$$

where $\mathcal{L}_{CE}(\cdot, \cdot)$ denotes the standard cross-entropy loss function, and θ_{FES} and θ_{HS} represent the parameters of F_{ES} and H_S , respectively.

Next, we denote the unlabeled target data as $D_T = \{(p_i^T)\}_{i=0}^{N_T}$. The target data comprise N_T palmprints from the target domain, but without corresponding labels. The target domain P_T is similar to the source domain P_S , though domain shift introduces some discrepancies. However, the label space for both domains is identical, i.e., $Y_T = Y_S$.

The goal of UDA is to enhance the performance of the pre-trained source model F_S in the target domain P_T by leveraging the labeled source data D_S and the unlabeled target data D_T .

B. LSFM Architecture

Existing deep learning-based cross-domain palmprint recognition methods often overlook resource constraints, which are critical in practical applications. Palmprint recognition systems are typically deployed on resource-limited hardware, so it is essential that the models used prioritize efficiency and lightweight design. To address this, we propose an efficient UDA method for palmprint recognition, as shown in Fig. 2, by developing the system closely aligned with practical applications. By incorporating specific style codes sc based on domain labels d , the generator can produce fake palmprint images across multiple domains. These style codes are derived either from a random noise vector z , processed through a mapping network M , or extracted from a style encoder SE using reference images. After mitigating pixel shifts through style transfer, features in task-specific layers are further aligned within the Reproducing Kernel Hilbert Space (RKHS). Specifically, features of target images are encouraged to resemble those of fake target images by reducing the Multiple Kernel Maximum Mean Discrepancy (MK-MMD), an alignment made more effective by the prior style transfer. This approach enables us to achieve outstanding results with a more resource-efficient solution. By employing LightStar-GAN, LSFM reduces the number of models, parameters, and computational demands across multiple domains. This approach eliminates the need to train numerous models for various domain pairs. While our method generates fake target images using both source and target images, it remains an unsupervised domain adaptation method because we do not utilize any target labels. We next detail the style matching and feature matching processes, respectively.

C. Style Matching

To reduce the domain gap at a low level, a style transfer model is commonly employed to transform source palmprint images into target palmprint images. After that, fake images with a similar style to the target images are generated. Notably, these fake images share the same labels as the source images, as they retain the same identity information. The performance of target models can then be enhanced through supervised fine-tuning using these labeled fake target images [9]. Nevertheless, some style transfer methods, such as [12] and [14], use a one-to-one transformation scheme, which typically requires two models for each cross-domain pair, as illustrated in Fig. 3(a). For example, the PolyU Multi-Spectral Palmprint Database [64] contains four domains, resulting in six cross-domain pairs. As a result, 12 generators must be trained and deployed. The total number of style transfer models N_{ST} scales quadratically with the number of domains, as shown in Eq. (2):

$$N_{ST} = 2 \times C_m^2 = m(m - 1), \quad (2)$$

where m is the number of domains owned by a specific database and C_m^2 is the combination number representing the number of cross-domain pairs. Consequently, existing style transfer methods are inefficient for implementation in

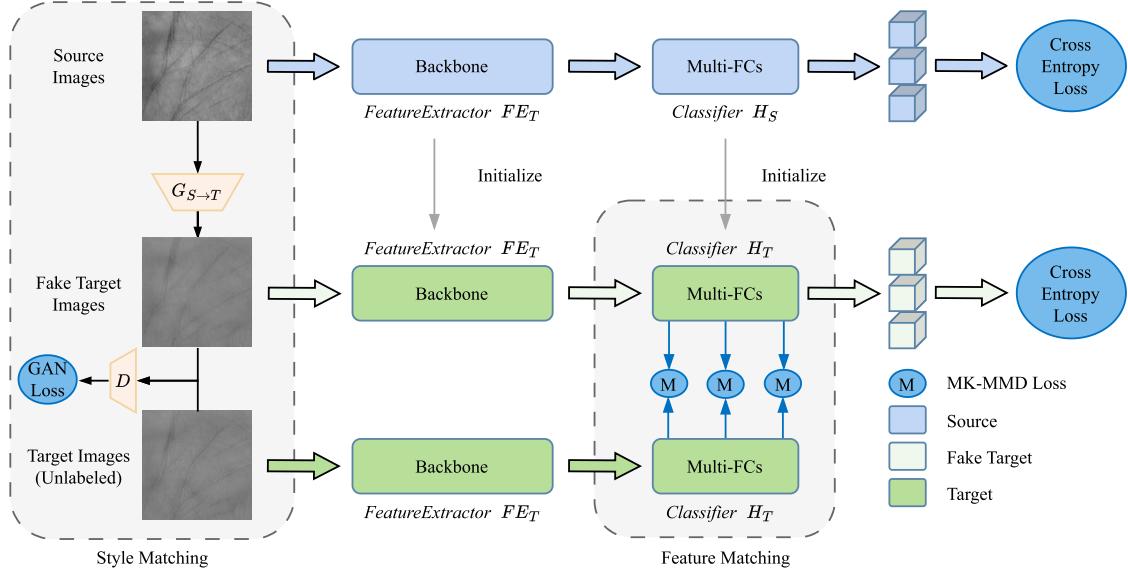


Fig. 2. Architecture of LSFM. Before adaptation, the source model is pre-trained on labeled source data. We then align target and source features through style transfer and feature matching. In style matching, we utilize a LightStarGAN for flexible style transfer, which requires fewer models and reduces computational costs. In feature matching, we guide the high-level layers to minimize the discrepancy between fake-target and target features using multiple MK-MMD regularizations. Note that the two sets of FE_T and HT share the same weights. The FeatureExtractor (Backbone) and classifier are general concepts for models composed of multiple convolutional (or transformer) layers and several fully connected layers.

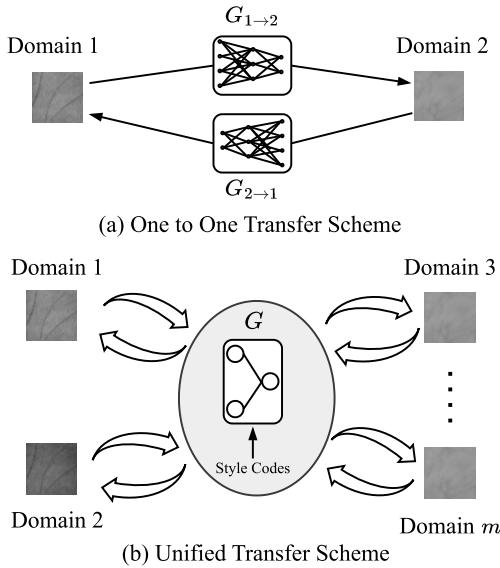


Fig. 3. Comparison between the inefficient style transfer scheme and the unified scheme. (a) Building two transfer models for each pair of palmprint image domains. (b) The unified model can transform styles among multiple domains.

resource-limited environments and struggle to manage multiple domains effectively, given the diverse nature of real-world scenarios. In addition to transfer schemes, the structures of transfer models is also crucial for achieving effectiveness. This implies that ResNet blocks [20] or transformer blocks [65] in existing transfer models may not be optimal for the style transfer of palmprint images. Building generators with these blocks incurs significant resource costs, as shown in Table I.

To address these significant issues, we introduce an efficient unified transfer scheme, as shown in Fig. 3(b). Based on this, we propose a lightweight unified style transfer model, as depicted in Fig. 4. This unified model, which

TABLE I
COMPARISON OF RESOURCE COSTS FOR DIFFERENT BLOCK-BASED GENERATORS

Resource Block Type	Storage Size (MB)	Params (Million)	Multi-Adds (Billion)
ShuffleNet v2 [66]	22.02	5.76	2.33
ResNet [20]	368.63	173.02	16.82
EfficientNet v2 [67]	124.92	32.69	6.56
ViT [65]	275.29	72.15	0.49
CSWinTransformer [68]	122.86	34.43	0.56

incorporates lightweight convolutional blocks, significantly reduces resource costs.

1) *Generator*: Inspired by StarGAN [18], [19], we employ a unified generator G to create fake images p^{FT} across different domains by integrating various style codes sc with source domain images p^S , as shown in Eq. (3):

$$p^{FT} = G(p^S, sc). \quad (3)$$

To reduce resource costs while maintaining the generator's effectiveness, we introduce light encoder layer and decoder layers, incorporating efficient shufflenetv2 blocks [66]. Specifically, the light encoders extract content features from the source images, which are then fused with specific style information in the light decoder layers. Additionally, we include shortcuts [69] between the encoder and decoder layers to preserve the content information of the source images, which is crucial for accurate palmprint recognition. As shown in Fig. 4(a), we use only four downsampling and upsampling layers to create a symmetric network that maintains the original image size while balancing the preservation of spatial details with computational efficiency. Fewer downsampling layers could retain more spatial information, but they would result in larger feature maps and higher computational costs. Conversely, more downsampling layers would reduce computational costs but at the expense of losing too much spatial detail.

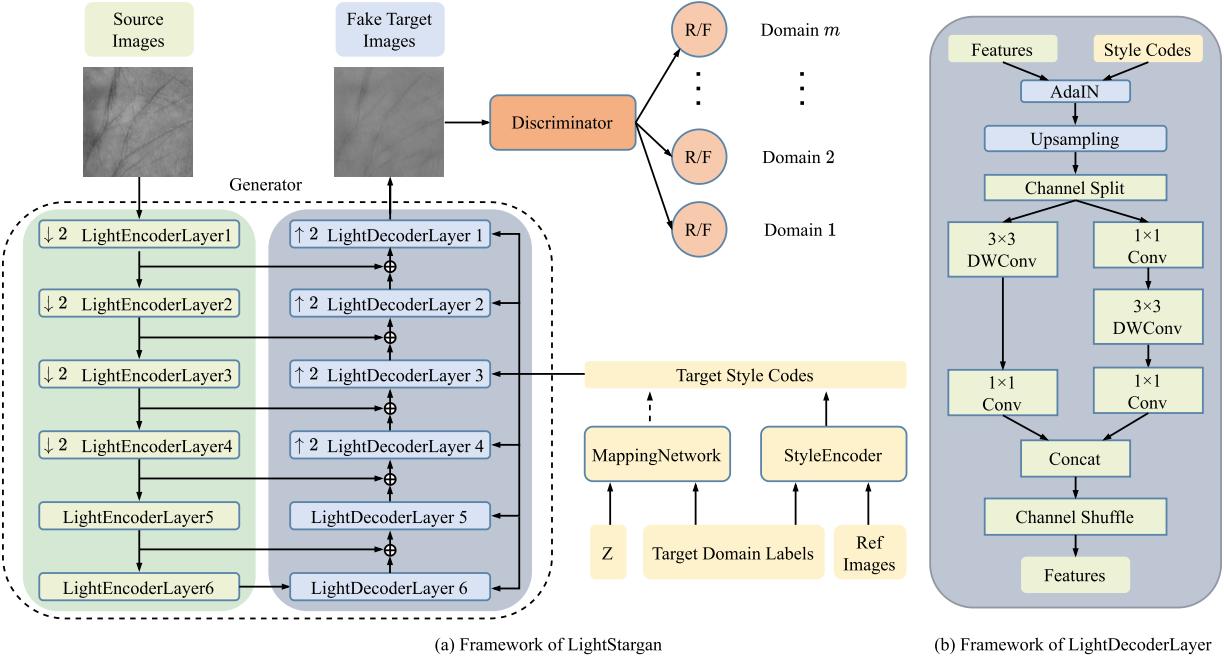


Fig. 4. Architecture of LightStarGAN. (a) Framework of the LightStarGAN. (b) Framework of the LightDecoderLayer. In the LightDecoder, the stride of DWConv is 1, and LightDecoders 5-6 lack Upsampling modules. Note that the LightDecoderLayer can be converted into a LightEncoderLayer by removing Style Codes, AdaIN, and Upsampling. Additionally, the strides of DWConv in LightEncoderLayers 1-4 and 5-6 are 2 and 1, respectively.

2) *Style Information*: The target style codes sc that contain style information are generated either by the mapping network M or the style encoder SE , as shown in Eq. (4):

$$sc = M(z, d) \text{ or } sc = SE(p^T, d). \quad (4)$$

The mapping network M transforms random noise vectors z sampled from the standard normal distribution D_z into style codes, while the style encoder SE captures style information from reference images p^T , with d representing the specific target domain label. The mapping network and style encoder follow the same architectures as outlined in [19]. In the inference stage, we use only the mapping network to generate style codes due to its lower resource consumption. However, in the training stage, both the mapping network and style encoder are trained simultaneously. This approach ensures that the style information in the style codes is effectively utilized by the generator. Consequently, the generator G can produce fake images guided by the specific style information.

3) *Discriminator*: To generate realistic fake images in target domains, the discriminator D functions as a critic. It assesses the likelihood that input images belong to a specific domain. For instance, if the input palmprint image p^{FT} is a fake image meant to belong to domain d , and the input palmprint image p^T is a real image from domain d , the discriminator aims to assign a low score to p^{FT} for being a real image from domain d . Conversely, the discriminator strives to assign a high score to p^T as a real image from domain d . Since the discriminator is not utilized during inference, but maintaining its modeling power is crucial, we retain its robust structure from [19].

4) *Training Objectives*: The entire training process of the generator G , mapping network M , style encoder SE , and

discriminator D forms a minimax game, as shown in Eq. (5):

$$\mathcal{L}_{gan} = \min_{\theta_G, \theta_M, \theta_{SE}} \max_{\theta_D} \left\{ \mathbb{E}_{p^T \sim D_T} [\log D(p^T)] + \mathbb{E}_{p^{FT} \sim D_{FT}} [\log (1 - D(p^{FT}))] \right\}, \quad (5)$$

where θ_G , θ_M , θ_{SE} , and θ_D indicate the parameters of G , M , SE , and D , respectively. D_T represents the data distribution of real images in the target domain, while D_{FT} denotes the distribution of fake target data.

Following StarGAN, we use a style regularization loss to ensure the generator effectively incorporates style information from the style codes, as shown in Eq. (6):

$$\mathcal{L}_{sty} = \mathbb{E}_{p^S, sc, d} [\|sc - SE(G(p^S, sc), d)\|_1]. \quad (6)$$

Additionally, a cycle consistency loss is applied to guarantee that the fake images contain the original content features from the source images, as depicted in Eq. (7):

$$\mathcal{L}_{cyc} = \mathbb{E}_{p^S, sc^T, sc^S} [\|p^S - G(G(p^S, sc^T), sc^S)\|_1], \quad (7)$$

where $sc^T = SE(p^T, d^T)$ and $sc^S = SE(p^S, d^S)$. sc^S and sc^T contain the style information corresponding to the specific target domain determined by d^S and d^T . Once the generator is pre-trained, labeled fake images can be generated by transforming the source images into the target domains.

Subsequently, the source model F_S , pre-trained on the source domain, can be fine-tuned using these labeled fake images. This fine-tuning process helps the model adapt to the target domain and can be formulated as Eq. (8):

$$\begin{aligned} & \min_{\theta_{FE_T}, \theta_{H_T}} \mathcal{L}_{SM}(P_{FT}, Y_S) \\ &= \frac{1}{N_S} \sum_{i=1}^{N_S} \mathcal{L}_{CE}(H_T(F_{ET}(p_i^{FT})), y_i^S), \end{aligned} \quad (8)$$

where θ_{FE_T} and θ_{H_T} represent the parameters of FE_T and H_T , respectively. p_i^{FT} indicates a generated palmprint image from the fake target domain P_{FT} .

D. Feature Matching

While the style transfer model helps alleviate pixel-level domain discrepancies, we believe that this alone is insufficient for successfully adapting the palmprint classifier from the source domain to the target domain. To achieve better performance, it is essential to align the data distribution of features across the task-specific layers H_T . Although previous works [10], [56] have acknowledged this issue and achieved performance improvements, they primarily focused on aligning features within a single layer, which proved inadequate for effective feature adaptation [4]. Aligning multiple hidden representations across task-specific layers is crucial for improving adaptation. Our approach emphasizes aligning the features of unlabeled target palmprint images p^T with those of fake target palmprint images p^{FT} . Since these images share similar styles, it is easier to align their features compared to directly aligning features between source and target domain images.

MMD [70] can be effectively utilized to measure distribution divergence between two sets of observed features without assuming any specific distributional form for the source and target features. MMD offers the numerical stability and computational efficiency. Moreover, its effectiveness has been demonstrated in previous works [4], [10], [23], [56], highlighting its strong performance in various applications. Consequently, we leverage MMD to measure the differences between these two feature distributions.

In the computation process, features are mapped to the Reproducing Kernel Hilbert Space \mathcal{H}_K using a mapping function $\phi(\cdot)$. The difference in expectations of features within this specialized space \mathcal{H}_K is then used to measure discrepancies between two different data distributions, as shown in Eq. (9):

$$\mathcal{L}_{MMD}^{FT-T}(feat^{FT}, feat^T) \triangleq \left\| \mathbb{E}_{p^{FT} \sim D_{FT}} [\phi(feat^{FT})] - \mathbb{E}_{p^T \sim D_T} [\phi(feat^T)] \right\|_{\mathcal{H}_K}^2, \quad (9)$$

where $feat = H(FE(p))$. For computational convenience, the kernel function $k(a, b) = \langle \phi(a), \phi(b) \rangle$ is introduced to compute the inner product in the high-dimensional RKHS. Then, Eq. (9) is transformed to Eq. (10):

$$\begin{aligned} \mathcal{L}_{MMD}^{FT-T}(feat^{FT}, feat^T) \\ \triangleq \left\| \frac{1}{N_S^2} \sum_{i_1=1}^{N_S} \sum_{i_2=1}^{N_S} k(feat_{i_1}^{FT}, feat_{i_2}^{FT}) \right. \\ \left. - \frac{1}{N_S N_T} \sum_{i=1}^{N_S} \sum_{j=1}^{N_T} k(feat_i^{FT}, feat_j^T) \right. \\ \left. - \frac{1}{N_T^2} \sum_{j_1=1}^{N_T} \sum_{j_2=1}^{N_T} k(feat_{j_1}^T, feat_{j_2}^T) \right\|_{\mathcal{H}_K}. \end{aligned} \quad (10)$$

Additionally, we leverage multiple kernel functions [71] with different weights to mitigate the impact of kernel selection, as shown in Eq. (11):

$$K \triangleq \left\{ k = \sum_{u=1}^{n_k} \beta_u k_u : \beta_u \geq 0, \sum_{u=1}^{n_k} \beta_u = 1, \forall u \right\}, \quad (11)$$

where n_k denotes the number of kernels. Finally, the feature matching loss is formulated as Eq. (12):

$$\mathcal{L}_{FM} = \sum_{i=1}^{N_H} \mathcal{L}_{MMD_i}^{FT-T}(feat_i^{FT}, feat_i^T), \quad (12)$$

where N_H represents the number of fully connected layers in H , and $feat_i$ denotes the features output by the i -th layer.

After adaptation, the features of unlabeled target images and generated fake target images are aligned, thereby enhancing the distinguishability of target features. This alignment significantly improves the classifier's performance in the target domain.

IV. EXPERIMENTS

In this section, we evaluate the proposed LSFm on two public multi-domain palmprint databases within the framework of unsupervised domain adaptation. We begin by describing the databases and implementation details, followed by a thorough examination of experiments, including palmprint identification, palmprint verification, ablation studies, and resource cost analysis.

A. Databases

1) *Database A*: The *PolyU Multi-Spectral Palmprint Database* [64] consists of 24,000 images from 250 individuals. These images were collected under four different lighting conditions: Blue, Green, Red, and NIR (Near-Infrared) illumination, with each condition treated as a separate domain. The data collection took place in two stages, approximately nine days apart. In each stage, volunteers were instructed to capture six images of each palm, resulting in 6,000 images per domain. The original images, with a resolution of 352×288 pixels, were processed using the Region of Interest (ROI) extraction method [72]. This processing produced ROI images with a resolution of 128×128 pixels, and both the original images and ROIs are in grayscale. Some examples are shown in Fig. 5.

2) *Database B*: The *CASIA Multi-Spectral Palmprint Database* [73] comprises 7,200 images from 100 volunteers. These images were captured under six different lighting conditions across two stages, with an interval of more than one month between the stages. The six spectral wavelengths used are 460nm, 630nm, 700nm, 850nm, 940nm, and white light. We refer to the corresponding domains as D1, D2, D3, D4, D5, and D6, respectively. In each domain, during each stage, three images were captured per volunteer, resulting in a total of 1,200 images per domain. Each original image is in grayscale with a resolution of 768×576 pixels. The ROI extraction method [72] was utilized to produce ROI images with a resolution of 128×128 pixels, as shown in Fig. 6.

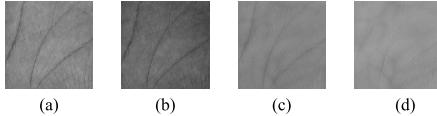


Fig. 5. Samples of database A. (a), (b), (c), and (d) are ROIs from the blue, green, red, and NIR domains, respectively.

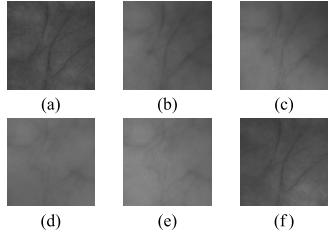


Fig. 6. Samples of database B. (a), (b), (c), (d), (e), and (f) are ROIs from the D1, D2, D3, D4, D5, and D6 domains, respectively.

B. Implementation Details

1) *Experimental Settings*: Each database is partitioned into three sets: training, validation, and testing. For database A, the data is divided in a 5 : 1 : 6 ratio for training, validation, and testing, respectively. This implies that out of 12 images from each palm, 5 images are used for training, 1 image for validation, and 6 images for testing. Consequently, the three resulting sets are as follows: training set A1 with 2,500 images (5 images \times 2 hands \times 250 individuals), validation set A2 with 500 images (1 image \times 2 hands \times 250 individuals), testing set A3 with 3,000 images (6 images \times 2 hands \times 250 individuals). Similarly, for database B, the data is divided in a 3 : 1 : 2 ratio for training, validation, and testing. Each palm in Database B has 6 images per domain. This results in training set B1 with 600 images (3 images \times 2 hands \times 100 individuals), validation set B2 with 200 images (1 image \times 2 hands \times 100 individuals), testing set B3 with 400 images (2 images \times 2 hands \times 100 individuals).

We compare our method with several state-of-the-art unsupervised domain adaptation methods: (i) Margin Disparity Discrepancy (**MDD**) [74], (ii) Minimum Class Confusion (**MCC**) [75], and (iii) Joint Pixel and Feature Alignment (**JPFA**) [10], (iv) Domain Adaptation via Transformer and Pseudo labels (**CDTrans**) [76], (v) Patch-Mix Transformer for UDA (**PMTrans**) [77], (vi) Domain Adaptation via Risk-Consistent Estimators (**RCE**) [78], and (vii) Palmprint Data and Feature Generation method (**PDFG**) [23]. For a fair comparison, we use VGG-16 (configuration D) [79] as the base palmprint classifier for all methods except CDTrans and PMTrans. To expedite training on the source domain, we initialized the palmprint classifier using pre-trained VGG-16 weights from PyTorch [80]. For CDTrans and PMTrans, we use DeiT-base [81] and Swin-base [82] as the backbones, given the outstanding performance reported in their respective studies. All experiments were conducted using PyTorch on a server equipped with four NVIDIA GeForce RTX 3090 GPUs.

As for the biometric recognition methods, there are two categories: (a) computing feature distances between query feature and registration set to determine the class of the query data [51], [83], [84], [85], and (b) predicting the probability that the input data belongs to a particular class [86], [87], [88], [89]. The first type allows for the recognition of new

classes by simply updating the registration set. However, this type of methods requires the storage of a large number of templates to build the registration set, and the time for computing feature distances cannot be overlooked. The second type has faster inference speeds and reduced storage requirements while recognizing unseen classes without fine-tuning is impossible. Although the two kinds of methods all achieve excellent performance, we adopt the second type due to its efficiency.

2) *Evaluation Metrics*: We use accuracy and EER (Equal Error Rate) as our evaluation metrics. Accuracy presents the ratio of correct identifications, as defined in Eq. (13):

$$\text{Accuracy} = \frac{1}{N_T^{\text{test}}} \sum_{i=1}^{N_T^{\text{test}}} \mathbb{I}(\hat{y}_i^T = y_i^T), \quad (13)$$

where N_T^{test} is the total number of testing samples in the target domain, and $\mathbb{I}(\cdot) = 1$ when the predicted class \hat{y}_i^T matches the true label y_i^T . In addition, EER represents the overall error rate of the recognition algorithms. Using the predicted confidence scores and a threshold, we calculate the FAR (False Accept Rate) and FRR (False Reject Rate), as shown in Eq. (14) and Eq. (15):

$$\text{FAR} = \frac{\text{FA}}{\text{FA} + \text{TR}}, \quad (14)$$

$$\text{FRR} = \frac{\text{FR}}{\text{TA} + \text{FR}}, \quad (15)$$

where FA (False Acceptance) is the number of cases that are incorrectly accepted, TR (True Rejection) is the number of cases that are correctly rejected, FR (False Rejection) is the number of cases that are incorrectly rejected, and TA (True Acceptance) is the number of cases that are correctly accepted. The threshold can be adjusted from 0 to 1 in incremental steps (we use 0.01 in our experiments), resulting in multiple pairs of FAR and FRR values. The point where FAR equals FRR is defined as EER. In practice, we calculate the EER as the average of the closest pair of values, FAR^* and FRR^* , as shown in Eq. (16):

$$\text{EER} = \frac{\text{FAR}^* + \text{FRR}^*}{2}. \quad (16)$$

C. Palmprint Identification

In this section, we conduct extensive comparative experiments between LSFm and other state-of-the-art methods for palmprint identification. Palmprint identification aims to determine the identity of individuals based on their palmprint images [90]. Upon receiving a palmprint image in an authentication system, the objective is to identify the corresponding individual. The palmprint classifier outputs predicted probabilities for the input image, and the class with the highest probability is considered the final identification result. For our experiments, one domain is selected as the source domain, while the remaining domains within the same database serve as target domains. This results in 12 and 30 different cross-domain scenarios for Databases A and B, respectively. The performance results for each cross-domain scenario are presented in Table II and Table III.

TABLE II
ACCURACY (%) OF CROSS-DOMAIN PALMPRINT IDENTIFICATION ON DATABASE A

Source Domain	Blue			Green			Red			NIR			Mean
Target Domain	Green	Red	NIR	Blue	Red	NIR	Blue	Green	NIR	Blue	Green	Red	
VGG16(Same Domain) [79]	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87
VGG16(Cross Domain) [79]	98.13	56.00	3.97	99.03	71.97	11.17	71.87	76.63	68.97	9.73	12.23	73.77	54.46
MDD [74]	95.20	80.60	25.97	96.63	89.63	28.43	58.03	62.43	67.23	18.00	22.93	71.43	59.71
MCC [75]	97.90	63.80	18.37	97.30	71.43	17.73	51.70	74.57	71.93	16.17	11.80	73.83	55.54
JPFA [10]	99.77	79.33	23.73	98.67	75.40	6.60	48.90	57.77	53.77	1.97	2.07	47.93	49.66
CDTrans [76]	5.13	2.63	0.13	12.00	5.43	1.47	15.13	9.80	3.70	3.57	2.43	8.40	5.82
PMTTrans [77]	3.53	5.97	0.17	1.27	0.70	0.30	3.93	0.13	3.03	0.07	0.30	0.57	1.66
RCE [78]	80.83	57.53	10.93	93.83	64.47	12.70	75.83	73.57	77.53	16.53	15.80	72.10	54.31
PDFG [23]	98.93	46.40	3.87	99.47	70.83	5.70	62.23	75.23	41.50	5.77	5.60	54.80	47.53
LSFM	99.87	98.83	84.93	99.60	98.00	78.03	99.57	98.90	97.63	92.87	91.70	98.47	94.87

TABLE III
ACCURACY (%) OF CROSS-DOMAIN PALMPRINT IDENTIFICATION ON DATABASE B

Source Domain	Target Domain	VGG16 [79] (Same Domain)	VGG16 [79] (Cross Domain)	MDD [74]	MCC [75]	JPFA [10]	CDTrans [76]	PMTTrans [77]	RCE [78]	PDFG [23]	LSFM
D1	D2	99.25	60.00	39.75	64.50	26.50	20.50	1.50	56.75	27.50	97.50
	D3	99.25	38.75	18.75	47.25	21.50	16.00	1.00	44.50	18.75	92.25
	D4	99.25	7.50	2.75	8.75	5.75	5.25	0.50	10.75	3.25	48.75
	D5	99.25	3.75	2.00	5.50	1.50	5.75	1.00	7.75	2.50	39.25
	D6	99.25	97.75	94.25	98.25	67.25	90.25	0.75	92.75	85.75	99.25
D2	D1	99.00	54.00	30.25	71.50	18.75	17.75	1.75	62.75	32.75	88.00
	D3	99.00	98.00	94.75	97.25	74.00	89.00	8.00	91.25	81.75	96.50
	D4	99.00	62.50	46.50	72.75	16.75	28.75	3.50	59.50	26.00	84.00
	D5	99.00	49.50	41.00	60.00	16.75	29.75	15.25	51.75	18.00	80.75
	D6	99.00	77.75	50.00	87.75	38.50	35.00	3.00	76.00	45.75	91.00
D3	D1	97.50	39.00	18.50	61.25	17.25	16.75	0.50	50.25	26.00	87.25
	D2	97.50	96.75	95.75	99.25	83.75	91.00	11.50	93.50	82.00	97.75
	D4	97.50	56.50	53.00	85.00	36.75	30.00	4.50	56.75	27.50	85.25
	D5	97.50	46.25	44.50	80.25	27.75	28.75	9.25	51.50	20.50	79.25
	D6	97.50	66.25	42.75	69.75	35.50	36.25	6.50	73.00	35.25	90.75
D4	D1	98.75	8.25	3.50	8.50	6.75	6.25	0.00	17.75	3.50	43.75
	D2	98.75	66.00	44.00	93.75	50.50	35.00	4.00	65.50	26.25	86.00
	D3	98.75	59.50	49.00	89.75	48.25	37.00	4.25	60.00	27.50	81.50
	D5	98.75	96.50	94.00	97.75	85.75	88.75	4.25	88.75	72.75	97.75
	D6	98.75	18.00	5.75	17.25	13.50	11.25	4.25	26.50	4.00	61.00
D5	D1	98.25	11.00	3.00	8.00	3.25	7.75	1.25	16.25	3.50	25.00
	D2	98.25	58.25	44.75	91.50	56.00	44.00	17.00	65.25	24.75	84.00
	D3	98.25	57.25	44.75	72.75	46.75	40.50	12.25	59.25	24.00	83.25
	D4	98.25	97.50	95.00	98.00	90.25	92.00	4.75	92.00	79.00	96.50
	D6	98.25	15.25	6.00	16.00	8.25	9.00	3.25	20.00	3.75	46.50
D6	D1	99.00	98.50	94.00	98.75	70.00	89.75	7.00	94.50	85.00	98.00
	D2	99.00	84.00	71.25	80.75	32.50	34.50	3.00	77.00	45.25	96.50
	D3	99.00	67.00	49.25	73.50	24.50	28.25	3.25	61.25	31.75	94.00
	D4	99.00	14.75	13.25	14.25	2.50	10.00	2.50	23.50	5.25	59.00
	D5	99.00	8.00	8.75	13.25	3.50	7.25	3.00	15.25	4.25	51.50
Mean		98.63	53.80	43.36	62.76	34.35	36.07	4.75	55.38	32.46	78.73

We first pre-train the base palmprint classifier (VGG16) on the source domain and evaluate it on the testing set of the same domain to obtain the *Same Domain* accuracy, which represents the classifier's optimal performance. Additionally, the *Cross Domain* accuracy is determined by evaluating the pre-trained models on the target domain's testing set, highlighting the performance degradation caused by domain shifts. The experimental results show that VGG16-based classifiers achieve superior intra-domain recognition accuracy, surpassing 97%. However, significant performance drops are observed when pre-trained source models are tested on target data, consistent with findings in previous studies [8], [9], [10]. For instance, accuracy sharply decreases from 99.87% to 3.97% in Blue→NIR. Furthermore, the average accuracy of VGG16 on database A significantly decreases from 99.87% to 54.46%. Similarly, for database B, accuracy drops from 98.63% to 53.80%. These results highlight the detrimental impact of domain shifts, revealing

the vulnerability of deep learning-based classifiers to data bias.

After applying existing UDA methods and LSFm to adapt the pre-trained source model to target data, palmprint identification accuracy improves. These enhanced results indicate that UDA methods can partially mitigate the domain shift issue. In addition, our LSFm method achieves state-of-the-art results, with higher average accuracy of 94.87% on database A and 78.73% on database B, compared to existing UDA methods. These superior results can be attributed to the two factors: 1) We emphasize both style matching and feature alignment simultaneously. 2) We utilize a more powerful and lightweight style transfer method, LightStarGAN, to facilitate adaptation and align feature distributions across multiple layers. After domain adaptation, the gap between the source and target domains is further minimized, enabling palmprint classifiers to generate more generalized and domain-invariant features for target domain data. In contrast, previous UDA methods exhibit

TABLE IV
EER (%) OF CROSS-DOMAIN PALMPRINT VERIFICATION ON DATABASE A

Source Domain	Blue			Green			Red			NIR			Mean
Target Domain	Green	Red	NIR	Blue	Red	NIR	Blue	Green	NIR	Blue	Green	Red	
VGG16 (Same Domain) [79]	0.00	0.00	0.00	0.03	0.03	0.03	0.03	0.03	0.03	0.01	0.01	0.01	0.02
VGG16 (Cross Domain) [79]	0.13	3.72	35.75	0.08	2.96	29.82	2.70	2.12	4.76	30.35	27.50	4.07	12.00
MDD [74]	1.72	12.62	31.33	0.59	3.44	27.81	13.53	7.71	11.72	31.31	30.49	6.86	14.93
MCC [75]	1.46	4.60	27.99	0.91	2.50	27.87	7.57	4.64	4.03	37.00	37.88	8.36	13.73
JPFA [10]	0.14	3.53	16.93	1.47	4.55	28.76	8.97	7.36	9.04	35.43	34.22	9.65	13.34
CDTrans [76]	26.02	30.69	49.51	20.04	24.58	37.68	17.66	21.79	33.20	28.57	37.80	24.72	29.36
PMTrans [77]	38.34	28.71	49.29	23.18	44.11	49.89	31.36	49.85	40.37	49.63	50.04	47.27	41.84
RCE [78]	1.82	4.43	27.06	0.49	4.73	27.93	2.31	2.76	3.49	23.69	23.63	3.88	10.52
PDFG [23]	0.64	13.82	40.68	0.65	7.73	40.25	13.29	8.52	24.59	42.56	41.50	21.52	21.31
LSFM	0.03	0.37	4.97	0.15	1.09	6.57	0.08	0.31	0.50	1.42	1.77	0.28	1.46

poor generalization ability. Interestingly, we also find that in some cases, the classifier without adaptation achieves higher accuracy than the adapted one, which aligns with observations in other studies [76], [77]. For example, *MDD* improves the accuracy from 56.00% to 80.60% in Blue→Red. However, accuracy drops from 98.13% to 95.20% in Blue→Green.

Three key factors contribute to these results: 1) Our experimental setup is more strict compared to previous methods. In our work, only half of the target domain's data is utilized for domain adaptation, whereas previous methods leverage the entire target domain dataset for both adaptation and testing. These approaches risk data leakage, where testing data is inadvertently seen during training, which undermines the true assessment of model generalization. 2) The limited data available in the two palmprint databases poses a challenge for previous methods in effectively learning valuable information. Transformer-based methods, such as CDTrans and PMTrans, are particularly affected. On database A, PMTrans and CDTrans achieve the lowest and second-lowest average accuracies at 1.66% and 5.82%, respectively. Similar poor results are observed on database B, with PMTrans and CDTrans achieving 4.75% and 36.07%, respectively. This is because transformer-based models require larger datasets compared to CNNs [65]. 3) Previous methods, such as MDD [74], MCC [75], CDTrans [76], PMTrans [77], and RCE [78], primarily focus on feature alignment, which is insufficient to address domain shifts when data is limited. On the other hand, methods such as JPFA [10] and PDFG [23] consider both style and feature alignment. However, their approaches to style transfer and feature matching are not robust enough to deliver superior performance under our stricter experimental conditions. Additionally, PDFG is not specifically designed for unsupervised domain adaptation. Despite these limitations, previous methods can still improve classifier performance and alleviate domain shifts. For instance, MDD and JPFA show modest improvements in Green→Red, with accuracy increasing from 71.97% to 89.63% for MDD and to 75.40% for JPFA.

In contrast, our method LSFm achieves the second-best results in certain scenarios from database B, such as D2→D3 and D6→D1. This is largely due to the small size of database B. Specifically, each domain in database B has a training set of only 600 images, significantly fewer than the 2,500 images in database A, which limits the effectiveness of our LightStarGAN model. Moreover, the subsequent feature-matching

process relies on the style-transfer method, which constraints LSFm from reaching optimal performance in these cases. At the same time, all methods show reduced performance in certain cases within database B. For example, for the scenario of D5→D1, the highest accuracy achieved is 25%, which was obtained by our method, LSFm. This is due to the significant differences between images from D5 and D1. As illustrated in Fig. 6(a) and Fig. 6(e), the image in domain D5 contains more palm-vein features compared to the palmprint textures in domain D1. Attempting to align palm-vein features with palmprint features can result in negative transfer, leading to lower accuracy. However, LSFm still outperforms previous methods in most scenarios and achieves state-of-the-art average accuracy on database B.

D. Palmprint Verification

To further evaluate the performance of LSFm, we conduct comprehensive experiments on palmprint verification. Palmprint verification involves determining whether the claimed identity of the input palmprint is valid [90]. For database A, the quantitative results of palmprint verification, presented in Table IV, demonstrate the superior performance of LSFm, which outperforms other methods by a large margin in all cross-domain scenarios. After adaptation, the average cross-domain EER is reduced from 12% to 1.46%, which significantly mitigates the negative impact of domain shifts. For database B, as shown in Table V, LSFm achieves the best performance in most cross-domain scenarios. However, in some scenarios, such as D4→D2 and D6→D1, it ranks second, consistent with the palmprint identification results in Table III. We observe that the constrained model capacity of LightStarGAN, due to the small dataset size, not only reduces the performance of LSFm in palmprint identification but also increases the EER in palmprint verification. Nonetheless, the average cross-domain EER of LSFm still surpasses that of all other methods. On database B, our method achieves a reduction of about 60% in the average EER, reaching 4.83% compared to the average EER with no adaptation. For both databases, the overall trend is consistent with the results in palmprint identification. Moreover, we draw the DET (Detection Error Tradeoff) curves of various methods to further demonstrate the effectiveness of LSFm. There are three cross-domain settings for each database. For database A, we show the results of Blue→Red, Green→Red, and NIR→Red in Fig. 7, where LSFm achieves optimal

TABLE V
EER (%) OF CROSS-DOMAIN PALMPRINT VERIFICATION ON DATABASE B

Source Domain	Target Domain	VGG16 [79] (Same Domain)	VGG16 [79] (Cross Domain)	MDD [74]	MCC [75]	JPFA [10]	CDTrans [76]	PMTrans [77]	RCE [78]	PDFG [23]	LSFM
D1	D2	0.19	6.65	18.03	8.01	16.07	18.48	46.16	8.18	24.90	1.03
	D3	0.19	10.93	26.02	15.10	23.17	21.23	46.49	10.22	30.66	1.29
	D4	0.19	31.18	43.80	31.87	30.58	36.10	50.90	30.28	43.80	12.95
	D5	0.19	35.88	43.84	34.87	43.10	35.98	45.22	34.05	43.84	14.52
	D6	0.19	0.25	3.32	0.22	6.59	3.75	39.35	0.97	6.76	0.22
	D1	0.02	8.13	20.82	5.08	24.37	16.70	42.39	5.88	26.96	2.55
D2	D3	0.02	0.55	3.06	0.31	5.06	3.71	23.68	2.02	9.66	0.60
	D4	0.02	7.88	16.41	6.35	23.50	18.50	39.04	9.53	30.07	4.41
	D5	0.02	12.12	21.32	10.43	27.43	18.07	21.81	11.82	30.39	5.47
	D6	0.02	3.49	14.56	2.82	13.99	12.74	37.25	3.78	20.35	1.58
	D1	0.28	12.33	27.07	8.02	22.77	19.09	46.47	9.68	29.78	2.87
	D2	0.28	0.22	2.00	0.19	3.20	3.26	23.09	1.05	9.82	0.33
D3	D4	0.28	12.43	15.94	2.92	18.52	16.94	35.30	10.68	27.41	4.52
	D5	0.28	14.12	18.64	4.38	21.79	15.99	28.30	13.25	31.20	4.21
	D6	0.28	5.53	15.79	7.76	14.95	13.06	31.20	5.33	22.80	1.50
	D1	0.19	32.09	44.81	34.16	33.68	31.21	46.35	23.81	42.87	11.45
	D2	0.19	7.79	16.22	2.08	9.74	13.37	36.20	6.50	26.42	3.03
	D3	0.19	8.30	16.27	2.87	12.37	13.07	35.78	8.09	29.97	3.10
D4	D5	0.19	0.44	2.97	0.33	2.66	3.50	36.87	2.89	11.73	0.64
	D6	0.19	23.90	36.87	26.82	27.23	27.02	35.79	20.70	40.49	8.01
	D1	0.24	32.31	43.66	33.56	38.89	32.81	43.13	28.05	43.69	16.85
	D2	0.24	7.78	17.53	2.07	12.10	13.56	22.05	7.79	29.15	3.98
	D3	0.24	8.38	17.05	5.80	14.01	13.19	28.77	8.22	27.65	4.07
	D4	0.24	0.55	2.01	0.36	2.39	2.50	34.38	1.94	11.43	0.45
D5	D6	0.24	25.65	38.87	28.89	30.15	28.37	37.78	23.30	42.31	11.12
	D1	0.06	0.40	2.33	0.20	5.08	3.75	32.51	0.88	8.30	0.29
	D2	0.06	2.45	6.97	4.13	15.37	12.26	39.51	6.01	21.37	0.81
	D3	0.06	4.15	12.84	5.83	18.26	17.65	38.95	7.64	23.71	1.21
	D4	0.06	24.40	31.01	27.72	41.12	27.92	42.44	23.68	39.95	9.49
	D5	0.06	31.39	36.55	31.49	41.36	34.06	37.44	27.19	42.60	12.32
Mean		0.16	12.39	20.55	11.49	19.98	17.59	36.82	11.78	27.67	4.83

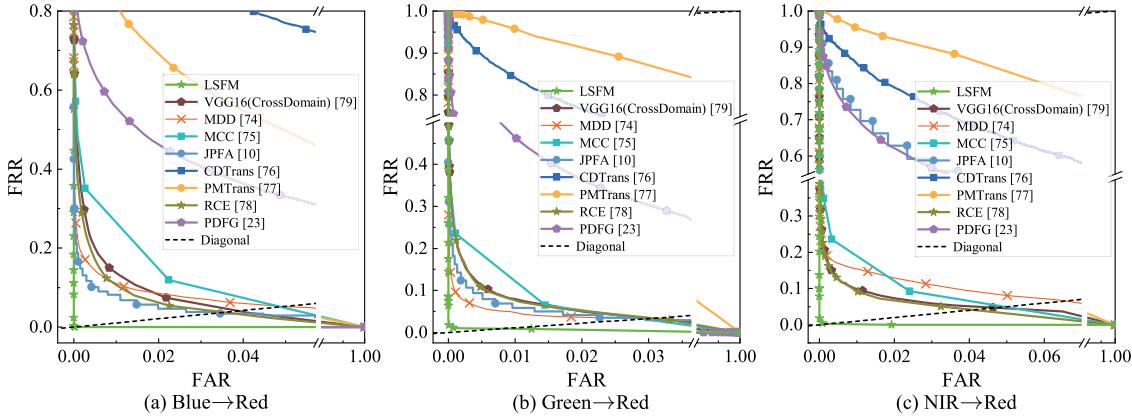


Fig. 7. DET curves of different methods on database A, including (a) Blue→Red, (b) Green→Red, and (c) NIR→Red.

performance across all three scenarios. The significant margins highlight LSFN's superior performance compared to other UDA methods. For database B, we present the results of D1→D4, D2→D4, and D6→D4 in Fig. 8. Despite the substantial decrease in available data size from 2,500 images (training set A1 in database A) to 600 images (training set B1 in database B), our method continues to outperform other state-of-the-art methods. In contrast, most methods exhibit negative transfer, meaning their performance is poorer than before adaptation.

E. Ablation Studies

In this section, we conduct ablation experiments on database A to evaluate the effectiveness of various components of our method. Our method includes the LightStarGAN and MultiFA

module (Fig. 2). To assess the impact of each component, we replace the LightStarGAN with the traditional style transfer method CycleGAN and downgrade the MultiFA to SingleFA. The palmprint identification accuracy of all variants is presented in Table VI, while the palmprint verification results are listed in Table VII. Note that “LightStarGAN + MultiFA” indicates our method LSFN. The experimental results show that upgrading CycleGAN to LightStarGAN improves average accuracy by 20%, and reduces the average EER from 8.49% to 2.84%. Similarly, when SingleFA is replaced by the MultiFA module, average accuracy increases by 2.37% and the average EER decreases from 8.48% to 6.39%. It is worth noting that the average accuracy slightly decreases when CycleGAN is incorporated with MultiFA instead of SingleFA. This is because feature alignment occurs between the features of fake

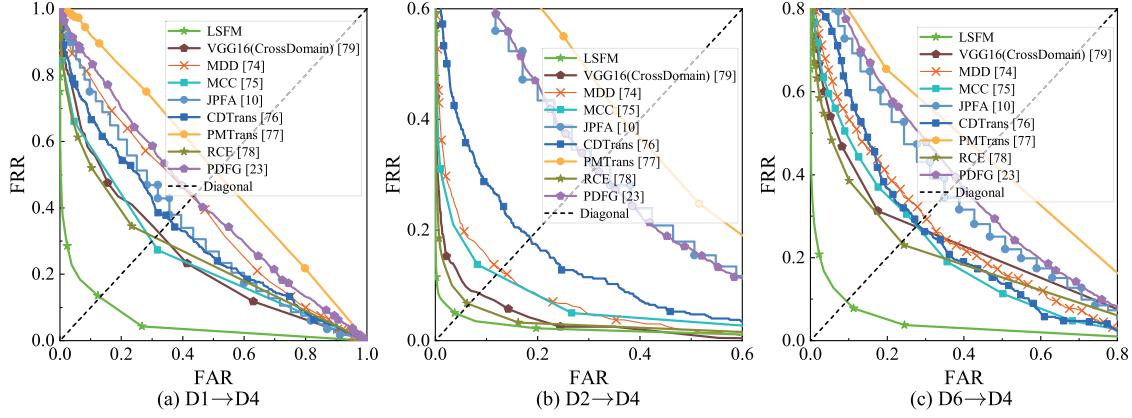


Fig. 8. DET curves of different methods on database B, including (a) D1→D4, (b) D2→D4, and (c) D6→D4.

TABLE VI
ACCURACY (%) OF ABLATION EXPERIMENTS ON PALMPRINT IDENTIFICATION

Source Domain				Blue			Green			Red			NIR			Mean
Target Domain				Green	Red	NIR	Blue	Red	NIR	Blue	Green	NIR	Blue	Green	Red	
VGG16 (Same Domain)				99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87
VGG16 (Cross Domain)				98.13	56.00	3.97	99.03	71.97	11.17	71.87	76.63	68.97	9.73	12.23	73.77	54.46
CycleGAN	LightStarGAN	SingleFA	MultiFA	99.53	86.47	16.10	99.87	90.73	19.53	92.30	95.13	80.50	16.93	18.37	85.57	66.75
		✓	✓	99.83	98.47	53.17	99.67	96.87	61.27	83.23	96.87	95.53	45.90	57.23	91.53	81.63
	✓	✓	✓	99.80	92.27	18.93	99.90	94.37	22.63	93.77	95.40	87.57	23.57	24.27	86.33	69.90
		✓	✓	99.37	94.00	13.53	99.80	95.47	36.57	86.67	90.53	91.13	38.13	33.20	88.83	72.27
	✓	✓	✓	99.90	95.73	27.93	99.83	96.47	32.50	95.07	96.83	89.07	21.50	20.63	78.97	71.20
		✓	✓	99.60	92.53	45.60	99.67	96.37	34.33	92.87	93.77	82.87	16.77	22.63	68.90	70.49
	✓	✓	✓	99.87	99.33	76.53	99.20	98.83	72.23	95.73	96.80	97.83	60.60	60.70	89.53	87.27
		✓	✓	99.87	98.83	84.93	99.60	98.00	78.03	99.57	98.90	97.63	92.87	91.70	98.47	94.87

TABLE VII
EER (%) OF ABLATION EXPERIMENTS ON PALMPRINT VERIFICATION

Source Domain				Blue			Green			Red			NIR			Mean
Target Domain				Green	Red	NIR	Blue	Red	NIR	Blue	Green	NIR	Blue	Green	Red	
VGG16 (Same Domain)				0.00	0.00	0.00	0.03	0.03	0.03	0.03	0.03	0.03	0.01	0.01	0.01	0.02
VGG16 (Cross Domain)				0.13	3.72	35.75	0.08	2.96	29.82	2.70	2.12	4.76	30.35	27.50	4.07	12.00
CycleGAN	LightStarGAN	SingleFA	MultiFA	0.03	0.93	24.27	0.00	0.76	22.72	0.72	0.37	2.63	23.98	24.03	1.45	8.49
		✓	✓	0.01	0.08	8.57	0.00	0.27	6.14	1.59	0.19	0.37	9.92	6.18	0.75	2.84
	✓	✓	✓	0.05	1.25	23.95	0.01	1.25	23.61	1.48	1.04	2.72	22.61	21.24	2.59	8.48
		✓	✓	0.05	0.94	25.05	0.03	0.98	15.16	1.53	1.90	1.71	12.28	13.46	3.61	6.39
	✓	✓	✓	0.04	0.85	19.48	0.04	0.67	19.12	1.29	0.77	2.61	22.95	22.83	3.93	7.88
		✓	✓	0.03	0.66	10.25	0.03	0.60	14.14	0.97	0.76	3.68	21.05	17.76	5.95	6.32
	✓	✓	✓	0.05	0.10	3.95	0.10	0.14	3.97	0.59	0.83	0.29	6.13	8.12	1.29	2.13
		✓	✓	0.03	0.37	4.97	0.15	1.09	6.57	0.08	0.31	0.50	1.42	1.77	0.28	1.46

target images and true target images, and CycleGAN's weaker performance results in inaccurate alignment. The stronger the feature alignment module, the more pronounced this inaccuracy becomes. In contrast, when MultiFA is combined with LightStarGAN instead of CycleGAN, the average identification accuracy significantly increases from 70.49% to 94.87%, and the average EER similarly improves, dropping from 6.32% to 1.46%. Moreover, the performance gains achieved by replacing SingleFA with MultiFA further validate the superior feature alignment capability of the MultiFA module. Then, our method achieves the best results on database A, with an average accuracy of 94.87% and an average EER of 1.46%.

F. Resource Cost

To show the superiority of our method in terms of resource efficiency, we theoretically and experimentally demonstrate significant reductions in storage size, parameters, and multi-adds. Most methods, except for the JPFA and PDFG, only

incur the cost of storing classifiers, as they lack pixel-level matching. However, these methods do not achieve excellent results. While PDFG accomplishes pixel-level alignment using a Fourier-based data augmentation method, which is cost-effective, it is not specifically designed for UDA and, as our experiments show, fails to deliver superior performance. Therefore, we conduct a comparative analysis between our method and JPFA, the state-of-the-art approach specifically tailored for UDA in palmprint recognition.

First, we theoretically prove the efficiency of our method. As shown in Eq. (2), traditional style transfer methods require $m(m - 1)$ generators for cross-domain palmprint recognition, where m is the number of domains in a database ($m > 1$). Assuming each generator requires $stor_g$ MB storage and $param_g$ parameters, the total storage costs is given by Eq. (17):

$$stor_{pre} = m \times (m - 1) \times stor_g. \quad (17)$$

TABLE VIII
COMPARISONS ABOUT RESOURCE COSTS ON DATABASE A

Method \ Resource	Storage Size (MB)		Params (Million)		Multi-Adds (Billion)	
JPFA	$43.44 \times 12 = 521.28$		$11 \times 12 = 132$		14.21	
LSFM	G	M	G	M	G	M
	22.02	15.58	5.76	4.08	2.33	4.08e-3
	37.6		9.84		2.33	
Reduction	92.79% ↓		92.55% ↓		83.60% ↓	

TABLE IX
COMPARISONS ABOUT RESOURCE COSTS ON DATABASE B

Method \ Resource	Storage Size (MB)		Params (Million)		Multi-Adds (Billion)	
JPFA	$43.44 \times 30 = 1303.2$		$11 \times 30 = 330$		14.21	
LSFM	G	M	G	M	G	M
	22.02	15.58	5.76	4.08	2.33	4.08e-3
	37.6		9.84		2.33	
Reduction	97.11% ↓		97.02% ↓		83.60% ↓	

Similarly, the total number of parameters is expressed by Eq. (18):

$$param_{pre} = m \times (m - 1) \times param_g. \quad (18)$$

In contrast, our method only requires a single unified lightweight generator G and a mapping network M , regardless of the number of domains. The total storage costs and parameters for our method are shown in Eq. (19) and Eq. (20):

$$stor_{lsfm} = stor_{lg} + stor_m, \quad (19)$$

$$param_{lsfm} = param_{lg} + param_m. \quad (20)$$

The values $stor_{lsfm}$ and $param_{lsfm}$ are smaller than $stor_g$ and $param_g$, respectively, as our generator and mapping network are lightweight. Experimental results indicate that $stor_{lsfm} \approx 86.56\% \times stor_g$, $param_{lsfm} \approx 89.45\% \times param_g$. Then, the theoretical reductions in storage costs and parameters are given by Eq. (21) and Eq. (22):

$$r_{stor} = \frac{stor_{pre} - stor_{lsfm}}{stor_{pre}} \approx (100 - \frac{86.56}{m(m-1)})\%, \quad (21)$$

$$r_{param} = \frac{param_{pre} - param_{lsfm}}{param_{pre}} \approx (100 - \frac{89.45}{m(m-1)})\%. \quad (22)$$

Consequently, as the number of domains m increases, the reduction percentages become larger.

Next, we present the experimental results in Tables VIII and IX to validate our theoretical analysis. These results are based on inferencing a single image with a resolution of $3 \times 128 \times 128$. For database A, which contains four different domains, $4 \times 3 = 12$ cross-domain scenarios require the training and deployment of 12 distinct generators, resulting in a total storage cost of 521.28MB for JPFA. In contrast, our method requires only 37.6MB, representing a 92.97% reduction in storage consumption. Similarly, LSFN reduces the parameters by 92.55%. For database B, which has more cross-domain pairs ($6 \times 5 = 30$), the storage cost reduction exceeds 97%, consistent with our theoretical predictions. Specifically, LSFN reduces storage costs from 1303.2MB to 37.6MB and cuts parameters from 330 million to 9.84 million. Beyond storage and parameter reductions, our method also

optimizes multi-adds, reducing them from 14.21 billion to approximately 2.33 billion, thanks to the use of lightweight blocks, as shown in Fig. 4.

In summary, our method conserves resources by more than 80% compared to the previous style transfer-aided optimal method for unsupervised cross-domain palmprint recognition. Additionally, LSFN strikes a balance between resource efficiency and performance, achieving excellent results with minimal additional resource costs for style transfer models compared to UDA methods focused solely on feature alignment.

V. CONCLUSION

In this paper, we propose LSFN, a lightweight method for cross-domain palmprint recognition to mitigate domain shifts while keeping resource consumption low. We develop a LightStarGAN, a unified model that performs style matching across multiple domains. After reducing the domain gaps by style matching, we align the features of task-specific layers between the fake target domain and the target domain data. Experimental results on two public multi-domain palmprint databases demonstrate that LSFN achieves optimal performance with minimal resource costs through the two-stage alignment process. We believe this work could significantly advance practical applications of palmprint recognition in the future.

REFERENCES

- [1] D. Zhong, X. Du, and K. Zhong, "Decade progress of palmprint recognition: A brief survey," *Neurocomputing*, vol. 328, pp. 16–28, Feb. 2019.
- [2] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, pp. 1–46, Oct. 2020.
- [3] Y. Zhang, "A survey of unsupervised domain adaptation for visual recognition," 2021, *arXiv:2112.06745*.
- [4] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, Jul. 2015, pp. 97–105.
- [5] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.
- [6] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
- [7] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang, "Confidence regularized self-training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5981–5990.
- [8] X. Du, D. Zhong, and H. Shao, "Cross-domain palmprint recognition via regularized adversarial domain adaptive hashing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2372–2385, Jun. 2021.
- [9] H. Shao, D. Zhong, and Y. Li, "PalmGAN for cross-domain palmprint recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1390–1395.
- [10] H. Shao and D. Zhong, "Towards cross-dataset palmprint recognition via joint pixel and feature alignment," *IEEE Trans. Image Process.*, vol. 30, pp. 3764–3777, 2021.
- [11] J. Hoffman et al., "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1989–1998.
- [12] Y. Deng et al., "StyTr2: Image style transfer with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11316–11326.
- [13] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1857–1865.
- [14] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

- [15] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4396–4405.
- [16] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8107–8116.
- [17] T. Karras et al., "Alias-free generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, Jun. 2021, pp. 852–863.
- [18] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.
- [19] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN v2: Diverse image synthesis for multiple domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8185–8194.
- [20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [21] P. Kapoor and T. D. Bui, "TinyStarGAN v2: Distilling StarGAN v2 for efficient diverse image synthesis for multiple domains," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2, 2021, p. 3.
- [22] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [23] H. Shao, Y. Zou, C. Liu, Q. Guo, and D. Zhong, "Learning to generalize unseen dataset for cross-dataset palmprint recognition," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 3788–3799, 2024.
- [24] L. Fei, G. Lu, W. Jia, S. Teng, and D. Zhang, "Feature extraction methods for palmprint recognition: A survey and evaluation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 2, pp. 346–363, Feb. 2019.
- [25] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.
- [26] A. W.-K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 1, Aug. 2004, pp. 520–523.
- [27] D.-S. Huang, W. Jia, and D. Zhang, "Palmprint verification based on principal lines," *Pattern Recognit.*, vol. 41, no. 4, pp. 1316–1328, Apr. 2008.
- [28] L. Fei, B. Zhang, Y. Xu, D. Huang, W. Jia, and J. Wen, "Local discriminant direction binary pattern for palmprint representation and recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 468–481, Feb. 2019.
- [29] D. Palma, P. L. Montessoro, G. Giordano, and F. Blanchini, "Biometric palmprint verification: A dynamical system approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2676–2687, Dec. 2019.
- [30] Y. Xu, L. Fei, J. Wen, and D. Zhang, "Discriminative and robust competitive code for palmprint recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 2, pp. 232–241, Feb. 2018.
- [31] S. Zhao and B. Zhang, "Learning salient and discriminative descriptor for palmprint feature extraction and identification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5219–5230, Dec. 2020.
- [32] S. Zhao and B. Zhang, "Learning complete and discriminative direction pattern for robust palmprint recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 1001–1014, 2021.
- [33] Z. Yang, L. Leng, T. Wu, M. Li, and J. Chu, "Multi-order texture features for palmprint recognition," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 995–1011, Feb. 2023.
- [34] A. K. Jain and J. Feng, "Latent palmprint matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1032–1047, Jun. 2009.
- [35] E. Liu, A. K. Jain, and J. Tian, "A coarse to fine minutiae-based latent palmprint matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2307–2322, Oct. 2013.
- [36] J. Dai, J. Feng, and J. Zhou, "Robust and efficient ridge-based palmprint matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1618–1632, Aug. 2012.
- [37] J. Dai and J. Zhou, "Multifeature-based high-resolution palmprint recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 945–957, May 2011.
- [38] I. S. Hussein, S. B. Sahibuddin, M. J. Nordin, and N. N. B. A. Sjarif, "Multimodal recognition system based on high-resolution palmprints," *IEEE Access*, vol. 8, pp. 56113–56123, 2020.
- [39] L. Fei, B. Zhang, Y. Xu, W. Jia, J. Wen, and J. Wu, "Precision direction and compact surface type representation for 3D palmprint identification," *Pattern Recognit.*, vol. 87, pp. 237–247, Mar. 2019.
- [40] A. Iula and D. Nardiello, "3-D ultrasound palmprint recognition system based on principal lines extracted at several under skin depths," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 12, pp. 4653–4662, Dec. 2019.
- [41] A. Iula and M. Micucci, "A feasible 3D ultrasound palmprint recognition system for secure access control applications," *IEEE Access*, vol. 9, pp. 39746–39756, 2021.
- [42] L. Dian and S. Dongmei, "Contactless palmprint recognition based on convolutional neural network," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Nov. 2016, pp. 1363–1367.
- [43] A. Genovese, V. Piuri, K. N. Plataniotis, and F. Scotti, "PalmNet: Gabor-PCA convolutional networks for touchless palmprint recognition," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3160–3174, Dec. 2019.
- [44] D. Zhong and J. Zhu, "Centralized large margin cosine loss for open-set deep palmprint recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1559–1568, Jun. 2020.
- [45] D. Zhong, H. Shao, and X. Du, "A hand-based multi-biometrics via deep hashing network and biometric graph matching," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3140–3150, Dec. 2019.
- [46] K. Zhou, X. Zhou, L. Yu, L. Shen, and S. Yu, "Double biologically inspired transform network for robust palmprint recognition," *Neurocomputing*, vol. 337, pp. 24–45, Apr. 2019.
- [47] Y. Liu and A. Kumar, "Contactless palmprint identification using deeply learned residual features," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 2, no. 2, pp. 172–181, Apr. 2020.
- [48] S. Zhao and B. Zhang, "Deep discriminative representation for generic palmprint recognition," *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107071.
- [49] S. Zhao and B. Zhang, "Joint constrained least-square regression with deep convolutional feature for palmprint recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 511–522, Jan. 2020.
- [50] X. Liang, Z. Li, D. Fan, J. Li, W. Jia, and D. Zhang, "Touchless palmprint recognition based on 3D Gabor template and block feature refinement," *Knowledge-Based Syst.*, vol. 249, Aug. 2022, Art. no. 108855.
- [51] S. A. Grosz, A. Godbole, and A. K. Jain, "Mobile contactless palmprint recognition: Use of multiscale, multimodel embeddings," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 8428–8440, 2024.
- [52] H. Shao and D. Zhong, "Few-shot palmprint recognition via graph neural networks," *Electron. Lett.*, vol. 55, no. 16, pp. 890–892, Aug. 2019.
- [53] H. Shao, D. Zhong, X. Du, S. Du, and R. N. J. Veldhuis, "Few-shot learning for palmprint recognition via meta-Siamese network," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
- [54] B. Liu, Z. Wang, and J. Feng, "PalmNet: A robust palmprint minutiae extraction network," in *Proc. Asian Conf. Pattern Recognit.*, 2021, pp. 459–473.
- [55] M. Korichi, D. Samai, A. Meraoumia, and A. Benlamoudi, "Towards effective 2D and 3D palmprint recognition using transfer learning deep features and ReliefF method," in *Proc. Int. Conf. Recent Adv. Math. Informat. (ICRAMI)*, Sep. 2021, pp. 1–6.
- [56] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*.
- [57] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.
- [58] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1180–1189.
- [59] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732.
- [60] Z. Du, J. Li, H. Su, L. Zhu, and K. Lu, "Cross-domain gradient discrepancy minimization for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3936–3945.
- [61] S. Xie, Z. Zheng, L. Chen, and C. Chen, "Learning semantic representations for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1–10.
- [62] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3801–3809.

- [63] J. Jiang, X. Wang, M. Long, and J. Wang, "Resource efficient domain adaptation," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 2220–2228.
- [64] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An online system of multispectral palmprint verification," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 480–490, Feb. 2010.
- [65] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–21.
- [66] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
- [67] M. Tan and Q. V. Le, "EfficientNetv2: Smaller models and faster training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10096–10106.
- [68] X. Dong et al., "CSWin transformer: A general vision transformer backbone with cross-shaped windows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12124–12134.
- [69] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
- [70] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu, "Equivalence of distance-based and RKHS-based statistics in hypothesis testing," *Ann. Stat.*, vol. 41, no. 5, pp. 2263–2291, 2013.
- [71] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, Jan. 2012.
- [72] H. Qin, M. A. E. Yacoubi, J. Lin, and B. Liu, "An iterative deep neural network for hand-vein verification," *IEEE Access*, vol. 7, pp. 34823–34837, 2019.
- [73] Y. Hao, Z. Sun, T. Tan, and C. Ren, "Multispectral palm image fusion for accurate contact-free palmprint recognition," in *Proc. 15th IEEE Int. Conf. Image Process.*, Mar. 2008, pp. 281–284.
- [74] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 7404–7413.
- [75] Y. Jin, X. Wang, M. Long, and J. Wang, "Minimum class confusion for versatile domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 464–480.
- [76] T. Xu, W. Chen, W. Pichao, F. Wang, H. Li, and R. Jin, "CDTrans: Cross-domain transformer for unsupervised domain adaptation," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–14.
- [77] J. Zhu, H. Bai, and L. Wang, "Patch-mix transformer for unsupervised domain adaptation: A game perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 3561–3571.
- [78] F. Ding, J. Li, W. Tian, S. Zhang, and W. Yuan, "Unsupervised domain adaptation via risk-consistent estimators," *IEEE Trans. Multimedia*, vol. 26, pp. 1179–1187, 2023.
- [79] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
- [80] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
- [81] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.
- [82] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
- [83] J. Huang, A. Zheng, M. S. Shakeel, W. Yang, and W. Kang, "FVFSNet: Frequency-spatial coupling network for finger vein authentication," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1322–1334, 2023.
- [84] L. Su, L. Fei, S. Zhao, J. Wen, J. Zhu, and S. Teng, "Learning modality-invariant binary descriptor for crossing palmprint to palm-vein recognition," *Pattern Recognit. Lett.*, vol. 172, pp. 1–7, Aug. 2023.
- [85] B. Huang, Z. Wang, J. Yang, Z. Han, and C. Liang, "Unlabeled data assistant: Improving mask robustness for face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 3109–3123, 2024.
- [86] Q. Zhu et al., "Contactless palmprint image recognition across smartphones with self-paced CycleGAN," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4944–4954, 2023.
- [87] Z. Pan, J. Wang, Z. Shen, and S. Han, "Disentangled representation and enhancement network for vein recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 4164–4176, Aug. 2023.
- [88] Y. Li, C. Tang, S. Deng, H. Qin, and H. Huang, "Mixed automatic adversarial augmentation network for finger-vein recognition," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–10, 2024.
- [89] W. Zhao, X. Zhu, K. Guo, H. Shi, X.-Y. Zhang, and Z. Lei, "Masked face transformer," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 265–279, 2024.
- [90] A. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*, vol. 479. Springer, 1999.



Song Ruan received the bachelor's degree in automation from Wuhan University of Technology, Hubei, China, in June 2020. He is currently pursuing the Ph.D. degree with the College of Computer Science, Chongqing University, China. His research interests include palm-vein identification and machine learning.



Yantao Li (Senior Member, IEEE) received the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in December 2012. He is currently a Professor with the College of Computer Science, Chongqing University. His research areas include mobile computing and security, the Internet of Things, sensor networks, and ubiquitous computing. He received 2020 Best Paper Award from IEEE Internet Computing in 2022. He was a recipient of the Outstanding Ph.D. Thesis Award in Chongqing in 2014 and the Outstanding Master's Thesis Award in Chongqing in 2011. He serves as an Associate Editor for IEEE INTERNET OF THINGS JOURNAL (IoT-J).



Huafeng Qin received the B.Sc. degree from the School of Mathematics and Physics, Chongqing University of Technology, the M.Eng. degree from the College of Electronic and Automation, Chongqing University of Technology, and the Ph.D. degree from the College of Opto-Electronic Engineering, Chongqing University. He was a Visiting Student for 12 months with Nanyang Technological University and then a Post-Doctoral Researcher for two years with Université Paris-Saclay. Currently, he is a Professor with the National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, China. His research interests include biometrics (e.g., vein, face, and gait) and machine learning.