VPCFormer: A Transformer-based Multi-View Finger Vein Recognition Model and A New Benchmark

Pengyang Zhao[a,b], Yizhuo Song[b], Siqi Wang[b], Jing-Hao Xue[c], Shuping Zhao[d], Qingmin Liao[a,b], Wenming Yang[a,b,*]

[a]*Department of Electronic Engineering, Tsinghua University, China.*
[b]*Shenzhen International Graduate School, Tsinghua University, China.*
[c]*Department of Statistical Science, University College London, UK.*
[d]*School of Computer Science, Guangdong University of Technology, China.*

Abstract

In the past decade, finger vein authentication garners significant interest. However, most existing databases and algorithms predominantly focused on single-view finger vein recognition. The current projection of vein patterns actually maps a 3D network topology into a 2D plane, which inevitably leads to 3D feature loss and topological ambiguity in 2D images. Additionally, single-view based methods are sensitive to finger rotation and translation in practical applications. So far, there are currently few dedicated studies and public databases on multi-view finger vein recognition. To address these issues, we first establish a benchmark for future research by constructing the multi-view finger vein database, named Tsinghua Multi-View Finger Vein-3 Views (THUMVFV-3V) Database , which is collected over two sessions. THUMVFV-3V provides three types of Regions of Interest (ROIs) and includes unified preprocessing operations, catering to the majority of existing methods. Furthermore, we propose a novel Transformer-based model named Vein Pattern Constrained Transformer (VPCFormer) for multi-view finger vein recognition, primarily composed of multiple Vein Pattern Constrained Encoders (VPC-Encoders) and Neighborhood-Perspective Modules (NPMs). Specifically, the VPC-Encoder incorporates a novel Vein Pattern Attention Module (VPAM) and an Integrative Feed-Forward Network (IFFN). Motivated by the fact that the strong correlations veins exhibit across different views, we devise the VPAM. Assisted by a vein mask, VPAM is meticulously designed to exclusively extract intra- and inter-view dependencies between vein patterns. Further, we propose IFFN to efficiently aggregate the preceding attention and contextual information of VPAM. In addition, the NPM is utilized to capture the correlations within a single view, enhancing the final multi-view finger vein representation. Extensive experiments demonstrate the superiority of our VPCFormer.

The THUMVFV-3V database is available at https://github.com/Pengyang233/THUMVFV-3V-Database.

Keywords: Database, multi-view finger vein recognition, Transformer, attention mechanism

1. Introduction

Biometric authentication has been extensively applied in many scenarios with the aim for enhancing security and convenience in people's daily life. Face and fingerprint are the most prevalent traits in practical applications, thanks to low-cost devices and user-friendly characteristics. However, these extrinsic traits are susceptible to damage and forgery, rendering them less than ideal for reliable recognition. In this situation, finger vein has garnered considerable interest in the realm of biometric authentication [1]. This trait refers to the vessels beneath the finger skin, which are inherently resistant to abrasion and variations in skin condition. Furthermore, as they are only detectable by using Near-Infrared (NIR) cameras, finger veins remain invisible to standard RGB cameras.

In recent years, significant advancements have been made in the field of finger vein recognition [2, 3, 4]. Predominantly, these algorithms utilized single-view images for recognition, which has been plagued by persistent challenges. For instance, the captured images often fail to encompass an extensive field of finger veins, resulting in limited identity information. Crucially, since real finger veins located within a finger resemble a 3D network structure, the 3D information is lost when veins are captured from a single view. Furthermore, the content of finger vein images captured from one view is heavily influenced by the rotations and translations of the finger, thereby leading to performance degradation.

Various studies have proposed solutions to the aforementioned issues separately. To counteract spoofing attacks using printed finger veins, Qiu et al. [5] differentiated between genuine finger vein images and spoofed attacks by analyzing the blurriness and noise distribution. Considering the unique characteristics of finger vein images, Yang et al. [2] extracted the anatomy structure of veins, encompassing the vein network and backbone. To address deformations caused by finger rotations and misalignments, an elliptical model was introduced in [6] to predict rotated

*Corresponding author.
Email address: yangelwm@163.com (Wenming Yang)

vein patterns during the matching process. Furthermore, Meng et al. [7] employed vein minutiae extraction and matching to circumvent deformation interference. However, attempting to resolve these issues with auxiliary algorithms in a finger vein recognition system can potentially decelerate the system and render it more complicated and vulnerable. Therefore, a simpler yet effective approach is required. Encouragingly, multi-view finger vein recognition can accomplish this task elegantly.

Unlike single-view finger vein recognition, multi-view finger vein recognition utilizes finger vein images from multiple views for authentication. Typically, the captured views are evenly distributed around the longitudinal axis of the finger. The reasons why multi-view finger vein recognition can address the aforementioned issues can be summarized as follows: 1) For multi-view finger vein images, the visual correlations between adjacent views can substantially elevate the difficulty of stealing vein patterns and diminish the likelihood of attacks utilizing printed vein patterns; 2) Multi-view images contain richer content than single-view images. In cases of pose variations, multiple views can reference each other, thereby compensating for the limitation of vein information in a single view; 3) Multi-view images encapsulate vein structure information in 3D space. Algorithms could potentially extrapolate spatial features of finger veins from these multi-view inputs, thereby enhancing the discriminability and robustness of the features.

As an emerging area in finger vein recognition research, multi-view recognition has only a limited number of dedicated studies and publicly available databases. Therefore, to foster progress in this field and establish a benchmark for future research, we contribute the following in this work:

1. We propose the THUMVFV-3V database[1], a multi-view finger vein database collected over two sessions, providing multiple types of ROIs and finger masks to cater to the majority of existing finger vein algorithms. Extensive experiments are conducted to verify the reliability and effectiveness of the THUMVFV-3V database.
2. A Transformer-based network, VPCFormer, is proposed for multi-view finger vein recognition. The VPCFormer consists of several Vein Pattern Constrained Encoders (VPC-Encoder) and Neighborhood-Perspective Modules (NPM). Equipped with a novel Vein Pattern Attention Module (VPAM) and an Integrative Feed-Forward Network (IFFN),

[1] https://github.com/Pengyang233/THUMVFV-3V-Database

VPC-Encoder effectively captures global intra- and inter-view correlations with the constraint of vein patterns. Additionally, NPM aims to extract local correlations within a single view.

3. Compared with a variety of methods, VPCFormer achieves the best performance on the multi-view finger vein recognition. Additionally, ablation studies clearly demonstrate the effectiveness of the proposed modules.

The remainder of this paper is organized as follows: Section 2 reviews several related topics. The detailed information about our multi-view finger vein imaging device and the THUMVFV-3V database are presented in Section 3 and Section 4, respectively. Section 5 provides a comprehensive description of our VPCFormer. Extensive experiments and discussion are conducted in Section 6. Finally, Section 7 summarizes the paper.

2. Related work

2.1. Finger Vein Recognition

Single-view finger vein recognition: Over a decade of development has seen significant advancements in single-view finger vein recognition technology. Drawing inspiration from the distinct visual contrast between finger vein regions and their background in images, methods such as Local Maximum Curvature (LMC) [8], Wide Line Detector (WLD) [6], and Enhanced Maximum Curvature (EMC) [9] were introduced. To extract more informative features, SSP-DBFL [4] was proposed for joint learning from two types of input features. Recognizing the potential of deep learning techniques in computer vision, several researchers adapted these methods specifically for finger vein recognition. A case in point is FV-GAN [10], a pioneering approach that utilizes Generative Adversarial Networks (GANs). Song et al. [11] introduced EIFNet, a model adept at fusing both implicit and explicit features to produce more discriminative results. Furthermore, motivated by the success of Vision Transformer (ViT) [12], Huang et al. [13] developed Finger Vein Transformer (FVT), leveraging a pyramid structure for multilevel feature extraction.

Multi-view finger vein recognition: As an emerging research field, there exist limited methods for multi-view finger vein recognition. Kang et al. [14] were pioneers in developing a full-view finger vein recognition algorithm, which maps finger veins to the finger surface using an

elliptical model. Subsequently, Yang et al. [15] employed multi-view finger surface and vein images to generate finger point clouds for recognition. They further integrated the attention mechanism into MVCNN [16] to effectively handle multi-view inputs [17]. More recently, HCAN [18] leveraged both global and local features to yield a robust and discriminative multi-view representation. FV-LT [19] implemented a pre-trained Transformer and stacked an additional three blocks, incorporating a local information matrix, to extract multi-view finger vein features.

Different from the aforementioned methods, our VPCFormer extracts intra- and inter-view correlations with the constraint of vein patterns and captures correlations between the background and veins within a local neighborhood. This design enhances the model's efficiency in extracting multi-view finger vein features.

2.2. Multi-view Finger Vein Databases

Multi-view finger vein recognition is an emerging research field with a limited number of databases. The detailed information of four databases is provided in Table 1.

Vein-Plus [20] is the first multi-view finger vein database. This database includes 252 classes from 63 subjects, each providing the index and middle fingers of both hands. The author adopted one NIR camera, rotating around the finger to record a video. Video frames is served as multi-view finger vein images. Each finger was recorded five times to obtain 358~360 views, resulting in a total of 454,840 images approximately. However, Vein-Plus is only available in the European Union (EU) region.

Kang et al. [14] proposed a publicly available multi-view finger vein database comprising 8,526 images across 203 classes. This database includes three views of each finger, each view being captured 14 times.

The MultiView-FV database [21], a three-view finger vein database, comprises 6,480 images from 135 volunteers, each providing the index and middle fingers of both hands. Four images were captured for each view of each finger, resulting in 540 classes.

The most recent database, LFMB-3DFB [15], includes 41,700 finger vein images. It includes 695 classes from 174 volunteers' index and middle fingers. Each finger was captured from six views, and each view was captured ten times.

However, in biometric authentications, the time span plays a crucial role in affecting the recognition performance. Compared with the existing databases, our THUMVFV-3V is collected over two separate sessions, which provides a more practical evaluation of multi-view finger vein

Table 1: THUMVFV-3V and other multi-view finger vein databases.

Database	Year	#Views	#Classes	Two sessions	Available	#Images/view	#Images	ROI
Vein-Plus [20]	2018	358~360	252	✘	✘[a]	5	≈454,860	✘
Kang et al. [14]	2020	3	203	✘	✔	14	8,526	✘
MultiView-FV [21]	2021	3	540	✘	✘[b]	4	6,480	✘
LFMB-3DFB [15]	2021	6	695	✘	✘	10	41,700	✘
THUMVFV-3V (ours)	2023	3	660	✔	✔	12	23,760	✔

[a] Only available in the EU region.

[b] The download link [21] for this database is unavailable.

recognition systems by capturing temporal variations in the collected samples. In addition, three types of ROIs are provided in THUMVFV-3V for fair comparisons.

2.3. Transformer

In 2017, Vaswani et al. [22] pioneered the Transformer model, leveraging the self-attention mechanism for natural language processing. Relative to Long Short-Term Memory (LSTM) networks [23], the Transformer exhibits superior parallel computing capabilities, long-range dependency capturing, and robust representation learning ability [22].

The Vision Transformer (ViT) [12] underscored the substantial potential of the Transformer for vision tasks, instigating a swift evolution of Transformers within the realm of computer vision in subsequent years. Later works sought to address prevalent challenges associated with Transformer, including the need for large-scale training data [24] and computational efficiency [25]. Furthermore, to efficiently extract features of varying granularities in images, the Pyramid Vision Transformer (PVT) [26] mapped image patches of different sizes into Transformer encoders at distinct levels, employing a pyramid structure to achieve this objective. Alternatively, the Transformer iN Transformer (TNT) [27] combined local and global Transformers for enhanced feature extraction and context modeling.

Encouraged by the ViT and its variants, we propose VPCFormer, a model based on the Transformer architecture, designed to exploit intra- and inter-view vein pattern correlations within multi-view finger vein images.

3. Multi-view finger vein imaging device system

The multi-view finger vein imaging system [18] is employed for a reliable data collection, with a few improvements introduced. This imaging device [18] encompasses four main compo-

nents: an NIR camera, an NIR light source, a finger support module, and a rotation controller, as shown in Fig. 1.

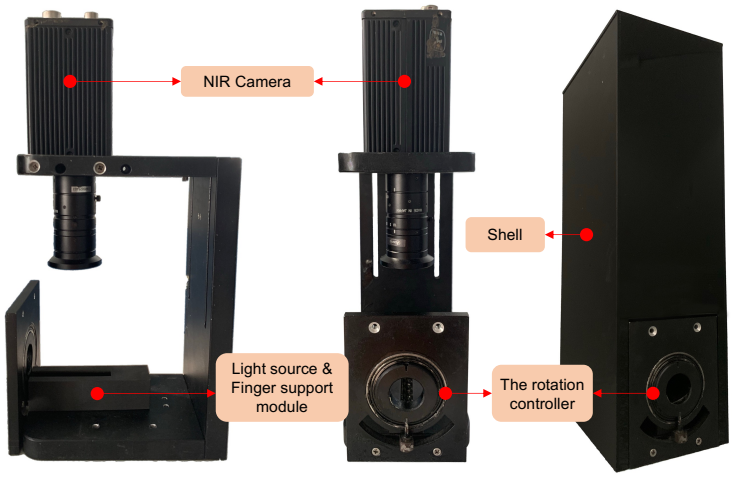


Figure 1: The multi-view finger vein imaging device.

The NIR camera is the AD-130GE manufactured by JAI Ltd.[2], receiving the NIR light with a wavelength in the range of 760nm to 1000nm. Based on the observations from [28] and our experimental findings, we found that a homogenizer plate tends to make the acquisition system more sensitive to variations in light intensity and may result in overexposed images. Consequently, we revised the NIR light source of the system. Seven LEDs were systematically arranged in a line on the circuit board to ensure uniform light, managed by a current source. Each LED possesses a 45° beam angle, emitting light at a wavelength of 850nm. Furthermore, the finger support module was also redesigned to improve volunteer comfort and prevent inadvertent finger bending. To mitigate the risk of overexposure, we selected black nylon as the 3D-printing material [28]. Different from the original work [18], the acquisition device is adapted to capture palmar veins, which have been demonstrated to be more discriminative than dorsal veins [20].

[2]https://www.jai.com/

4. Database

4.1. Basic Information

To collect a substantial number of samples[3], we invited 180 volunteers to participate in our data acquisition process. The entire collection process was divided into two separate sessions with a minimum interval of 30 days, a maximum of 106 days, and an average of 45.8 days. In addition, a total of 171 subjects attended both sessions, while the remaining 9 were absent for the second session. In each session, all subjects offered the index and middle fingers of both hands for imaging. For each finger, six samples were obtained in each session, resulting in 18 finger vein images in total. Each sample consists of 3 images, each from a distinct view.

In constructing the THUMVFV-3V, only the subjects who participated in both sessions were considered. After excluding images with non-compliance and overexposure, our THUMVFV-3V comprises 660 classes with a total of 23,760 finger vein images. Furthermore, THUMVFV-3V is a gender-balanced database, including 92 males and 79 females (1.16:1).

4.2. Acquisition Details

During each capture process, subjects are guided to place their finger into the device in a relaxed and natural manner, with the finger palm facing upward. At this point, we control the device to capture finger vein images from the 0° view. The subject then rotates their finger in a clockwise direction until reaching the maximum angle indicated by the rotation controller, and images from the $+45^\circ$ view are obtained. Similarly, the subject rotates their finger counter-clockwise until reaching a predetermined stop, at which point images from the -45° view are acquired.

Upon completion of the aforementioned capture process in each session, the subject is asked to remove their finger from the device and rest for approximately 5 to 10 seconds. Subsequently, the subject repositions their finger within the device and repeats the entire capture process for the next acquisition.

Given that our device does not enforce a strict finger posture, slight displacements and minor rotational variations are occurred among different multi-view finger vein samples for each category. This further enhances the intra-class diversity.

[3] A 'sample' in this study refers to a set of images obtained from different views of a finger.

4.3. Preprocessing Operations

Numerous finger vein databases only provide raw images without finger masks or ROIs, leading to a lack of description of preprocessing operations in many studies towards model design. Given the variations in preprocessing steps or ROIs, methods generally require to re-tune hyperparameters or otherwise produce quite different performance. This situation poses significant challenges for the reproducibility of the algorithms. To ensure a fair and reliable comparison, we standardized the preprocessing operations for THUMVFV-3V, and provided three types of ROIs to accommodate the majority of existing algorithms.

(a) (b) (c) (d)

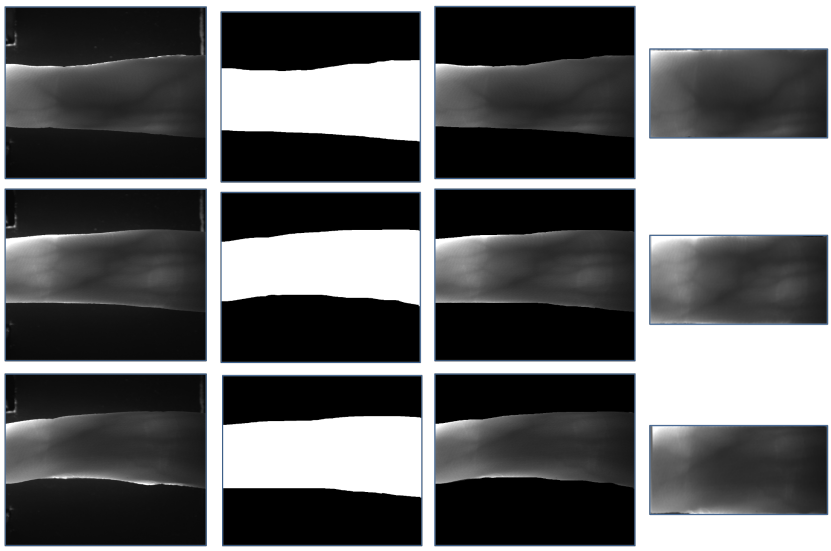
Figure 2: Illustration of preprocessing operations. (a) Vein image. (b) Coarse finger mask. (c) Refined finger mask. (d) ROI after angle alignment.

The preprocessing operations have four steps: 1) Coarse detection of finger boundary. 2) Refinement of finger boundary. 3) Angle alignment. 4) ROI generation. We give a brief illustration of the preprocessing operations in Fig. 2.

In our THUMVFV-3V, three types of ROIs are provided, denoted as ROI_1, ROI_2 and ROI_3, respectively.

- ROI_1: This type of ROI includes all content and only undergoes angle alignment.
- ROI_2: This type of ROI only contains finger regions. All non-finger areas are masked by the finger mask.
- ROI_3: Based on ROI_2, the finger region is linearly interpolated along the column direction to expand the vein area, covering the entire image. Subsequently, ROI_3 is resized into 100×200.

Some examples of ROI_1, ROI_2, ROI_3 and finger mask are shown in Fig. 3.



(a) ROI_1 (b) Finger mask (c) ROI_2 (d) ROI_3

Figure 3: Examples of THUMVFV-3V images with three types of ROIs and finger masks.

5. Methodology

5.1. Overall architecture

Vein patterns are recognized as the most distinctive features in finger vein images. The background variations in finger vein images are influenced by the changes in subcutaneous tissue thickness and fat content along the NIR light path [4]. The involvement of multiple views introduces complexity and unpredictability in the correlations among different view background. On the other hand, vein patterns in adjacent views, situated within the same spatial region, should exhibit strong correlations between views, despite visual differences in vein patterns across the two view images. Therefore, it is more reasonable and simpler to consider the correlations between vein patterns across different views than those between background.

In finger vein images, both the veins and the surrounding background jointly contribute to the final imaging outcomes [4]. Hence, it is essential to consider not only the relationships between different vein patterns, but also the local correlations between veins and background.

Based on these observations, we propose a Transformer-based model, named Vein Pattern Constrained Transformer (VPCFormer) for multi-view finger vein feature extraction. The VPC-

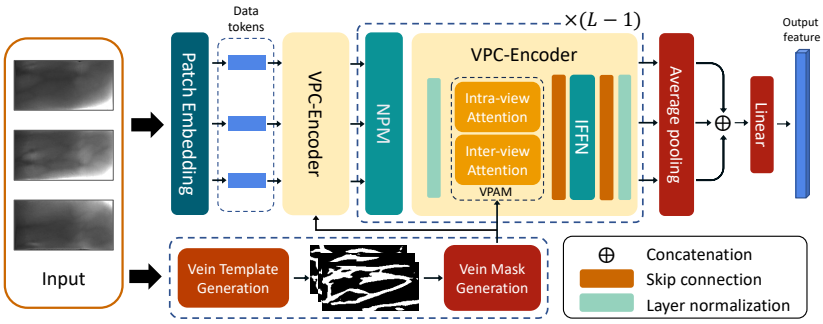


Figure 4: Overview of VPCFormer.

Former is mainly stacked by several VPC-Encoders and NPMs. In detail, a VPC-Encoder consists of a VPAM and an IFFN. With an introduction of vein mask, VPAM aims to facilitate interactions between vein information through the intra-view and inter-view attention. The IFFN is utilized to effectively aggregate preceding attention and contextual information. Furthermore, NPM is designed to capture pixel correlations within a local region. The overall architecture of VPCFormer is shown in Fig. 4.

Given the input multi-view finger images $\left\{\mathbf{I}^i\right\}_{i=1}^{U}$, where U denotes the number of views. Each view $\mathbf{I}^i$ is transformed into a data token sequence $\mathbf{X}_0^i \in \mathbb{R}^{\frac{HW}{p^2} \times d}$ by the patch embedding layer. Here, p signifies the patch size, H and W represent the height and width of the input image respectively, and d denotes the token dimension. For the l-th VPC-Encoder, its output of the i-th view is represented as $\mathbf{X}_l^i$. In the end, VPCFormer extracts the multi-view finger vein feature $\boldsymbol{f}$ for recognition.

5.2. Vein Mask Generation

To constrain the model's attention to vein patterns in the self-attention module, it is necessary to acquire the corresponding vein mask for each view.

Initially, we need to obtain vein template to indicate the positions of the vein pixels. To date, none of the existing finger-vein databases contain ground-truth for vein segmentation, with the exception of the THU-FVS [11] database. Additionally, manually segmenting and annotating all samples in a database is not only time-consuming and labor-intensive but also detrimental to the generalization of algorithms. Drawing inspiration from the previous work [10], the outputs of

some existing algorithms are employed as vein templates, to mitigate the influence of a single method. Four reliable methods are adopted, including LMC [8], Kumar et al. [28], EMC [9], and EIFNet [11], with the latter being the only one trained on the THU-FVS [11] database. In the fusion process, a majority voting approach is employed: a pixel is designated as a vein point in the fused vein template if it is identified as such by three or more algorithms; otherwise, it is labeled as a background point. When generating the fused vein template $\mathbf{T}$, the value at the position identified as a vein point is set to 1, and 0 otherwise. Fig. 5 depicts this process.

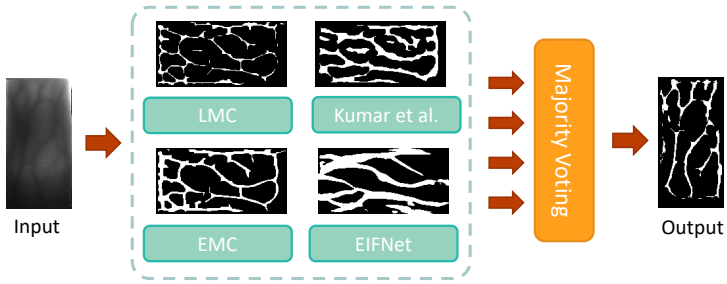


Figure 5: Vein template generation process.

Subsequently, a vein mask consisting of $\frac{HW}{p^2}$ elements is derived from a vein template. Specifically, a square kernel of size $p \times p$ with all element values set to 1 is employed. The kernel is convolved with the vein template using a stride of p. The convolution result is denoted as $\hat{\mathbf{M}}$, and the aforementioned operation can be expressed as

$$\hat{\mathbf{M}}(i,j) = \sum_{m=0}^{p-1} \sum_{n=0}^{p-1} \mathbf{T}\left(p \cdot i + m, p \cdot j + n\right), \tag{1}$$

where $\mathbf{T}(i,j)$ represents the value of position (i,j). Thus, the vein mask $\mathbf{M}$ can be computed as

$$\mathbf{M}(i,j) = \begin{cases} 1 & \text{if } \hat{\mathbf{M}}(i,j) \geq \zeta, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where ζ is a predefined threshold to avoid errors caused by a small number of vein pixels.

5.3. VPC-Encoder

This section details VPCFormer, which consists of two primary components: a VPAM and an IFFN.

5.3.1. VPAM

The self-attention module excels in capturing global information and long-range dependencies. However, applying this module directly to multi-view inputs essentially allows the model to learn all conceivable long-range dependencies autonomously. As aforementioned, due to the limited training samples and weak long-range correlations between background across different views, we design VPAM based on the Multi-Head Self-Attention (MHSA), to extract the correlations within and between views of vein patterns. The overall structure of the VPAM is depicted in Fig. 6.

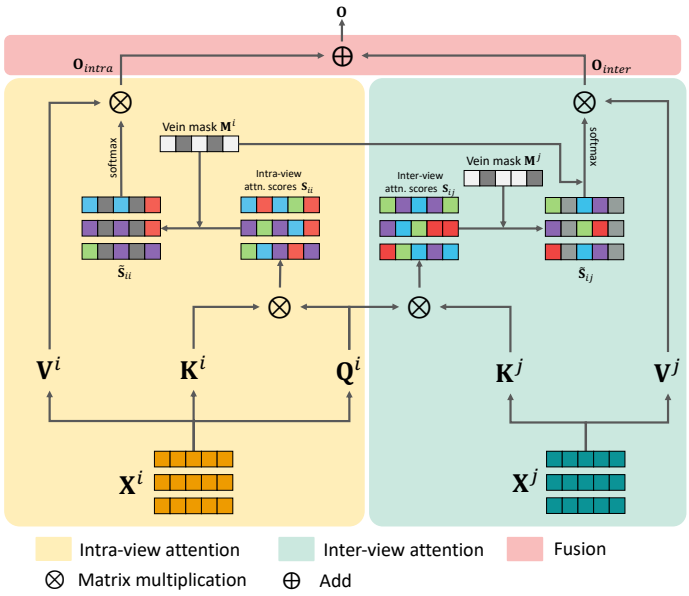


Figure 6: The overall structure of VPAM. For clarity and ease of illustration, this figure depicts operations within a single attention head. Operations within other attention heads are identical to those shown.

Extraction of intra-view correlations: For the data token $\mathbf{X}^i$ of the i-th view, the corresponding query $\mathbf{Q}^i$, key $\mathbf{K}^i$, and value $\mathbf{V}^i$ are calculated as

$$\begin{aligned} \mathbf{Q}^i &= \mathbf{X}^i\mathbf{W}_q, \\ \mathbf{K}^i &= \mathbf{X}^i\mathbf{W}_k, \\ \mathbf{V}^i &= \mathbf{X}^i\mathbf{W}_v, \end{aligned} \tag{3}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are the learnable weight matrices with the dimensions of $d \times d_{head}$. Based on these three vectors, the attention score matrix $\mathbf{S}_{ii}$ of the i-th view can be computed as $\mathbf{S}_{ii} = \mathbf{Q}^i \cdot \mathbf{K}^{i\top}$. In order to constrain the model's attention to vein patterns, we utilize the vein mask $\mathbf{M}^i$ generated in Section 5.2 to set the correlation scores in $\mathbf{S}_{ii}$ related to the background tokens to negative infinity[4], i.e.,

$$\tilde{\mathbf{S}}_{ii} = -\varepsilon(1 - \mathbf{M}^i) + \mathbf{S}_{ii}, \tag{4}$$

where ε denotes a very large positive number. In this paper, $\varepsilon = 2^{32} - 1$. Next, we use the $softmax$ function to convert $\tilde{\mathbf{S}}_{ii}$ into a probability distribution $\tilde{\mathbf{P}}_{ii}$:

$$\tilde{\mathbf{P}}_{ii} = softmax\left(\frac{\tilde{\mathbf{S}}_{ii}}{\sqrt{d_k}}\right), \tag{5}$$

where d_k is the dimension of $\mathbf{K}^i$. At this point, $\tilde{\mathbf{P}}_{ii}$ only retains the correlations between different positions and the vein patterns within i-th view, while ignoring their connections with background areas. Finally, based on the probability distribution $\tilde{\mathbf{P}}_{ii}$, the output of the intra-view self-attention $\mathbf{O}^i_{intra}$ is calculated as

$$\mathbf{O}^i_{intra} = \text{Attention}(\mathbf{Q}^i, \mathbf{K}^i, \mathbf{V}^i) = \tilde{\mathbf{P}}_{ii} \cdot \mathbf{V}^i. \tag{6}$$

Extraction of inter-view vein correlations: The process of extracting correlations between different views is similar to that within the single view. For the given data tokens $\mathbf{X}^i$ and $\mathbf{X}^j$ from two views, we need to multiply the query $\mathbf{Q}^i$ from view i with the key $\mathbf{K}^j$ from view j to calculate the attention scores between different positions in view i and view j, denoted as $\mathbf{S}_{ij} = \mathbf{Q}^i \cdot \mathbf{K}^{j\top}$. For the c-th row (denoted as $\mathbf{S}_{c*}$) in $\mathbf{S}_{ij}$, it essentially represents the attention scores of the c-th token in $\mathbf{X}^i$ with all tokens in $\mathbf{X}^j$. Similarly, we need to limit the attention areas in view j. Therefore, by using the vein mask $\mathbf{M}^j$ from view j to block out the model's attention to the non-vein areas, similarly to Eq. (4), we have

$$\tilde{\mathbf{S}}_{ij} = -\varepsilon(1 - \mathbf{M}^j) + \mathbf{S}_{ij}. \tag{7}$$

At this point, the values within $\tilde{\mathbf{S}}_{c*}$ denote the attention scores of the c-th token in $\mathbf{X}^i$ when correlated with all tokens in $\mathbf{X}^j$ that signify vein patterns. As previously highlighted, the key objective

[4]Setting it to negative infinity has the advantage of directly outputting a probability of zero for positions where correlation is not desired when using the $softmax$ function later.

of inter-view attention operations is to capture correlations exclusive to the tokens contained vein patterns across different views. Evidently, if the c-th token in $\mathbf{X}^i$ signifies a background area, $\tilde{\mathbf{S}}_{c*}$ would represent the correlation between a background token in $\mathbf{X}^i$ and all vein tokens in $\mathbf{X}^j$. Consequently, it becomes imperative to block the attention scores in $\tilde{\mathbf{S}}_{ij}$ corresponding to the background tokens in $\mathbf{X}^i$. However, direct manipulation of $\tilde{\mathbf{S}}_{ij}$ would necessitate supplementary operations on the $softmax$ output. Therefore, we manage the probability distribution output from $softmax$ directly as

$$\tilde{\mathbf{P}}_{ij} = \left(\mathbf{M}^i \cdot \mathbf{1}^\top\right) \odot softmax\left(\frac{\tilde{\mathbf{S}}_{ij}}{\sqrt{d_{kj}}}\right), \tag{8}$$

where $\odot$ is Hadamard product; $\mathbf{1} \in \mathbb{R}^{\frac{HW}{p^2} \times 1}$ represents a column vector with all elements set to 1; $\mathbf{M}^i \in \mathbb{R}^{\frac{HW}{p^2} \times 1}$ denotes the vein mask for view i, as procured in Section 5.2. At this stage, $\tilde{\mathbf{P}}_{ij}$ exclusively preserves the connections between the vein tokens in views i and j. Ultimately, we compute the output of the inter-view attention operation, $\mathbf{O}^{ij}_{inter}$, as

$$\mathbf{O}^{ij}_{inter} = \tilde{\mathbf{P}}_{ij} \cdot \mathbf{V}^j. \tag{9}$$

Output fusion: Once we have obtained the two types of attention outputs $\mathbf{O}^i_{intra}$ and $\mathbf{O}^{ij}_{inter}$, we integrate them by directly adding them together to derive the ultimate output of VPAM, i.e.,

$$\mathbf{O}^i = \mathbf{O}^i_{intra} + \mathbf{O}^{ij}_{inter}. \tag{10}$$

5.3.2. IFFN

In ViT [12], tokens encapsulating raw attention information undergo two layers of MLP to facilitate feature learning. Despite this, our observations suggest that the conventional Feed-Forward Network (FFN) falls short in the task of vein feature extraction, thereby causing a significant performance decrement (as discussed in Section 6.5.2). In response to this, we put forth an IFFN aimed at enhancing the model's performance.

Specifically, in order to adapt to image data and more effectively aggregate preceding attention, we incorporate pointwise convolution based on a 1×1 kernel, as well as depthwise convolution utilizing a 3×3 kernel. The detailed structure of IFFN is visually represented in Fig. 7.

In IFFN, we first spatially rearrange the token $\mathbf{O}^{i\prime} \in \mathbb{R}^{\frac{HW}{p^2} \times d}$[5] that contains intra-view and inter-view attention information to obtain the view feature map $\mathbf{F}^i \in \mathbb{R}^{d \times \frac{H}{p} \times \frac{W}{p}}$. Next, a pointwise

[5]It is worth noting that $\mathbf{O}^{i\prime}$ is the result of $\mathbf{O}^i$ after the residual connection layer, i.e., $\mathbf{O}^{i\prime} = \mathbf{O}^i + \mathbf{X}^i$.

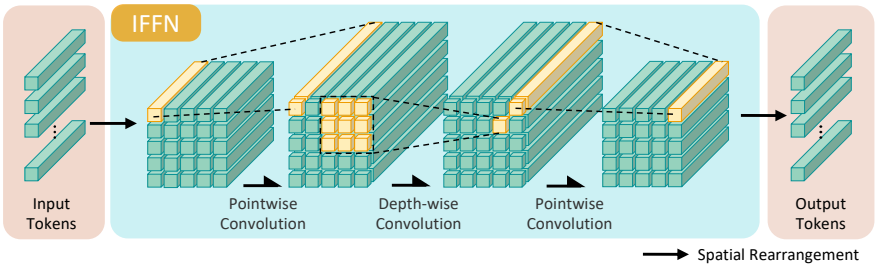


Figure 7: The structure of IFFN.

convolution is utilized to augment the channel of $\mathbf{F}^i$ from d to $4d$. This technique enhances its representation capability in the high-dimensional space. Following this, we implement depth-wise convolution on its output, which facilitates the aggregation of attention and contextual information without a substantial increase in the number of parameters. Ultimately, a pointwise convolution is employed once more to revert the channels back to d, serving as the output of the IFFN after spatial rearrangement.

5.4. Neighborhood-Perspective Module (NPM)

In VPCFormer, a VPC-Encoder based on MHSA is designed to capture the interconnections of vein patterns both within and across views, which can be perceived as global features. To compensate for the model's deficiency in capturing local neighborhood correlations, the NPM is designed to fulfill this objective.

Convolutional operations inherently possess the ability to capture the correlations between image pixels within a local neighborhood. This property can be adjusted by configuring the size of the convolution kernel or by utilizing varying numbers of convolutional layers. To this end, we construct the NPM using two convolutional layers with a 3×3 kernel size, which are designed to capture pixel correlations within a 5×5 neighborhood. The rationale behind not directly employing a 5×5 convolutional layer is to introduce additional nonlinearity while simultaneously reducing parameters. A schematic diagram of the NPM is depicted in Fig. 8.

For the input of NPM, data tokens from each view are rearranged into a spatial structure that mirrors the original image grid, allowing us to employ convolution operations effectively.

Figure 8: The structure of NPM.

5.5. Patch Embedding Layer and Output Layer

Patch embedding layer: To process images using Transformer-based models, images are first fed into a patch embedding layer, which converts each non-overlapping patch into a token embedding. In VPCFormer, we borrow the patch embedding layer from Visformer [29] to achieve a balance between the number of parameters and overall performance.

Output layer: After passing through $L-1$ NPMs and L VPC-Encoders, the data tokens of all views, denoted as $\left\{\mathbf{X}_L^i\right\}_{i=1}^U$, are obtained. The next step is to fuse $\left\{\mathbf{X}_L^i\right\}_{i=1}^U$ in order to derive a multi-view finger vein feature $\boldsymbol{f}$ for recognition. Specifically, for the i-th view, the average value is calculated for each position across $\mathbf{X}_L^i$ to obtain the view representation $\boldsymbol{f}^i$:

$$\boldsymbol{f}^i = \frac{1}{\frac{HW}{p^2}} \sum_{j=1}^{\frac{HW}{p^2}} \mathbf{x}_j^i, \tag{11}$$

where $\mathbf{x}_j^i$ denotes the jth token of $\mathbf{X}_L^i$. The averaging operation helps mitigate the influence of noise while preserving global information. To map the concatenated view representations $[\boldsymbol{f}^1; ...; \boldsymbol{f}^U]$ to the target feature space, a linear layer is utilized to learn the latent relationship between view representations and ultimately obtain the multi-view finger vein feature $\boldsymbol{f}$ for recognition. This process can be expressed as

$$\boldsymbol{f} = Linear\left(Concat\left(\boldsymbol{f}^1, ..., \boldsymbol{f}^U\right)\right). \tag{12}$$

For the extracted features, we utilized the nearest neighbor (1-NN) classifier, adopting cosine similarity as the distance metric for finger vein recognition.

6. Experiments

In this section, we conduct extensive experiments to provide a benchmark based on our THUMVFV-3V database. Furthermore, VPCFormer is evaluated to demonstrate its superior-

ity.

6.1. Experiment Settings

In the experiments, we evaluate three types of methods on the THUMVFV-3V database, i.e., traditional methods, subspace learning-based methods and deep learning-based methods.

In order to guarantee fair and reliable comparisons, we have standardized the preprocessing operations for different types of methods[6]. This unification encompasses a range of factors including image preprocessing, ROI type, input size, and data augmentation. Furthermore, any methods with hyperparameters have been meticulously tuned to ensure the attainment of optimal results.

Regarding the programming languages and experimental platforms utilized, all traditional methods are executed in MATLAB, with the input ROIs sequentially undergoing median filtering and Contrast Limited Adaptive Histogram Equalization (CLAHE). Likewise, all subspace learning-based methods are implemented in MATLAB, with all ROIs resized to 64×144 without additional preprocessing operations. The involved deep-learning methods are tailored for feature extraction by removing the last classification layer and replacing the softmax loss with the circle loss [30]. All traditional methods and subspace learning-based methods are executed on a Windows workstation, equipped with Windows 10, 64GB of memory, and an Intel(R) Xeon(R) CPU E5-2695 v2 @2.40GHz processor. In contrast, all deep learning-based methods, implemented in Python under the PyTorch framework, are executed on four Nvidia GeForce GTX 1080Ti GPUs, each with 11GB of memory.

For the experiments, unless otherwise specified, we employed a consistent strategy for data partitioning: the samples of all classes collected during the first session were used exclusively for training, and those collected in the second session were strictly used for evaluation.

6.2. Single-view Recognition

THUMVFV-3V can be naturally separated into three single-view finger vein databases. In this section, we study the performance of different methods under different views. Three types

[6]This unification may yield differences from the original papers, but for the selected methods, the preprocessing is entirely separated from the method itself. Therefore, while ensuring the integrity of the algorithm, the impact of preprocessing operations is considered uniform across all methods of the same type.

of methods are re-implemented to conduct a comprehensive analysis of the performance, such as traditional-based (i.e., LDC [31], Kumar et al. [28], WLD [6], LMC [8], and PWBDC [32]), subspace learning-based (i.e., WSRC [33], ESRC [34], DDBPD [35], and LCMFC [36]), and deep learning-based (i.e., ResNet 18/50 [37], attention models [38, 39, 40, 12], FVCAE [41], Hong et al. [42], ArcVein [3], and MRFBCNN [43]). Note that some subspace learning-based methods are borrowed from other biometrics. It is fair and reasonable because there is no biometric-specific prior is incorporated into these methods. The identification performance is reported in Table 2, which is evaluated by accuracy (ACC) [44] as

$$\text{ACC}\ (\%) = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}} \times 100\%. \tag{13}$$

The results presented in Table 2 lead us to several conclusions.

Firstly, concerning the performance of methods for each view, the results demonstrate that all methods perform optimally at the 0° view. This suggests that the finger vein images from the 0° view encompass more informative features. This inference aligns not only with our expectations but also consistent with conclusions of other studies [20]. This is because that images from the 0° view contain more valid vein patterns than those in the $\pm 45^\circ$ views.

Secondly, among the five traditional methods, we found that works based on directional features (PWBDC [32], Kumar et al. [28], and LDC [31]) exhibit commendable performance across all views. It also can be found that these three methods have a smaller performance decline at the $\pm 45^\circ$ views compared to the 0° view. This suggests that directional features can yield robust and distinctive information for the vein recognition task.

Thirdly, among the methods based on the subspace learning, the methods (DDBPD [35] and LCMFC [36]) integrating additional constraints achieved better results.

Fourthly, deep learning-based methods have shown quite impressive results in this task. It is clear to see that classic ResNets [37] still perform remarkably. Considering the outcomes derived from ResNet18 [37] and ResNet50 [37], one can infer that an increase in network depth brings a certain improvement to the results given the current data volume. For CNN-based models, it is noticeable that works incorporating attention mechanisms, such as MRFBCNN [43] and ECANet [40], have aided in improving performance at the $\pm 45^\circ$ views. Nevertheless, such improvement does not consistently benefit from all attention mechanisms. Intriguingly, the latest model based on the Transformer architecture, ViT [12], performs on par with ResNet18 [37]

Table 2: Single-view finger vein identification performance (%) for each view on THUMVFV-3V database. The best is in bold; the second best is underlined; the third best is in italics.

Type	Methods	Year	−45°	0°	+45°
Traditional	LMC [8]	2007	69.22	82.45	67.02
	WLD [6]	2010	77.90	86.09	70.30
	LDC [31]	2012	84.39	91.11	84.44
	Kumar et al. [28]	2012	83.74	89.14	84.72
	PWBDC [32]	2019	85.08	87.15	80.98
Subspace Learning	ESRC [34]	2012	34.92	42.25	32.60
	WSRC [33]	2019	33.94	41.92	31.72
	DDBPD [35]	2018	78.26	89.24	76.29
	LCMFC [36]	2020	85.15	92.85	82.78
Deep Learning	ResNet 18 [37]	2016	86.79	*96.87*	*87.90*
	ResNet 50 [37]	2016	<u>90.24</u>	**97.40**	**90.40**
	Hong et al. [42]	2017	78.44	89.55	79.55
	SENet [38]	2018	85.83	96.16	86.99
	SKNet [39]	2019	78.19	88.26	74.80
	FVCAE [41]	2019	81.44	89.97	76.31
	ECANet [40]	2020	*87.70*	95.54	86.48
	ArcVein [3]	2021	82.42	94.32	76.65
	MRFBCNN [43]	2021	**91.08**	<u>97.28</u>	<u>90.24</u>
	ViT [12]	2021	87.37	96.86	86.77

across all views. This suggests that Transformer-based architectures hold considerable potential for the current task, indicating a promising direction for future research.

6.3. Multi-view Recognition

Close-set protocol: In this section, we study the performance of multi-view finger vein recognition. Multi-view finger vein recognition refers to complete identity authentication by using several finger vein images from different views simultaneously.

In fact, most single-view based methods can be extended to multi-view scenario by adopting different fusion strategy [44]. In order to balance robustness and fairness, we adopt a score fusion strategy to extend some single-view methods to adapt multi-view recognition task. Due to lim-

ited multi-view finger vein recognition methods [18, 14], we borrowed some multi-view learning methods from other research field for comparison. These methods include MVCNN [16], RotationNet [45], CVR [46], view-GCN [47] and OVPT [48]. Table 3 reports the identification accuracy of different methods on the THUMVFV-3V database, and Fig. 9 plots the DET curves.

Table 3: Multi-view finger vein identification performance.

Type	Methods	Year	ACC (%)
Extensions of Single-view methods	PWBDC [32]	2019	91.26
	Kumar et al. [28]	2012	90.88
	WLD [6]	2010	95.45
	LMC [8]	2007	88.46
	LDC [31]	2012	92.73
	ResNet 50 [37]	2016	99.02
	ECANet [40]	2020	98.91
	Hong et al. [42]	2017	91.97
	MRFBCNN [43]	2021	98.94
	ArcVein [3]	2021	96.97
	ViT [12]	2021	98.41
Multi-view methods	MVCNN [16]	2015	98.81
	RotationNet [45]	2018	*99.62*
	view-GCN [47]	2020	<u>99.65</u>
	CVR [46]	2021	98.99
	HCAN [18]	2022	99.44
	OVPT [48]	2023	99.02
	VPCFormer	2023	**99.79**

The experimental results delineated in Table 3 and Fig. 9 not only corroborate some of the conclusions derived in Section 6.2, but also offer several new findings.

Firstly, for methods based on single view fusion, the average ACC of traditional methods is 91.75%, while deep learning methods average at 97.37%. This result validates that contemporary deep learning-based models typically outperform traditional methods reliant on handcrafted features. This superiority might be attributed to the fact that deep models can autonomously learn distinctive features from training data and its impressive generalization capabilities. Mean-

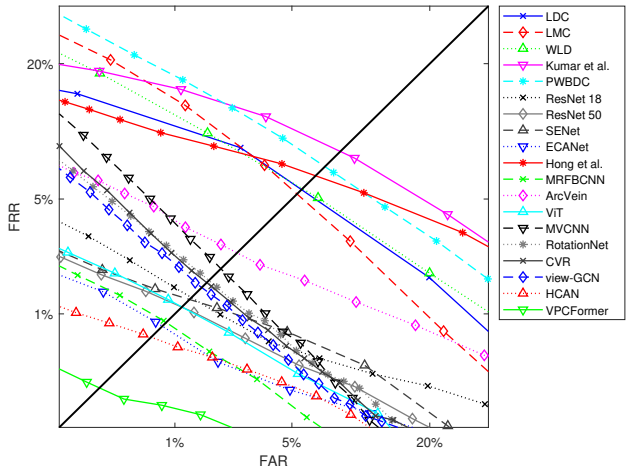


Figure 9: DET curves of different extensions of single-view methods and multi-view methods on the THUMVFV-3V database.

while, it can be found that the average ACC of the four methods based on multi-view input is 99.38%. This suggests that, compared to the amalgamation of multiple single-view results, the correlations between different views can serve as additional features to significantly enhance the performance.

Secondly, compared with the result at the 0° view, the multi-view recognition performance achieved through score fusion exhibits an overall improvement. This can be attributed to the enriched discriminative features provided by multiple views. Interestingly, the methods based on directional information, such as PWBDC [32], LDC [31] and Kumar et al. [28], do not gain much performance benefit from score fusion. Potential explanations include the paucity of additional directional information due to the absence of veins in the ±45° views, and the diminished marginal benefits caused by the strong performance at the 0° view.

Thirdly, among all the methods evaluated, our VPCFormer consistently delivers the best performance in terms of both identification task and verification task. Compared to the other methods, the DET curve of VPCFormer is markedly closer to the lower-left corner. This indicates that VPCFormer maintains a lower False Accept Rate (FAR) while ensuring a low False Reject Rate (FRR), achieving a good trade-off between these two types of errors. This superior

performance can be attributed to VPCFormer's ability to effectively capture the global vein pattern correlations across different views and the local relationship between veins and backgrounds within a single view, achieved through the incorporation of VPAM and NPM, respectively.

Open-set protocol: In this part, we assessed multi-view deep learning-based models within an open-set scenario, critical for authentic biometric recognition systems. We partitioned the THUMVFV-3V database, allocating 80% of the classes (528) for training and the remaining 20% (132) for evaluation. What is more, the evaluation involved 132 unseen classes with enrollment samples from the first session and probe samples from the second session.

The comparative results are presented in Table 4.

Table 4: Multi-view finger vein recognition performance under open-set protocol

Methods	Year	ACC (%)	EER (%)
MVCNN [16]	2015	95.91	2.65
RotationNet [45]	2018	96.54	2.25
view-GCN [47]	2020	97.27	1.75
CVR [46]	2021	97.73	1.93
HCAN [18]	2022	99.39	0.80
OVPT [48]	2023	*98.48*	*1.45*
VPCFormer	2023	**99.65**	**0.48**

By comparing with the performance (in Table 3) of methods under the closed-set protocol, Table 4 reveals a diminished performance for multi-view methods in the open-set scenario. Notably, VPCFormer surpasses competing models.

6.4. Cross-view Recognition

In this section, we investigate cross-view finger vein recognition, where the probe image and the gallery image originate from different views. This experiment serves to study the performance degradation when finger rotation occurs during testing.

Deep learning-based methods are tested in this section. The decision to exclude other types of methods stems from their reliance on handcrafted features. Such features are not only non-robust to rotation but also potentially time-consuming during the matching process. Specifically, the algorithms initially train on the 0° view, as detailed in Section 6.2, followed by separate

testing on the ±45° views. The division of training and test sets remains consistent with the approach previously outlined. The results are organized in Fig. 10.

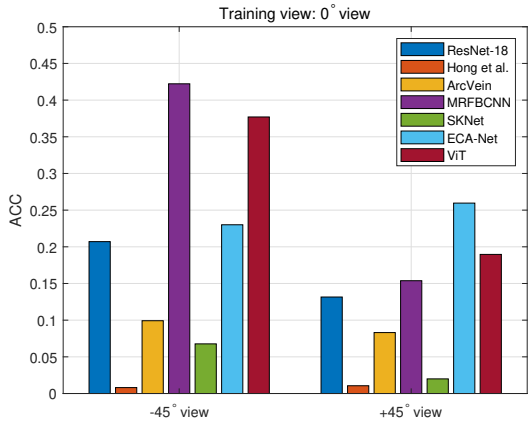


Figure 10: Identification results of cross-view recognition.

As can be seen from Fig. 10, compared to single-view recognition, the performances of cross-view recognition have significantly declined. The primary reason is the stark difference between the test and training images of the same class. Despite the overlap between two adjacent views in this experiment, the positions and shapes of the relevant veins do not correlate through simple translation. Rather, they are 2D projections of an elliptical surface [6].

In addition, except for the method of Hong et al. [42], all methods managed to correctly identify a portion of cross-view samples, with ACC ranging from 5% to 43%. This indicates that even when different views with overlap regions are employed for training and testing, some deep learning-based methods can achieve acceptable performance. This efficacy stems from the strong inference capability and generalization intrinsic to deep learning methods. We posit that with further data expansion, the performance of deep learning-based methods in cross-view recognition could see continued improvement.

6.5. Further Experiments and Discussion

6.5.1. The Influence of the Number of Views on Multi-view Recognition Results

In multi-view finger vein recognition, the number of views utilized for training or fusion substantially influences the outcomes. In this section, we investigate this impact by varying the

number of fused views. The corresponding experimental results are presented in Table 5.

Table 5: The identification performance of different methods with different views combination (%). The best is in bold; the second best is underlined; the third best is in italics.

Methods	Input views				
	0°	0°&-45°	0°&+45°	-45°&+45°	All views
LMC [8]	82.45	85.91	84.92	80.10	88.46
WLD [6]	86.09	91.19	93.11	90.10	95.45
LDC [31]	91.11	92.20	92.13	89.49	92.73
Kumar et al. [28]	89.14	90.05	90.10	87.55	90.88
PWBDC [32]	87.15	88.79	90.43	89.02	91.26
ResNet18 [37]	*96.87*	98.03	97.93	94.24	98.31
ResNet50 [37]	**97.40**	*98.64*	97.90	96.74	*99.02*
Hong et al. [42]	89.55	91.41	90.91	86.52	91.97
SENet [38]	96.16	98.23	98.26	96.24	98.94
SKNet [39]	88.26	94.52	93.84	91.74	96.39
FVCAE [41]	89.97	96.34	95.05	93.84	96.87
ECANet [40]	95.54	98.48	98.66	96.57	98.91
MRFBCNN [43]	<u>97.28</u>	98.43	98.51	97.05	98.94
ArcVein [3]	94.32	95.96	96.52	91.41	96.97
ViT [12]	96.86	98.33	98.03	93.79	98.41
HCAN [18]	-	<u>98.90</u>	<u>99.07</u>	*97.61*	<u>99.44</u>
OVPT [48]	-	98.63	*98.81*	<u>97.62</u>	*99.02*
VPCFormer	-	**99.12**	**99.56**	**97.83**	**99.79**

Table 5 reveals that all methods yield the optimal identification results when incorporating three views. This observation suggests that the inclusion of more informative views can indeed enhance the distinctiveness of the features. Considering the view positions, the results for the combination of the ±45° views, in most cases, are lower than those for the combination of the 0° and ±45° views, even underperforming the 0° view input solely. This phenomenon may be attributed to the fact that the side views contain less robust and informative features.

Additionally, it can be seen that the improvement in ACC with three views is not significant on some algorithms. Given that identification performance is merely one metric in recognition tasks, to depict the influence of the number of views more holistically and reliably, we selected

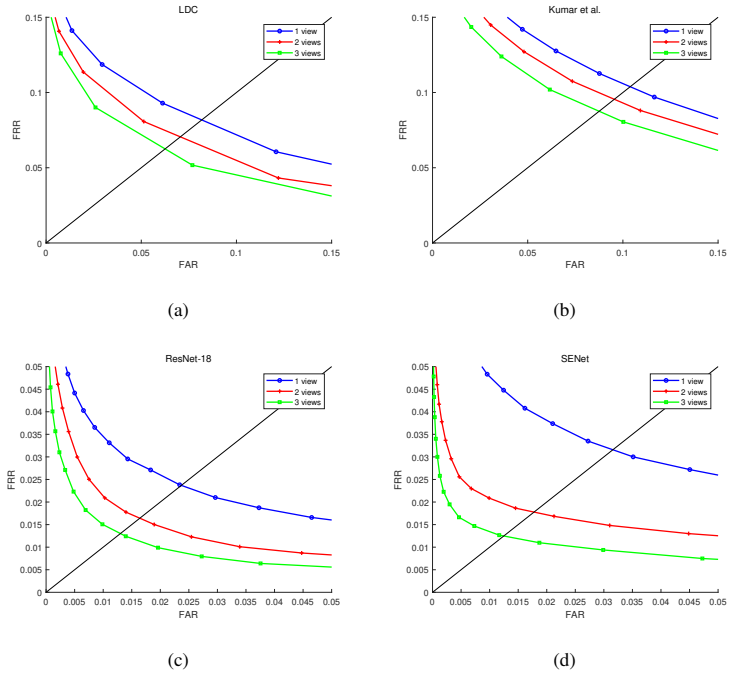


Figure 11: DET of several methods with different number of views as input.

some algorithms exhibiting relatively minor improvements in ACC and illustrated their DET curves for the vein verification task in Fig. 11. It can be seen that the DET curves of these methods show a noticeable shift towards the lower-left corner with the inclusion of more views. This suggests that more views indeed bring benefits to finger vein recognition, achieving lower error rates at a given threshold. Furthermore, it is noteworthy that the marginal gain notably diminishes when utilizing three views. Consequently, it is necessary to judiciously select the number of input views to strike an optimal balance between model performance and model complexity.

6.5.2. Ablation Studies of VPCFormer

To further verify the effectiveness of the vein mask and the proposed modules in VPCFormer, we conducted a series of ablation studies. The corresponding results are presented in Table 6.

Several conclusions can be inferred from Table 6. Firstly, when juxtaposed with the conventional FFN, the IFFN proposed in this study markedly augments the model's expressive capacity,

Table 6: The results of ablation experiments.

NPM	Inter-view attn.	IFFN	Vein mask	EER (%)	ACC (%)
				7.62	80.50
✔				1.82	98.11
✔		✔		0.50	98.93
✔	✔	✔		0.41	99.31
	✔	✔	✔	0.39	99.38
✔	✔	✔	✔	**0.27**	**99.79**

reducing the Equal Error Rate (EER) by 1.32%. This outcome can be ascribed to the fact that the convolutional operation, being more compatible with image data, can more effectively aggregate the contextual information of tokens.

Secondly, in comparison to constraining attention independently within each view, implementing an inter-view attention mechanism can substantially enhance the model's performance. Even though the entire view image is directly considered when capturing inter-view correlations, the model is permitted to independently infer the interrelation between different views.

Thirdly, upon further incorporation of the vein mask, the inter-view attention of the VPAM is constrained within the areas contained vein patterns, resulting in an approximate 0.14% enhancement in the EER. This indicated that capturing the inter-view vein pattern correlations directly can reduce the difficulty of feature extraction while simultaneously boosting the model's performance.

Finally, NPM is employed to capture local correlations within a single view, thereby compensating for the limitations introduced by the vein mask. By integrating global inter-view vein correlations and local features, VPCFormer achieves superior results.

6.5.3. Experiments about VPCFormer Hyperparameters.

Similar to other models based on ViT [12], VPCFormer requires the predefinition of several hyperparameters, which include the image size $[H, W]$, patch size p, the token dimension d, the depth L of the VPC-Encoder (refer to Fig. 4), the number of attention heads h in VPAM, and the dimension d_{head} of each attention head. We primarily referenced the hyperparameter settings of ViT-Tiny [12] and ViT-Small [12] to conduct parameter experiments for VPCFormer. Experimental results are shown in Table 7.

Table 7: Results of VPCFormer with different hyperparameters.

No.	$[H, W]$	L	d	p	h	d_{head}	EER (%)	ACC (%)
1	[224, 224]	6	192	[16, 16]	12	32	0.86	99.22
2	[224, 224]	6	384	[16, 16]	12	32	0.34	99.75
3	[224, 224]	12	384	[16, 16]	12	32	**0.27**	**99.79**
4	[224, 224]	12	384	[16, 16]	8	32	0.41	99.57
5	[224, 224]	12	384	[16, 16]	8	64	0.39	99.70
6	[224, 224]	12	384	[16, 16]	12	64	**0.27**	**99.79**
7	[112, 112]	12	384	[16, 16]	12	32	5.54	89.14
8	[112, 112]	6	192	[8, 8]	12	32	0.87	99.19
9	[112, 112]	12	192	[8, 8]	12	32	0.66	99.29
10	[112, 112]	12	384	[8, 8]	12	32	0.63	99.52

Results from experiments 2 vs. 3 (or 8 vs. 9) suggest that increasing the depth L only can lead to improved performance. A similar conclusion can be drawn for the parameter d based on comparison experiments 1 vs. 2 (or 9 vs. 10). Regarding the choice of patch size p, setting it to $[16, 16]$ for an input size of 224×224, and $[8, 8]$ for an input size of 112×112, yields the same number of tokens $\frac{HW}{p^2}$. According to the outcomes of experiments 7 vs. 10, when the patch size is relatively large for an input size, the performance of the VPCFormer noticeably declines. This decrease in performance might stem from VPCFormer's inadequacy in learning robust contextual relationships from a reduced number of input tokens. This also places higher demands on the patch embedding layer's ability to infer patch tokens. Additionally, less tokens could result in more vein patterns within each patch (particularly in vein-rich images), which might render the vein mask ineffective (all values equating to 1). Comparison between experiments 3 vs. 4 (or 5 vs. 6) indicates that superior results are achieved when $h = 12$. Concurrently, setting all other parameters optimally (experiment 3 and 6), the best outcomes are attained when parameter d_{head} is designated either 32 or 64.

In conclusion, to attain peak performance, we finally configure the relevant parameters of the VPCFormer model outlined in experiment 3 in Table 7.

6.5.4. Discussion about Global Tokens

In ViT [12], the input is concatenated with a class token after the patch embedding layer. The authors argue that the class token serves as a global representation. Only the class token is employed for feature extraction or classification. Contrastingly, some studies opt not to utilize the class token. Instead, these investigations conduct global average pooling on the data tokens to generate global features [29]. Additionally, in certain studies based on multi-view inputs, the 'view token' concept has been proposed [48]. To put it succinctly, the view token solely abstracts information from a specific view, which can be regarded as the feature of the corresponding view.

In order to evaluate the effectiveness of class tokens or view tokens in VPCFormer, we instituted modifications to the model under distinct scenarios. The specific alterations and designs are delineated below.

- **Class token only:** The inclusion of a class token will disrupt the spatial arrangement of the token sequence. Consequently, the class token does not go through the depthwise convolution layer in NPM and IFFN, nor does it participate in VPAM. To facilitate the exchange of information between the class token and data tokens, we introduce a new MHSA module subsequent to VPAM. This MHSA operation exclusively updates the class token, without modifying data tokens. Ultimately, only the class token is deployed for the representation of the output feature $\boldsymbol{f}$.

- **View tokens only:** Similarly, the incorporation of view tokens disrupts the spatial rearrangement of data tokens within each view. As a view token should only interact with the data tokens from its corresponding view, each view token partakes in the intra-view self-attention of their own view in VPAM by adding an additional mask. The forward propagation direction of view tokens is depicted in Fig. 12. By splitting view and data tokens at appropriate positions, we can prevent any disruption to the data token flow in the model. Ultimately, all view tokens are concatenated directly, fed into a linear layer to obtain the final feature $\boldsymbol{f}$.

- **Both tokens:** As aforementioned, the view token constitutes an abstraction of each individual view, and the class token represents a global feature of all views. When incorporating both types of global tokens simultaneously, we preserve the flow direction of the view token. Additionally, a new MHSA is inserted after VPAM, facilitating interaction

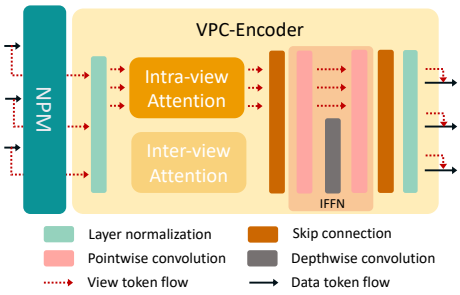


Figure 12: Flow direction of view tokens in the presence of view tokens.

Table 8: The effectiveness of global tokens on VPCFormer.

View Tokens	Class Token	EER (%)	ACC (%)
✘	✘	**0.27**	**99.79**
✔	✘	3.37	88.68
✘	✔	1.74	97.81
✔	✔	49.13	0.25

exclusively between the class token and all view tokens. The final feature $\boldsymbol{f}$ is obtained from the class token.

Based on the aforementioned modifications and designs, we conducted relevant experiments and the corresponding results are presented in Table 8.

From Table 8, it can be seen that the inclusion of either view tokens or class tokens individually has a detrimental impact on performance, with the optimal results achieved when neither is utilized. This could potentially be attributed to the challenges faced by the VPCFormer in extracting distinctive global features with limited training data. Unexpectedly, the concurrent addition of view tokens and class tokens results in the network failing to converge, with the EER persisting around 50%. This situation could be due to the inability of the view tokens to effectively represent the global information of each view. Subsequently, the class token is difficult to capture global features from view tokens. This deficiency leads to convergence difficulties of the model.

6.5.5. Discussion about Positional Encoding

Many studies employing the Transformer architecture incorporate Positional Encoding (PE) to capture the spatial information of input data. In this experiment, we investigated the effect of PE on VPCFormer.

Since the input consists of multi-view images, we devised two strategies, Strategy A and Strategy B, to validate the effectiveness of PE. These strategies are illustrated in Fig. 13.

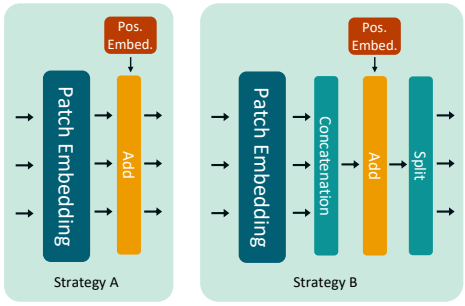


Figure 13: The two strategies for adding PE.

Strategy A: The same sinusoidal PE [12] is uniformly applied to the input tokens of each view, ensuring identical PEs across different view images.

Strategy B: Tokens from different views are concatenated in view order to form a token sequence. This token sequence adds the sinusoidal PE and then they are split back to their respective views in the original sequence. In this case, the PEs for different view images are continuous.

Experimental results, both without PE and employing Strategies A and B, are detailed in Table 9.

Table 9: The impact of different PE strategies.

PE	ACC (%)	EER (%)
✘	99.79	**0.27**
Strategy A	**99.80**	0.30
Strategy B	99.72	0.37

As can be seen, the experimental results indicate that incorporating PE into VPCFormer

yields marginal improvement, and can even slightly reduce performance. This could be attributed to the introduction of convolution, which enables the network to leverage the inherent property of convolutional layers in modeling local spatial relationships, which potentially reducing the effectiveness of PE [49]. Given the model's performance and the uncertainty introduced by PE, VPCFormer was designed without PE.

7. Conclusion

To stimulate advances in multi-view finger vein recognition and establish a benchmark for future research, we propose a multi-view finger vein database, THUMVFV-3V, which is collected over two separate sessions with an average interval of 45.8 days. In addition, to ensure a fair comparison across related methods, we standardized the preprocessing operations in THUMVFV-3V, and provided three types of ROIs and finger masks to accommodate the majority of existing finger vein recognition algorithms. Moreover, we proposed a Transformer-based model, VPCFormer, for multi-view finger vein feature extraction, which is mainly stacked by several VPC-Encoders and NPMs. Equipped with a VPAM and an IFFN, the VPC-Encoder aims to constrain the model's attention solely on the vein patterns and aggregate contextual information, thereby effectively capturing intra- and inter-view correlations. In our experiments, we demonstrate the superiority of VPCFormer in comparison with other single-view and multi-view methods. If we continue to reduce the image patch size, the memory demand of VPCFormer becomes unacceptable. Consequently, future research could focus on the design of the Transformer architecture, exploring ways to decrease memory consumption as the image patch size shrinks and the number of tokens increases, while still ensuring either maintained or enhanced training and inference efficiency.

Acknowledgment

This work was partly supported by the Special Foundation for the development of Strategic Emerging Industries of Shenzhen (No.JCYJ20170817161845824).

References

[1] M. Kono, A new method for the identification of individuals by using of vein pattern matching of a finger, in: Symposium on Pattern Measurement, 2000, pp. 9–12.

[2] L. Yang, G. Yang, Y. Yin, X. Xi, Finger vein recognition with anatomy structure analysis, IEEE Transactions on Circuits and Systems for Video Technology 28 (8) (2018) 1892–1905.

[3] B. Hou, R. Yan, ArcVein: Arccosine center loss for finger vein verification, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–11.

[4] P. Zhao, S. Zhao, J.-H. Xue, W. Yang, Q. Liao, The neglected background cues can facilitate finger vein recognition, Pattern Recognition 136 (2023) 109199.

[5] X. Qiu, W. Kang, S. Tian, W. Jia, Z. Huang, Finger vein presentation attack detection using total variation decomposition, IEEE Transactions on Information Forensics and Security 13 (2) (2018) 465–477.

[6] B. Huang, Y. Dai, R. Li, D. Tang, W. Li, Finger-vein authentication based on wide line detector and pattern normalization, in: International Conference on Pattern Recognition (ICPR), 2010, pp. 1269–1272.

[7] X. Meng, J. Zheng, X. Xi, Q. Zhang, Y. Yin, Finger vein recognition based on zone-based minutia matching, Neurocomputing 423 (2021) 110–123.

[8] 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 E90-D (8) (2007) 1185–1194.

[9] M. A. Syarif, T. S. Ong, A. B. J. Teoh, C. Tee, Enhanced maximum curvature descriptors for finger vein verification, Multimedia Tools and Applications 76 (5) (2016) 6859–6887.

[10] W. Yang, C. Hui, Z. Chen, J.-H. Xue, Q. Liao, FV-GAN: Finger vein representation using generative adversarial networks, IEEE Transactions on Information Forensics and Security 14 (9) (2019) 2512–2524.

[11] Y. Song, P. Zhao, W. Yang, Q. Liao, J. Zhou, EIFNet: An explicit and implicit feature fusion network for finger vein verification, IEEE Transactions on Circuits and Systems for Video Technology 33 (5) (2023) 2520–2532.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations ICLR, 2021, pp. 1–12.

[13] J. Huang, W. Luo, W. Yang, A. Zheng, F. Lian, W. Kang, FVT: Finger vein transformer for authentication, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–13.

[14] W. Kang, H. Liu, W. Luo, F. Deng, Study of a full-view 3D finger vein verification technique, IEEE Transactions on Information Forensics and Security 15 (2020) 1175–1189.

[15] W. Yang, Z. Chen, J. Huang, L. Wang, W. Kang, LFMB-3DFB: A large-scale finger multi-biometric database and benchmark for 3D finger biometrics, in: IEEE International Joint Conference on Biometrics (IJCB), 2021, pp. 1–8.

[16] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3D shape recognition, in: IEEE/CVF International Conference on Computer Vision (ICCV), 2015, pp. 945–953.

[17] W. Yang, J. Huang, Z. Chen, J. Zhao, W. Kang, Multi-view finger vein recognition using attention-based MVCNN, in: Biometric Recognition, Springer Nature Switzerland, 2022, pp. 82–91.

[18] P. Zhao, S. Zhao, L. Chen, W. Yang, Q. Liao, Exploiting multi-perspective driven hierarchical content-aware network for finger vein verification, IEEE Transactions on Circuits and Systems for Video Technology 32 (11) (2022) 7938–7950.

[19] H. Qin, R. Hu, M. A. El-Yacoubi, Y. Li, X. Gao, Local attention transformer-based full-view finger-vein identification, IEEE Transactions on Circuits and Systems for Video Technology 1 (1) (2022) 1–16.

[20] B. Prommegger, C. Kauba, A. Uhl, Multi-perspective finger-vein biometrics, in: IEEE International Conference on

Biometrics Theory, Applications and Systems (BTAS), 2018, pp. 1–9.

[21] L. Lin, H. Liu, W. Zhang, F. Liu, Z. Lai, Finger vein verification using intrinsic and extrinsic features, in: IEEE International Joint Conference on Biometrics (IJCB), 2021, pp. 01–07.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems (NeurIPS) 30 (2017) 1–11.

[23] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning (ICML), 2021, pp. 10347–10357.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9992–10002.

[26] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 568–578.

[27] K. Han, A. Xiao, E. Wu, J. Guo, C. XU, Y. Wang, Transformer in transformer, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems (NeurIPS), 2021, pp. 15908–15919.

[28] A. Kumar, Y. Zhou, Human identification using finger images, IEEE Transactions on Image Processing 21 (4) (2012) 2228–2244.

[29] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, Q. Tian, Visformer: The vision-friendly transformer, in: IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 569–578.

[30] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, Y. Wei, Circle loss: A unified perspective of pair similarity optimization, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 6398–6407.

[31] X. Meng, G. Yang, Y. Yin, R. Xiao, Finger vein recognition based on local directional code, Sensors 12 (11) (2012) 14937–14952.

[32] W. Yang, W. Ji, J.-H. Xue, Y. Ren, Q. Liao, A hybrid finger identification pattern using polarized depth-weighted binary direction coding, Neurocomputing 325 (2019) 260–268.

[33] X. Mei, H. Ma, Finger vein recognition algorithm based on improved weighted sparse representation, in: International Conference on Information Technology and Computer Application (ITCA), 2019, pp. 6–8.

[34] W. Deng, J. Hu, J. Guo, Extended SRC: Undersampled face recognition via intraclass variant dictionary, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (9) (2012) 1864–1870.

[35] L. Fei, B. Zhang, Y. Xu, Z. Guo, J. Wen, W. Jia, Learning discriminant direction binary palmprint descriptor, IEEE Transactions on Image Processing 28 (8) (2019) 3808–3820.

[36] L. Fei, B. Zhang, L. Zhang, W. Jia, J. Wen, J. Wu, Learning compact multifeature codes for palmprint recognition from a single training image per palm, IEEE Transactions on Multimedia 23 (2021) 2930–2942.

[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[38] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: IEEE/CVF Conference on Computer Vision and

Pattern Recognition (CVPR), 2018, pp. 7132–7141.

[39] X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 510–519.

[40] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11531–11539.

[41] B. Hou, R. Yan, Convolutional autoencoder model for finger-vein verification, IEEE Transactions on Instrumentation and Measurement 69 (5) (2020) 2067–2074.

[42] H. Hong, M. Lee, K. Park, Convolutional neural network-based finger-vein recognition using NIR image sensors, Sensors 17 (6) (2017) 1297.

[43] K. Wang, G. Chen, H. Chu, Finger vein recognition based on multi-receptive field bilinear convolutional neural network, IEEE Signal Processing Letters 28 (2021) 1590–1594.

[44] B. Prommegger, C. Kauba, A. Uhl, Different views on the finger: Score-level fusion in multi-perspective finger vein recognition, in: Handbook of Vascular Biometrics, Springer Cham, 2020, pp. 261–305.

[45] A. Kanezaki, Y. Matsushita, Y. Nishida, RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5010–5019.

[46] Y. Xu, C. Zheng, R. Xu, Y. Quan, H. Ling, Multi-view 3D shape recognition via correspondence-aware deep learning, IEEE Transactions on Image Processing 30 (2021) 5299–5312.

[47] X. Wei, R. Yu, J. Sun, View-GCN: View-based graph convolutional network for 3d shape analysis, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1850–1859.

[48] W. Wenju, C. Gang, Z. Haoran, W. Xiaolin, OVPT: Optimal viewset pooling transformer for 3D object recognition, in: Asian Conference on Computer Vision (ACCV), 2023, pp. 486–503.

[49] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, CvT: Introducing convolutions to vision transformers, in: IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 22–31.