




## Research Article

# A Comparative Study of Cross-Device Finger Vein Recognition Using Classical and Deep Learning Approaches

Tuğçe Arıcan <sup>1</sup>, Raymond Veldhuis,<sup>1</sup> Luuk Spreeuwers <sup>1</sup>, Loïc Bergeron,<sup>2</sup> Christoph Busch <sup>2</sup>, Ehsaneddin Jalilian,<sup>3</sup> Christof Kauba,<sup>3</sup> Simon Kirchgasser,<sup>3</sup> Sébastien Marcel,<sup>4</sup> Bernhard Prommegger,<sup>3</sup> Kiran Raja,<sup>2</sup> Raghavendra Ramachandra,<sup>2</sup> and Andreas Uhl<sup>3</sup>

<sup>1</sup>Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, Netherlands

<sup>2</sup>Department of Computer Science, Norwegian University of Science and Technology, Gjøvik, Norway

<sup>3</sup>Department of Computer Sciences, University of Salzburg, Salzburg, Austria

<sup>4</sup>IDIAP Research Institute, Martigny, Switzerland

Correspondence should be addressed to Tuğçe Arıcan; [t.arican@yandex.com](mailto:t.arican@yandex.com)

Received 17 May 2023; Revised 25 January 2024; Accepted 16 February 2024; Published 25 March 2024

Academic Editor: Peter Peer

Copyright © 2024 Tuğçe Arıcan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Finger vein recognition is gaining popularity in the field of biometrics, yet the inter-operability of finger vein patterns has received limited attention. This study aims to fill this gap by introducing a cross-device finger vein dataset and evaluating the performance of finger vein recognition across devices using a classical method, a convolutional neural network, and our proposed patch-based convolutional auto-encoder (CAE). The findings emphasise the importance of standardisation of finger vein recognition, similar to that of fingerprints or irises, crucial for achieving inter-operability. Despite the inherent challenges of cross-device recognition, the proposed CAE architecture in this study demonstrates promising results in finger vein recognition, particularly in the context of cross-device comparisons.

## 1. Introduction

Finger vein recognition is a biometric verification technique that utilises vein patterns to authenticate individuals' identities. Unlike other biometric traits like the face or iris, finger veins are not visible to the naked eye and can only be captured under infrared light. This unique characteristic of finger veins offers several advantages, including increased resistance to stealing or copying. Moreover, since vein patterns do not leave any visible traces (e.g., latent prints), finger vein recognition inherently offers higher privacy compared to other biometric methods, such as faces or fingerprints. Finger vein patterns are also less prone to being affected by external factors such as dirt or oil, and since each finger has a distinct vein pattern, finger vein recognition offers a high level of robustness. All these factors make finger vein patterns as a reliable and promising biometric trait suitable for deploying in challenging and demanding environments.

Finger vein recognition typically involves four steps: image acquisition, pre-processing, feature extraction, and comparison. The feature extraction step aims to derive identity information from finger vein images. Shape, texture, and key point information are commonly employed features in finger vein recognition. Shape-based feature extraction methods, such as curvature methods [1–3], detect vein patterns by using the curvature information in cross-sections of a finger vein image. These methods demonstrate a reliable vein extraction performance under challenging illumination conditions. While shape features focus on only the vein pattern, texture-based features combine finger vein patterns and the finger background by exploring the local relations between image pixels [4, 5]. Since the finger background is susceptible to illumination changes, texture features are more sensitive to illumination conditions compared to curvature methods. Key points of vein patterns, such as bifurcations and end-points, are another type of feature used for finger vein recognition

[6, 7]. These points are tracked and compared to the finger vein images from other individuals. Key point-based methods are less sensitive to translation and rotation compared to shape and texture features. However, due to the low contrast of finger vein images, point extraction does not always yield reliable feature descriptors.

In recent years, deep learning models have emerged as a new approach to finger vein recognition. Unlike classical ones (i.e., hand-crafted features), these methods encapsulate feature extraction and comparison in a neural network architecture, which reduces pre-processing and feature engineering efforts. As the learned representations are derived from data rather than being manually engineered on individual instances, the resulting descriptions are regarded as more resilient to variations such as changes in illumination. Convolutional neural networks (CNNs) are commonly used for finger vein recognition because of their ability to recognise patterns. While initial research on CNNs for finger vein recognition typically builds these models from scratch [8–10], more recent studies adopt a transfer-learning approach due to the limited availability of large finger vein datasets. The transfer-learning approach, which involves leveraging pre-existing networks to perform finger vein recognition, achieves state-of-the-art results on publicly available finger vein datasets [11–13]. Convolutional auto-encoders (CAEs) are another type of promising deep learning approach for finger vein recognition. While CNNs require label information to learn feature representations, CAEs can learn it without the need for label information, which reduces the need for large labelled datasets. Though the early studies using CAEs for finger vein recognition fall short of CNN models, they demonstrate great potential in this field [14–17]. CAEs are also used with traditional methods and CNNs, where they serve as an enhancement method to help the main feature extraction method to achieve more accurate and robust finger vein descriptors [18].

Though the current advancements in finger vein recognition indicate great potential for this biometric trait, there is limited work exploring the inter-operability of finger vein images. Inter-operability of a biometric trait refers to the ability of being recognised and verified across multiple systems. This means that the biometric trait is acquired by one system can be used by other systems for various purposes, such as access control, identification, or authentication. Inter-operability is a crucial aspect of open biometric systems as it increases the usefulness of biometric traits for various applications without compromising accuracy or security. However, the inter-operability of finger vein images has not been fully explored yet. Though the representation of captured finger vein patterns stays consistent across different acquisition devices [19], image quality and illumination may vary. Some limited studies on cross-device finger vein recognition [19, 20] indicate a significant performance drop in cross-device settings, yet they lack a thorough analysis of the inter-operability of finger vein recognition across different recognition methods.

Inter-operability requires the establishment and adaptation of standard data formats, sample quality control, protocols, and interfaces. Though such standards already exist for

fingerprints, iris, and faces, there is no such standard currently defined for finger vein image quality. This study underscores the importance of establishing such standards for finger vein recognition to achieve inter-operability. Moreover, it analyses the strengths and weaknesses of the finger vein recognition methods across different acquisition settings and proposes a new patch-based CAE architecture that shows great potential under harsh conditions such as cross-device comparisons.

The structure of this paper is as follows: Section 2 provides a comprehensive review of the research on finger vein recognition from its inception to recent advances. The methodology used in this study is outlined in detail in Section 3. Section 4 describes the cross-device dataset used in the experiments, while Section 5 elaborates on the details of the experiments conducted. The results of the experiments are presented in Section 6, and a thorough analysis and discussion of the results are given in Section 7. Finally, in Section 8, the paper concludes with a summary of the findings and suggestions for future research.

## 2. Background and Related Work

*2.1. Background on Finger Vein Recognition.* Classical methods for finger vein recognition widely exploit the characteristics of finger veins, such as shape and texture or key points of the vein patterns, e.g., bifurcations and end-points. The methods employing shape features utilise the curvature information from image cross-sections, where the veins are observed as dents. Miura et al. [1] propose a method that iteratively tracks these dents over the finger vein image. Though the method is successful at extracting vein patterns, the approach has a potential to track noise over the image. Later the same authors employed the maximum curvature method to detect the vein pixels by utilising the local maxima point of curvatures [2]. The maximum curvature method achieves a reliable vein extraction performance under various illumination conditions, making it a commonly used baseline method in the literature. Huang et al. [21] propose a wide line detector to detect vein lines on a finger vein image. The authors also introduce a pattern normalisation method in the same work, which corrects the translations on vein patterns. The proposed approach achieves an improved recognition performance on an in-house dataset compared to [1] and [2]. Yang et al. [22] incorporate vein anatomy structure into the computation of curvatures. The authors show that the introduction of orientation maps based on curvature direction results in less noisy extraction of vein patterns compared to previously mentioned methods. Additionally, they propose a comparison method to compensate for translational errors between finger vein pairs. The proposed vein extraction and comparison method outperforms widely used vein pattern-based methods on two publicly available finger vein datasets. Qin et al. [23] employ radon transform to enhance finger vein images prior to vein extraction. The proposed enhancement step achieves an improved recognition performance on public datasets, indicating the importance of pre-processing for shape-based features. While shape features focus solely on the vein pattern, texture

features encompass both vein patterns and the finger background information. Rosdi et al. [4] utilise local line binary patterns (LLBP) to extract texture information from finger vein images. Different from the local binary pattern (LBP), LLBP takes into account the horizontal shape of veins. The proposed texture descriptor outperforms LBP and local derivative patterns on an in-house dataset. Hu et al. [5] combine LBP descriptors with 2D principal component analysis to minimise the redundant information in LBP images. Though the finger background contains valuable information for finger vein recognition, it is more susceptible to illumination changes compared to vein patterns. Consequently, texture-based features are generally regarded as less robust to illumination changes than shape features. Besides the shape and texture, key points of vein patterns, such as bifurcations and endpoints, are utilised as feature descriptors for finger vein recognition. Though key-points are considered more robust against rotations, irregular shading and low contrast of finger vein images affect the accuracy of extracted features. In order to mitigate irregular shading, Matsuda et al. [6] utilise curvature vein templates instead of the grey images for key-point extraction. Liu et al. [7] employ singular value decomposition for key-point pairing to achieve robustness against rotations and translations.

Over the last decade, deep learning models, particularly the CNNs, have gained popularity for finger vein recognition. Researchers propose a variety of architectures to address challenges in finger vein recognition, such as intra-class variation. The majority of these models used for finger vein recognition typically require a 3-channel input. Zeng et al. [24] propose to combine a pair of finger vein images with their corresponding difference image in these channels, to reinforce the learning of intra-class variations. In contrast, Song et al. [25] opted for a different strategy, utilising a composite image of finger image pairs instead of a different image. Tang et al. [11] propose a Siamese CNN along with contrastive loss to effectively learn both inter- and intra-class variances. Wang et al. [26] investigate multi-scale features using Inception modules in order to achieve robust finger vein representations against translations. Additionally, the authors incorporate centre loss to further enhance the discriminative properties learned by the proposed architecture. The study conducted by Kuzu et al. [12] on various state-of-the-art architectures highlights the superiority of pre-trained weights over random initialisation on public finger and palm vein datasets. A subsequent study by the same authors [13] emphasises the importance of the loss function on the recognition performance. CNNs are also employed as a complementary tool to enhance the classical finger vein recognition methods. Prommeger et al. [27] evaluate several CNN architectures for finger segmentation and analyse various training strategies, including combining all datasets for training the architectures and excluding the evaluation set from this combination. The study reveals that when the evaluation set is not represented in the train data, all the subjected networks fail to achieve compatible segmentation results.

Although CNNs are now more popular in finger vein recognition, the literature presents diversity in deep learning architectures on finger vein recognition. Ou et al. [28]

propose a generative adversarial network architecture to artificially increase finger vein samples for training deep learning models. The architecture specifically focuses on increasing intra-class variance while generating new finger vein samples. The considerable improvement in recognition performance achieved through the use of synthetic data underscores the significance of training data size in deep learning models. Huang et al. [29] employ a vision transformer (ViT)-based architecture to combine local and global finger vein features. The authors employ an extreme learning machine (ELM) for multi-level feature extraction. The proposed ViT+ELM approach outperforms base ViT architectures on finger vein recognition. Apart from supervised methods, researchers also explore unsupervised approaches for finger vein recognition. Bros et al. [18] utilise a CAE architecture for enhancing finger vein images, where the enhanced version is obtained as a linear combination of the raw finger vein image and its corresponding manually annotated vein patterns. The approach achieves a 20% improvement in recognition performance on a publicly available dataset. Similarly, Chen et al. [30] employ a CAE architecture for finger vein segmentation. Later, a descriptor network is trained to learn key point information from the segmented finger vein images. This approach achieved a 5% improvement in recognition performance compared to baseline vein extraction and comparison methods. Pan et al. [31] propose a two-branch CAE architecture for finger vein enhancement. The authors aim to disentangle texture and shape features by using the two-branch architecture. The proposed approach outperforms the majority of the classical methods examined in this work, indicating the potential of unsupervised methods in finger vein recognition.

*2.2. Inter-Operability.* Inter-operability is essential for the efficient and accurate exchange and utilisation of biometric data across multiple systems. To facilitate the exchange of biometric data across systems standards, protocols, and interfaces as well as data format, such as ANSI/NIST-ITL [32], ISO/IEC 19784 [33], or ISO/IEC 39794-9 [34] are established. These standards and protocols define conventions for data capture, including aspects such as resolution, positioning, or the dimensions of the captured biometric trait. While these standards are well-established and widely used in fingerprint or iris recognition, there is currently no such standard specifically implemented for finger vein recognition.

Despite its importance, there is little work exploring the inter-operability of finger vein images. Kauba et al. [35] make one of the few attempts in this regard. The authors introduce two finger vein acquisition devices equipped with LED and laser illumination modules. Though the work examines the impact of illumination modules on finger vein recognition performance, it presents cross-device comparisons as well. Though the presented cross-device performance is at an acceptable level, the authors indicate a substantial drop in recognition performance compared to the single-device setting. Moreover, since both devices in this study have the same design except for the illumination modules, the results provide limited insight into the factors affecting cross-device finger vein recognition. Prommeger et al. [20] investigate

cross-device finger vein recognition on four acquisition devices. They evaluate the performance using classical baseline methods and a CNN architecture. The results indicate a significant performance drop in cross-device recognition, particularly when the acquisition devices exhibit different characteristics, such as contact features. Though this work includes more variety in devices and recognition methods, it is limited in terms of exploring the reasons causing this performance drop with cross-device pairs. Arican et al. [19] perform a cross-device recognition experiment on five different finger vein acquisition devices. The work implements only one classical finger vein recognition method. The results of the cross-device experiments support the previous works by showing a substantial performance drop compared to single-device experiments. The authors attribute this poor performance in the cross-device setting to the deformations on the captured vein patterns due to the different properties of the acquisition devices. Though this study provides a deeper understanding of inter-operable finger vein recognition compared to the other works, it employs only a single recognition method for analysis. Due to challenges in the data gathering process, Arican et al. [19] utilises only a limited part of the cross-device data introduced in their study. This study extends the dataset presented by Arican et al. [19] by utilising all available data captured by six acquisition devices. Furthermore, in addition to the classical method employed by Arican et al. [19], this work incorporates two more recent deep-learning methods for cross-device finger vein recognition. As a result, this study provides a more comprehensive understanding of cross-device finger vein recognition. Additionally, this study introduces a patch-based CAE architecture in conjunction with the expanded dataset and the recognition methods.

### 3. Methodology

**3.1. Miura Method.** Miura method is a well-known classical approach used for finger vein recognition. It utilises the maximum curvature method [2] for extracting vein patterns and the Miura match technique [1] for comparing the vein patterns. The maximum curvature method provides a reliable vein extraction under challenging illumination conditions, and the Miura match improves the robustness against translation errors that are frequently observed in finger vein images. As a result, the Miura method is widely used as a baseline method for finger vein recognition. Hence, in this study, the Miura method is utilised to establish a reference point for cross-device recognition experiments.

**3.1.1. Maximum Curvature.** In an infrared image, finger veins appear as dark ghost-like lines, and their cross-section is represented as dents. The maximum curvature method calculates the local maxima of these dents which locates a vein pixel (Figure 1). The depth and the width of the dents are utilised to determine the likelihood of a detected point being on a vein. Then, the likelihoods are compared against a threshold to achieve the binary vein pattern. Since the method only utilises the location of the local maxima, maximum curvature cannot provide width information about the

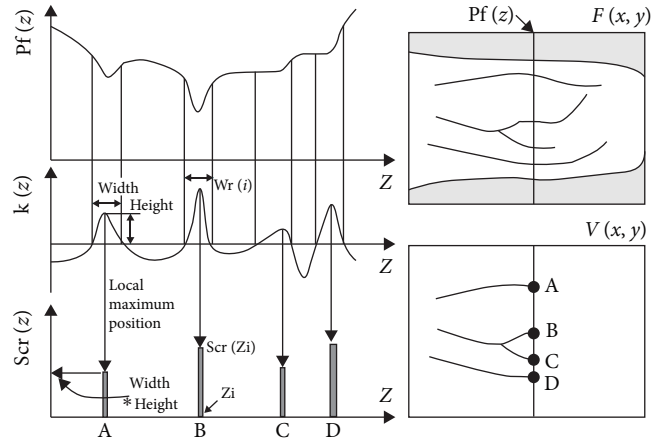


FIGURE 1: Cross-sectional profile and curvature points.

veins. As a result, the extracted vein pattern has a constant width for all detected veins. The curvature behaviour is robust against illumination, allowing the maximum curvature method to extract reliable vein patterns under challenging illumination conditions.

**3.1.2. Miura Match.** Miura match is proposed by Miura et al. [1] is a method used to compute the correlation between two binary finger vein patterns. The method calculates the correlation scores as the ratio of the total number of vein pixels to the number of correlated vein points, resulting in scores ranging from 0.0 (no correlation) to 0.5 (perfect correlation). The Miura match method provides robustness against translation errors by cropping a small window of the probe image, which allows the method to compensate for small translations between finger vein pairs. Figure 2 shows how the method is executed. Reducing the crop size allows compensation of larger translation errors but also increases the likelihood of false matches.

**3.2. CNNs.** CNNs are a type of neural network capable of capturing sophisticated spatial and temporal relations among image pixels by using a series of convolutional kernels. These kernels are trained with the input data, allowing CNNs to generalise finger vein features compared to classical methods. The learned representations through layers of kernels can then be used to make predictions. Owing to these properties, CNNs have led to breakthroughs in computer vision tasks such as biometric recognition and medical imaging.

The CNN architecture used in this work is proposed by Kuzu et al. [13] and achieved promising recognition performances on publicly available finger and palm vein datasets. The authors have modified the DenseNet-161 architecture by adding a “Custom Embedding Layer” (Table 1) to extract finger vein features. This custom layer encapsulates an average pooling layer, a fully connected layer, and batch normalisation. The custom embedding layer is trained from scratch to learn dedicated finger vein features, while the rest of the layers are initialised with ImageNet [36] weights. The network is trained on a publicly available finger vein dataset (SDUMLA-HMT [37]) in identification mode, where every

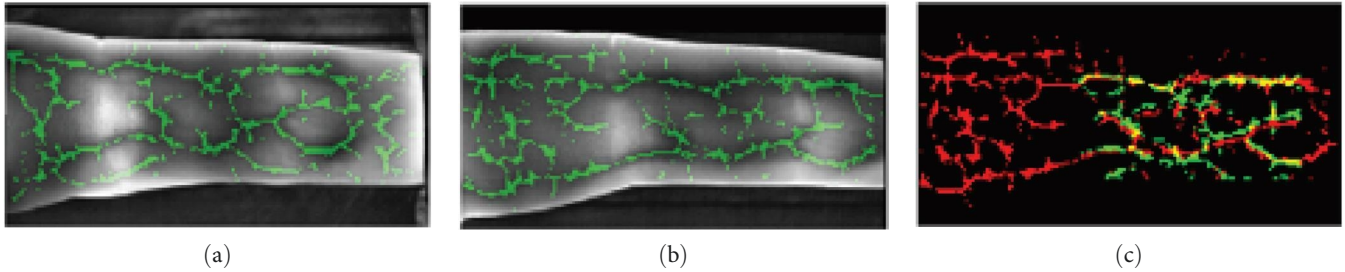


FIGURE 2: Maximum curvature veins and Miura match: (a) reference image; (b) probe image; (c) comparison.

TABLE 1: CNN architecture.

| Layers          |   | Input size                 | Output size                |
|-----------------|---|----------------------------|----------------------------|
| Convolution     | $7 \times 7$ conv, str.2                            | $224 \times 224 \times 3$  | $112 \times 112 \times 96$ |
| Pooling         | $3 \times 3$ max pool, str.2                        | $112 \times 112 \times 96$ | $56 \times 56 \times 384$  |
| Dense block 1   | $1 \times 1$ conv<br>$3 \times 3$ conv $\times 6$   | $56 \times 56 \times 96$   | $56 \times 56 \times 384$  |
| Transition 1    | $1 \times 1$ conv<br>$2 \times 2$ avr. pool., str.2 | $56 \times 56 \times 384$  | $28 \times 28 \times 192$  |
| Dense block 2   | $1 \times 1$ conv<br>$3 \times 3$ conv $\times 12$  | $28 \times 28 \times 192$  | $28 \times 28 \times 768$  |
| Transition 2    | $1 \times 1$ conv<br>$2 \times 2$ avr. pool., str.2 | $28 \times 28 \times 768$  | $14 \times 14 \times 384$  |
| Dense block 3   | $1 \times 1$ conv<br>$3 \times 3$ conv $\times 36$  | $14 \times 14 \times 384$  | $14 \times 14 \times 2112$ |
| Transition 3    | $1 \times 1$ conv<br>$2 \times 2$ avr. pool., str.2 | $14 \times 14 \times 2112$ | $7 \times 7 \times 1056$   |
| Dense block 4   | $1 \times 1$ conv<br>$3 \times 3$ conv $\times 24$  | $7 \times 7 \times 1056$   | $7 \times 7 \times 2208$   |
| Custom embedder | $7 \times 7$ global avr. pool                       | $7 \times 7 \times 2208$   | $1 \times 2208$            |
|                 | Batch normalisation                                 |                            |                            |
|                 | Dropout (50%)                                       |                            |                            |
|                 | Fully connected layer                               |                            |                            |
|                 | Batch normalisation                                 | $1 \times 2208$            | $1 \times 1024$            |
| Classifier      | Output layer  | $1 \times 2024$            | $1 \times U$               |

finger represents a different identity. Evaluation is conducted in verification mode, where the network outputs learned representations of fingers, and the Euclidean distance is used to measure the similarity between two finger vein representations. SDUMLA-HMT dataset exhibits rotation and translation errors, therefore, training the model on such data could come in handy while dealing with transformations observed on cross-device finger vein pairs.

**3.3. Patch-Based CAE (P-CAE).** An auto-encoder is a neural network that consists of two sub-networks, namely, an encoder and a decoder. The encoder network compresses the input to a lower dimensionality while the decoder decompresses it to the input dimensions again. Through this process, the AE aims to learn the most representative features of the input data, which can later be used for comparison. Training focuses on achieving the closest

decompression output to the input data, allowing AEs to be trained in an unsupervised fashion. This property is useful when the label information is difficult to realise for large amounts of data, such as finger veins. Convolutional variants of AEs (CAEs) involve “convolution layers” rather than traditional fully connected layers. These convolutional layers help to discover abstract relations among image pixels. CAEs can be a promising alternative to complex and deep CNN models used in finger vein recognition since the simplicity of the architecture can allow one to learn pure finger vein features without compromising the generalisation of the extracted features.

Despite the CAEs theoretically holding promise in learning finger vein representations, few studies [38, 39] point out significant challenges in this direction. Figure 3 illustrates the hurdles faced in reconstructing vein patterns using a CAE model proposed in [38]. The authors state that due to the low

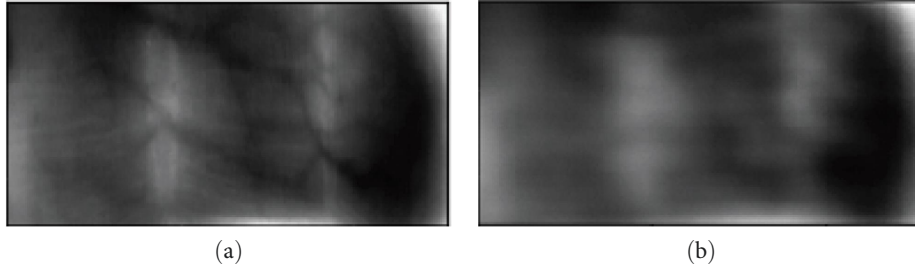


FIGURE 3: Finger vein image reconstruction with a CAE [38]: (a) input; (b) reconstruction.

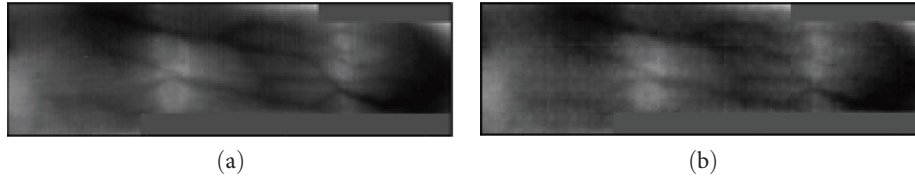


FIGURE 4: Finger vein image reconstruction with the patch-based CAE: (a) input; (b) reconstruction.

TABLE 2: CAE architecture.

| Blocks                 | Input size<br>( $H \times W \times C$ ) | Output size<br>( $H \times W \times C$ ) |
|------------------------|---|--|
| Convolution block 1    | $65 \times 65 \times 1$                 | $32 \times 32 \times 16$                 |
| Convolution block 2    | $32 \times 32 \times 16$                | $16 \times 16 \times 32$                 |
| Convolution block 3    | $16 \times 16 \times 32$                | $8 \times 8 \times 64$                   |
| Convolution block 4    | $8 \times 8 \times 64$                  | $4 \times 4 \times 128$                  |
| Convolution block 5    | $4 \times 4 \times 128$                 | $2 \times 2 \times 256$                  |
| Convolution block 6    | $2 \times 2 \times 256$                 | $1 \times 1 \times 512$                  |
| Latent vector          | $1 \times 1 \times 512$                 | $1 \times 32$                            |
| De-convolution block 1 | $1 \times 1 \times 512$                 | $3 \times 3 \times 256$                  |
| De-convolution block 2 | $3 \times 3 \times 256$                 | $5 \times 5 \times 128$                  |
| De-convolution block 3 | $5 \times 5 \times 128$                 | $9 \times 9 \times 64$                   |
| De-convolution block 4 | $9 \times 9 \times 64$                  | $17 \times 17 \times 32$                 |
| De-convolution block 5 | $17 \times 17 \times 32$                | $33 \times 33 \times 16$                 |
| De-convolution block 6 | $33 \times 33 \times 16$                | $65 \times 65 \times 16$                 |
| Convolution            | $65 \times 65 \times 16$                | $65 \times 65 \times 1$                  |

contrast and sparsity of the vein patterns, the CAE struggles to learn vein pattern encoding, yet it rather learns global structures such as finger background and joint shapes and locations.

This study introduces a patch-based approach to learning vein representations to address the challenges encountered in prior studies. The P-CAE is designed to reconstruct small patches extracted from the finger region instead of reconstructing the entire finger region in one step. By focusing on patches, the P-CAE aims to simplify the task of reconstructing sparse and complicated vein patterns to learn the representations of small vein segments. Figure 4 illustrates the improved reconstruction performance of the proposed P-CAE on the same finger compared to the CAE proposed in [38] (Figure 3).

TABLE 3: Content of a convolution and de-convolution block.

| Block type           | Block content                         |
|----------------------|---------------------------------------|
| Convolution block    | $3 \times 3$ conv., str. 2            |
|                      | Batch normalisation                   |
|                      | LeakyReLU                             |
| De-convolution block | $3 \times 3$ transposed conv., str. 2 |
|                      | Batch normalisation                   |
|                      | LeakyReLU                             |

The P-CAE model proposed in this study, presented in Table 2, is trained to learn finger vein features from scratch. The architecture consists of six layers of encoder and decoder networks. The encoder performs compression through convolution blocks, while the decoder utilises de-convolution blocks for the reconstruction of the input data (Table 3).  $65 \times 65$  pixel patches are compressed to a latent vector with a dimension of 32. The patch size is determined after a series of experiments. The patches are extracted only from the finger region. The P-CAE is trained on the UTFVP dataset [40], which offers high-quality finger vein images that could help in learning more representative finger vein features. Mean absolute error is utilised to measure how close the reconstructed image is to the input. The model is trained with Adam optimiser with a learning rate of  $10^{-5}$  and a weight decay of  $10^{-8}$  for 50 epochs.

Comparison of image pairs is conducted at patch level. Initially, overlapping patch pairs of size  $65 \times 65$  pixels are extracted from corresponding positions in both the reference and probe images. Subsequent to patch extraction, the latent representations of these patch pairs are computed through the encoder of the P-CAE. The similarity between each patch pair is quantified using cosine similarity. The overall similarity of the image pair is determined by averaging all patch pair similarities.

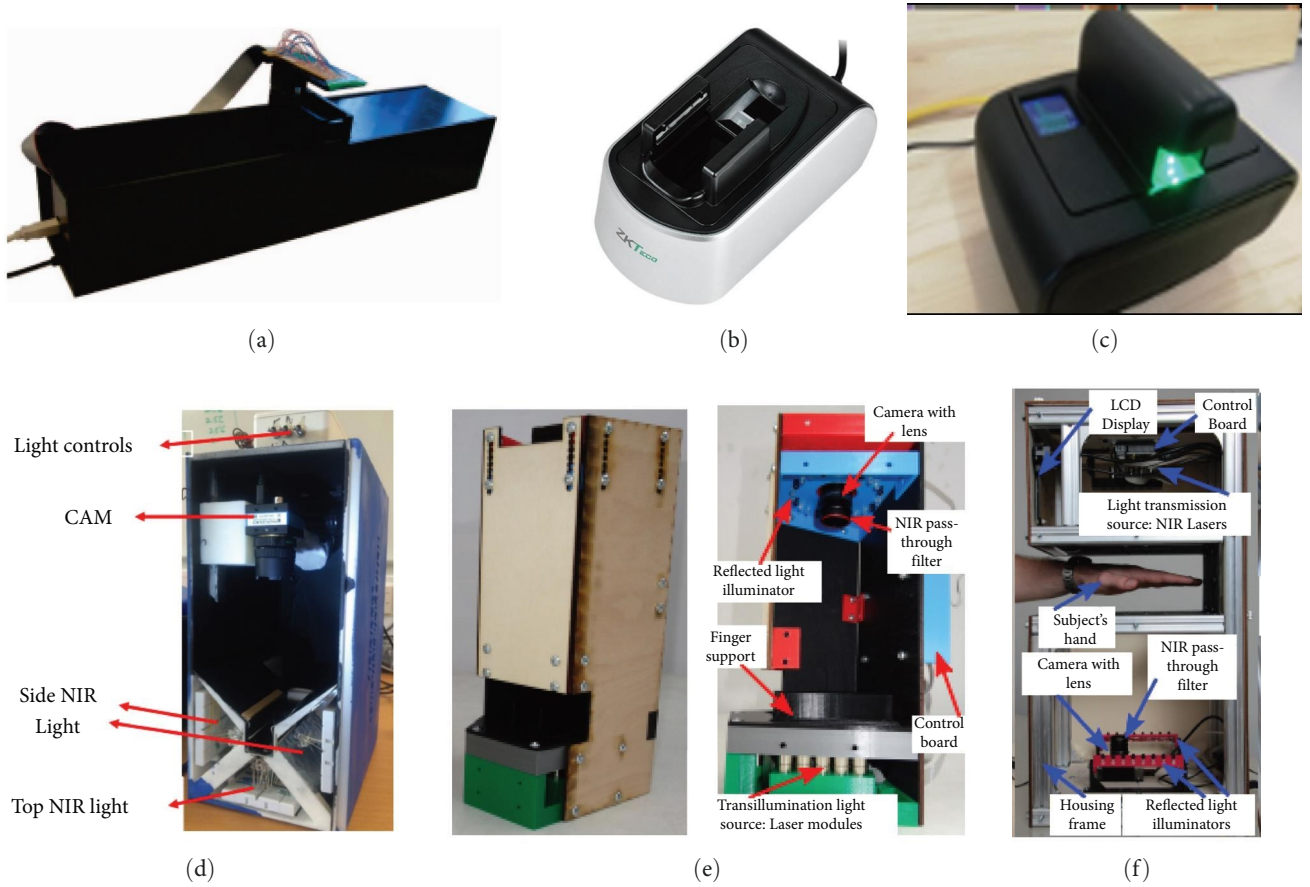


FIGURE 5: Acquisition devices: (a) UTFV device; (b) ZkTeco device; (c) IDIAP device; (d) NTNU device; (e) PFV\_L device; (f) PFV\_C device.

TABLE 4: Device properties.

| Device | Subjects per session | Number of images | Resolution       | Illumination type | Illumination side |
|--------|----------------------|------------------|------------------|-------------------|-------------------|
| UTFV   | 58/45                | 618              | $340 \times 648$ | LED               | Top               |
| ZkTeco | 58/46                | 624              | $240 \times 320$ | LED               | Two-side          |
| IDIAP  | 55/41                | 567              | $320 \times 240$ | LED               | Top               |
| NTNU   | 58/44                | 612              | $132 \times 200$ | LED               | Top               |
| PFV_L  | 41/9                 | 294              | $180 \times 500$ | Laser             | Bottom            |
| PFV_C  | 48/9                 | 342              | $182 \times 505$ | Laser             | Top               |

## 4. Devices and Dataset

The cross-device data [19] is a collaboration involving the University of Twente, Salzburg University, the Norwegian University of Science and Technology, and the IDIAP Research Institute. The dataset was compiled in 2019 during the BIOSIG conference in Darmstadt, Germany. Over a span of 2 days, finger vein images of 59 individuals were recorded using six distinct capturing devices. Figure 5 shows the devices used in data acquisition. Images of six fingers (index, middle, and ring fingers of both hands) are captured by each device in two sessions. Table 4 provides information about the number of subjects involved in each device as well as device properties. The finger vein images are acquired under limited supervision in order to simulate a more realistic recognition scenario. The cross-device finger vein

dataset is publicly available (<https://www.utwente.nl/en/eemcs/dmb/downloads/utcdfvp/>).

**4.1. UTFV.** The device developed by the University of Twente [40] features a semi-open design that allows user to see the position of their finger. The device includes a finger rest, which restricts finger positioning. Illumination is provided through 8 850 nm LED-infrared modules from the top, while the camera is located at the bottom. The illumination modules are automatically adjusted. The device captures high-quality finger vein images at a resolution of  $340 \times 648$  pixels (Figure 6(a)).

**4.2. ZkTeco.** ZkTeco [41] is a commercial device that comes with both finger vein and fingerprint sensors. The device has

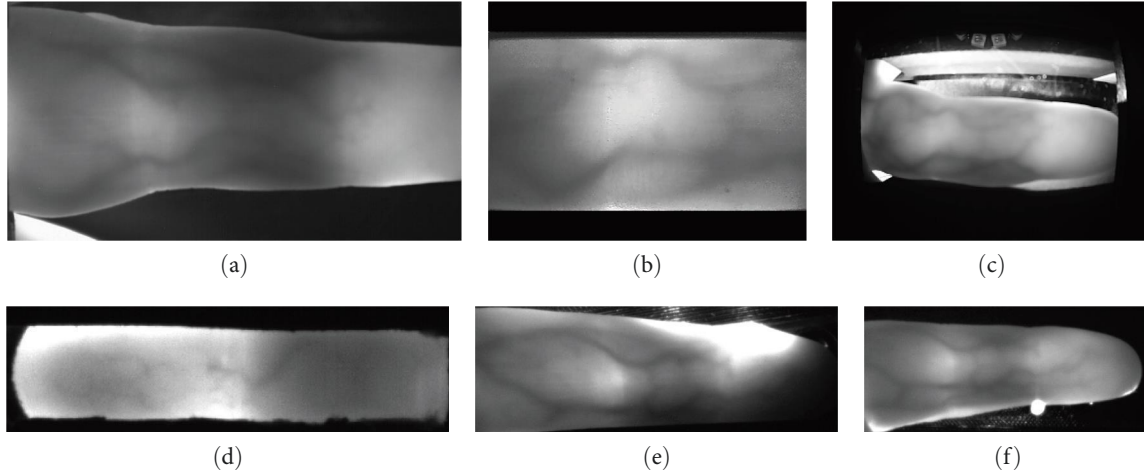


FIGURE 6: Finger vein images acquired by different devices: (a) UTFV image; (b) ZkTeco image; (c) IDIAP image; (d) NTNU image; (e) PFV\_L image; (f) PFV\_C image.

a fully open design and features a finger rest. LED-infrared illumination modules are located on both sides of the finger. Unlike other devices, ZkTeco captures only a small region of the finger around the first knuckle. The captured images have a resolution of  $240 \times 320$  pixels (Figure 6(b)).

**4.3. IDIAP.** The device provided by the IDIAP research institute [42] has a fully closed design and does not feature finger support, allowing the user more flexibility in finger placement. As a result, the images exhibit more pronounced geometric transformations, such as 3D rotations and/or bending, when compared to the previous two devices. In addition to geometric transformations, this device also exhibits slight radial distortion (Figure 6(c)). The radial distortion is corrected by using the approach presented in [43]. The illumination is provided from the top side, and the captured images have a resolution of  $320 \times 240$  pixels.

**4.4. NTNU.** The acquisition device developed by the Norwegian University of Science and Technology [44] captures finger vein images with a resolution of  $744 \times 480$  pixels. However, a significant portion of the captured images consist of a black background. After removing the background, the images are resized to  $132 \times 342$  pixels. This part is removed in this work. After removing this black background, images are resized to  $132 \times 342$  pixels (Figure 6(d)). The device is equipped with LED-NIR modules positioned at the top of the finger. A finger rest guides the user with finger positioning and restricts finger movements during image acquisition.

**4.5. Plus\_FV3\_Laser (PFV\_L).** The device developed by Salzburg University [35] is equipped with infrared-laser illumination modules and features a finger support that restricts the placement of the fingers on the device. This device captures three fingers at once, which are then separated in a pre-processing stage. Due to the design of the finger support, some images may include a part of the neighbouring fingers. After pre-processing, the image resolution is set at  $300 \times 450$  pixels (Figure 6(e)). This device has only nine subjects with two sessions.

**4.6. PLUS\_FV3\_Contactless (PFV-C).** The device developed by Salzburg University [45] enables fully contactless image acquisition. As PFV\_L device, this device is featured with Laser LED modules. Though being a contactless device, it is equipped with a touchscreen that guides the user with finger positioning. The resolution of the captured images is set at  $182 \times 505$  pixels (Figure 6(e)). The data collected by this device have only nine subjects with two sessions.

## 5. Experiments

Finger vein recognition performance is evaluated on each device using the Miura method, CNN, and our proposed P-CAE architecture mentioned in Section 3. Single-device pair experiments serve as a reference point for cross-device pairs and also examine the factors influencing finger recognition. PFV\_L and PFV\_C devices have only nine subjects with two sessions. Single-device pair comparisons on these devices include the pairs generated among only these nine subjects, while the cross-device pairs involve all common subjects among device pairs.

The Miura method parameters are optimised for each single-device pair, and the reference device parameters are used in cross-device evaluations to simulate a more realistic cross-device recognition scenario. The CNN model is pre-trained on ImageNet [36] and then fine-tuned with the SDUMLA-HMT [37] finger vein dataset. This data exhibits severe rotations and translations. Since it is indicated in [19], the rotation and translation errors are one of the main reasons for bad inter-operable performance, the CNN can benefit from the geometric transformations observed in SDUMLA-HMT. On the other hand, our P-CAE model is trained on the UTFVP dataset from scratch. This dataset involves high-quality finger vein images, allowing the P-CAE to learn more representative finger vein features. There is no further tuning performed on both CNN and P-CAE models.

Recognition performances are compared using an equal error rate (EER) in percentage and false non-match rate where the false match rate equals 0.1% (FMR1000). EER



TABLE 5: Number of mated and non-mated pairs per device pair.

|        | UTFV  |           | ZkTeco |           | IDIAP |           | NTNU  |           | PFV_L |           | PFV_C |           |
|--------|-------|-----------|--------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|-----------|
|        | Mated | Non-mated | Mated  | Non-mated | Mated | Non-mated | Mated | Non-mated | Mated | Non-mated | Mated | Non-mated |
| UTFV   | 528   | 10,000    | 1,056  | 10,000    | 888   | 10,000    | 1,032 | 10,000    | 528   | 10,000    | 600   | 10,000    |
| ZkTeco | 1,056 | 10,000    | 540    | 10,000    | 912   | 10,000    | 1,056 | 10,000    | 540   | 10,000    | 612   | 10,000    |
| IDIAP  | 888   | 10,000    | 912    | 10,000    | 456   | 10,000    | 888   | 10,000    | 432   | 10,000    | 480   | 10,000    |
| NTNU   | 1,032 | 10,000    | 1,056  | 10,000    | 888   | 10,000    | 528   | 10,000    | 534   | 10,000    | 606   | 10,000    |
| PFV_L  | 528   | 10,000    | 540    | 10,000    | 432   | 10,000    | 534   | 10,000    | 84    | 6,888     | 366   | 10,000    |
| PFV_C  | 600   | 10,000    | 612    | 10,000    | 480   | 10,000    | 606   | 10,000    | 366   | 10,000    | 84    | 6,888     |

The columns of the table represent the probe devices, while the rows are the reference.

indicates the error rate where the false non-match rate equals to false match rate, while FMR1000 presents the false non-match rate where the false match rate equals 0.1% at most. Additionally, similarity score histograms are utilised to interpret how the recognition method is affected by different device pairs.

*5.1. Pre-Processing.* The images acquired by each device have different properties, such as the image background and the number of fingers captured in a frame, which require adjustments to the pre-processing step accordingly. For example, the PFV\_L device captures three fingers per frame, while the PFV\_C device involves a single finger. On these devices, fingers are detected using the finger tip and finger edges; then, the fingers are cropped using these reference points. Table 4 presents the image resolutions after cropping. The images acquired by the IDIAP device are subject to radial distortion. Yet, since the distortion parameters are unknown, the images from this device are corrected using the fact that straight lines transform into curves under radial distortion [43]. Contrast-limited adaptive histogram equalisation is applied to improve the contrast of the finger vein images. In-plane finger rotations are corrected using the center line method proposed by Huang et al. [21]. First, the finger edges are detected using the edge detection method proposed by Lee et al. [46]. Due to the complex background on the IDIAP device, the Sobel edge detector [47] is used. Then, a straight line is fitted in between the edges, and the line is aligned to the centre of the image.

*5.2. Image Pair Generation, Alignment, and Comparison.* A mated pair refers to a pair of biometric samples from the same individual and from the same biometric instance (i.e., the same finger), and similarly, a non-mated pair is a pair of images coming from different individuals. For single-device datasets, for each image of a finger, there is one mated sample, since only two images are captured per finger. For the cross-device pairs, for each finger image, there are two mated pairs since both images captured on the probe device could be considered as a new image of the finger. The number of mated and nonmated image pairs generated for each device setting is presented in Table 5.

While comparing single-device pairs is straightforward, the challenge arises when dealing with cross-device image pairs due to variations in the resolutions of the reference and probe devices. To ensure equivalent conditions in

cross-device comparisons, it is crucial to calibrate the devices beforehand, aligning the structures at the same scale. In the absence of calibration parameters in this dataset, manual scaling between reference and probe image pairs is required for each device setting. To establish these scaling parameters, maximum curvature veins are employed. Initially, a subset of mated pairs is randomly selected for each device setting, and maximum curvature veins are extracted. Horizontal and vertical scaling parameters are then determined with the objective of achieving the maximum correlation between the reference and probe veins. It is crucial to note that the lower-resolution image is consistently scaled to match the higher-resolution image.

Normakristagalu et al. [48] emphasise the importance of the proper alignment of finger vein image pairs for effective comparisons. In line with their insight, the iterative closest point (ICP) [49] method is employed to align reference and probe finger vein images. Figure 7 illustrates the ICP registration process. The ICP uses finger-edge information to correct in-plane rotations. However, it falls short in addressing other registration errors, such as translations, where the captured vein pattern is shifted in the probe image. Van der Spek et al. [50] argue that the finger bone structure carries identity information, indicating that this structure should be consistent for mated pairs and can be used to detect and correct translation errors, particularly in the context of horizontal shifts. The method proposed in [51] is employed to compensate for horizontal translations between reference and probe images. Figure 6 demonstrates that ZkTeco captures only a region around the first joint while the whole finger is captured by the other devices. Previous work [19] points out this difference as one of the reasons for false rejections in the UTFV-ZkTeco device pair. To align the ZkTeco device images with those of other devices, the same horizontal alignment method is utilised. Figure 8 illustrates the horizontal alignment process on an image pair from UTFV-ZkTeco.

Miura method possesses distinct parameters for vein extraction and translation compensation. Similarly, the comparisons with the P-CAE employ specific parameters for translation compensation. These parameters are device-specific and finely tuned to optimise comparisons within individual devices. However, in the context of cross-device comparisons, instead of the conventional approach of optimising these parameters for each device pair, the parameters optimised for the reference device are used.

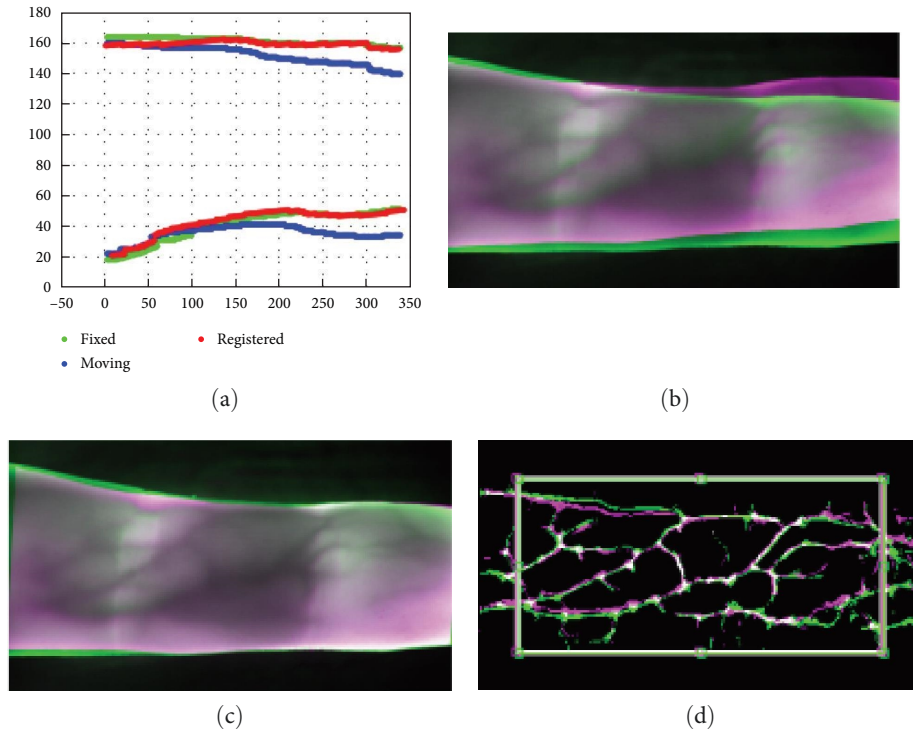


FIGURE 7: ICP registration [48]: (a) contour points in  $xy$ -axis; (b) overlay fixed moving; (c) overlay fixed registered (ICP); (d) overlay fixed and translated (ICP).

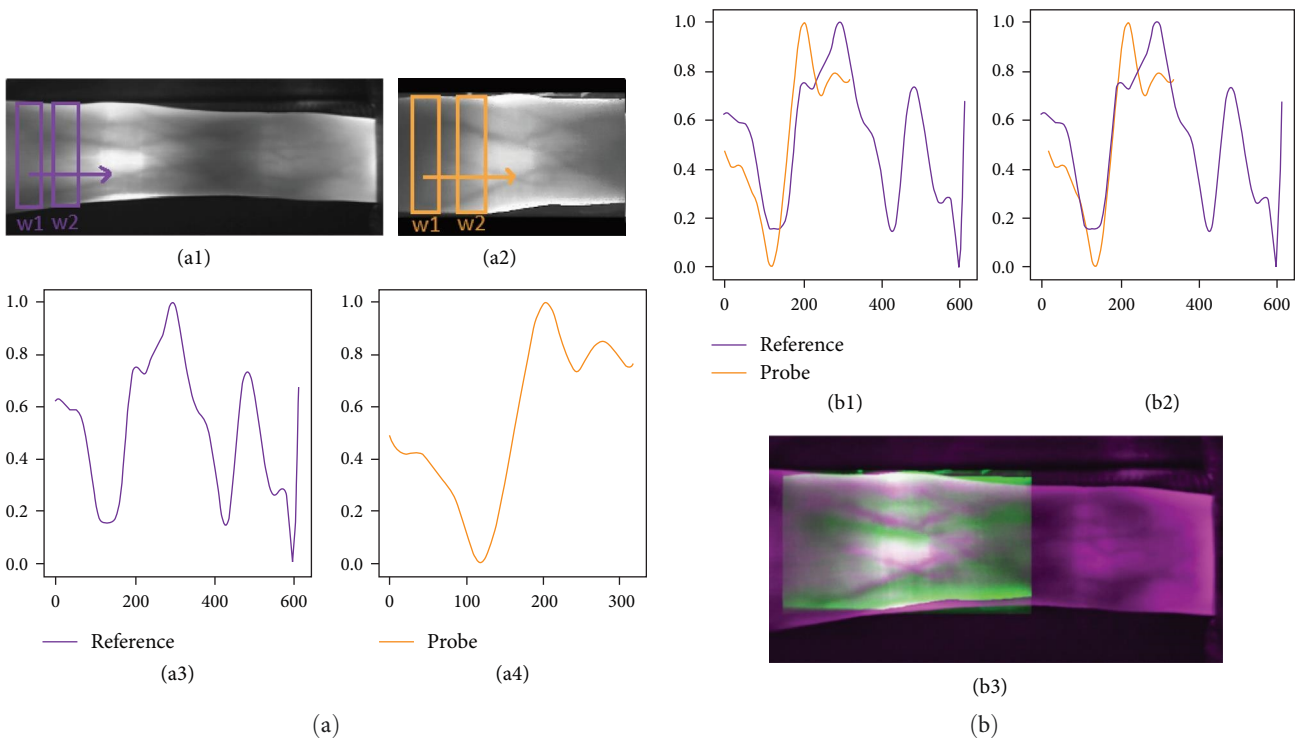


FIGURE 8: Image pair alignment using bone structure: (a) finger bone structure ((a1) reference image, (a2) probe image, (a3) reference grey profile, (a4) probe grey profile); (b) pair alignment ((b1) before alignment, (b2) after alignment, (b3) aligned image pair).

TABLE 6: EER and FNMR1000 performances in the percentage of Miura method, CNN, and CAE with different device pairs.

|        | Miura |         |        |         |       |         |       |         |       |         |      |         |
|--------|-------|---------|--------|---------|-------|---------|-------|---------|-------|---------|------|---------|
|        | UTFV  |         | ZkTeco |         | IDIAP |         | PFV_L |         | PFV_C |         | NTNU |         |
|        | EER   | FMR1000 | EER    | FMR1000 | EER   | FMR1000 | EER   | FMR1000 | EER   | FMR1000 | EER  | FMR1000 |
| UTFV   | 0.57  | 1.52    | 10.4   | 50.19   | 25.0  | 82.67   | 47.1  | 97.54   | 49.8  | 99.83   | 47.7 | 99.13   |
| ZkTeco | 15.1  | 40.91   | 5.13   | 24.63   | 35.0  | 88.16   | 45.4  | 97.78   | 34.4  | 98.04   | 49.4 | 99.83   |
| IDIAP  | 36.4  | 88.44   | 29.1   | 66.56   | 16.2  | 44.74   | 37.6  | 89.12   | 40.3  | 94.58   | 39.9 | 99.12   |
| PFV_L  | 48.7  | 97.54   | 44.4   | 99.81   | 44.7  | 99.92   | 9.52  | 34.52   | 42.3  | 87.98   | 49.8 | 100.0   |
| PFV_C  | 49.8  | 99.83   | 48.5   | 98.37   | 37.4  | 97.92   | 43.1  | 90.16   | 22.6  | 58.33   | 46.8 | 99.67   |
| NTNU   | 49.3  | 99.82   | 49.9   | 99.91   | 48.5  | 100.0   | 46.8  | 100.0   | 49.5  | 100.0   | 32.6 | 76.30   |
|        | CNN   |         |        |         |       |         |       |         |       |         |      |         |
|        | UTFV  |         | ZkTeco |         | IDIAP |         | PFV_L |         | PFV_C |         | NTNU |         |
|        | EER   | FMR1000 | EER    | FMR1000 | EER   | FMR1000 | EER   | FMR1000 | EER   | FMR1000 | EER  | FMR1000 |
| UTFV   | 12.7  | 95.19   | 24.4   | 93.74   | 33.4  | 97.58   | 42.4  | 99.64   | 42.2  | 99.42   | 47.8 | 99.88   |
| ZkTeco | 24.4  | 93.74   | 10.6   | 85.02   | 46.4  | 99.43   | 41.9  | 97.71   | 34.2  | 99.54   | 49.9 | 99.97   |
| IDIAP  | 33.4  | 97.58   | 46.4   | 99.43   | 23.2  | 99.57   | 43.3  | 99.41   | 48.5  | 100.0   | 49.5 | 99.64   |
| PFV_L  | 42.4  | 99.94   | 41.9   | 97.71   | 43.3  | 99.41   | 25.0  | 97.13   | 39.6  | 99.96   | 48.1 | 99.50   |
| PFV_C  | 42.2  | 94.42   | 34.2   | 99.54   | 48.5  | 100.0   | 39.6  | 99.96   | 27.4  | 96.62   | 49.5 | 99.49   |
| NTNU   | 47.8  | 99.88   | 49.9   | 99.97   | 49.5  | 99.64   | 48.1  | 99.50   | 49.5  | 99.49   | 34.3 | 98.93   |
|        | CAE   |         |        |         |       |         |       |         |       |         |      |         |
|        | UTFV  |         | ZkTeco |         | IDIAP |         | PFV_L |         | PFV_C |         | NTNU |         |
|        | EER   | FMR1000 | EER    | FMR1000 | EER   | FMR1000 | EER   | FMR1000 | EER   | FMR1000 | EER  | FMR1000 |
| UTFV   | 1.25  | 3.03    | 7.95   | 18.47   | 12.3  | 55.18   | 34.8  | 97.54   | 40.1  | 98.83   | 48.2 | 99.88   |
| ZkTeco | 9.85  | 17.42   | 2.41   | 12.96   | 21.3  | 75.33   | 28.9  | 96.67   | 25.8  | 88.24   | 49.4 | 100.0   |
| IDIAP  | 10.3  | 49.66   | 22.3   | 75.00   | 14.5  | 49.78   | 37.0  | 97.45   | 47.6  | 94.72   | 49.1 | 99.56   |
| PFV_L  | 26.8  | 95.45   | 39.4   | 99.07   | 48.3  | 99.19   | 8.62  | 34.52   | 42.1  | 94.54   | 48.5 | 100.0   |
| PFV_C  | 42.8  | 98.67   | 48.5   | 98.53   | 42.9  | 99.97   | 29.8  | 73.77   | 14.3  | 36.90   | 49.3 | 99.83   |
| NTNU   | 46.9  | 99.91   | 49.9   | 100.0   | 48.2  | 99.89   | 49.4  | 100.0   | 49.1  | 99.50   | 40.7 | 95.64   |

Adhering to reference device parameters in cross-device scenarios aligns coherently with scenarios in which the subjects register on the reference device, while the probe device used for verification remains susceptible to potential replacements due to technical issues or technological advancements.

## 6. Results

**6.1. Single-Device.** The diagonal entries of Table 6 display the recognition performance of the Miura method, CNN, and CAE for each single-device pair. The recognition performance is presented in EER (%) and FNMR1000 metrics. It is observed that the recognition performance varies considerably among acquisition devices, even when the same fingers are captured on each device. For instance, the UTFV device achieves a recognition performance of 0.57% EER with the Miura method, while the performance of the same fingers is 5.13% and 16.2% EER for the ZkTeco and IDIAP devices, respectively. Additionally, the table shows that the NTNU device performs significantly worse compared to the other devices. The best performance achieved on the NTNU device is presented as 32.6% with the Miura method.

**6.2. Cross-Device.** Regardless of the recognition method, cross-device performance is noticeably worse compared to single-device settings. The recognition performance on the

UTFV device drops from 0.57% to 15.1% EER when the reference device is replaced with ZkTeco. Table 6 demonstrates that the change in recognition performance significantly varies among cross-device pairs. For example, the UTFV–ZkTeco pair achieves 10.4% EER with the Miura method, while the recognition performance drops by approximately 55% when the probe device is IDIAP. The histogram plots of single and cross-device pairs (Figures 9–11) reveal that mated cross-device pairs have substantially lower scores than single-device pairs. The plots also demonstrate that in cross-device settings, the subjected recognition methods struggle to distinguish mated pairs from non-mated ones.

Although PFV devices capture high-quality images, their cross-device pairs demonstrate substantially worse performance compared to the other device pairs, such as UTFV–ZkTeco and UTFV–IDIAP. Furthermore, despite the small size of the evaluation set, the PFV\_C device achieves noticeably higher EER compared to the PFV\_L device. Figure 12 also shows that mated pairs of UTFV–PFV\_L and UTFV–PFV\_C are indistinguishable from non-mated pairs.

Although none of the recognition methods achieves competitive performance on cross-device pairs, it is evident that the proposed P-CAE outperforms the classical method on cross-device pairs, including UTFV, ZkTeco, and IDIAP devices. Miura method achieves 10.4% EER on the UTFV–ZkTeco

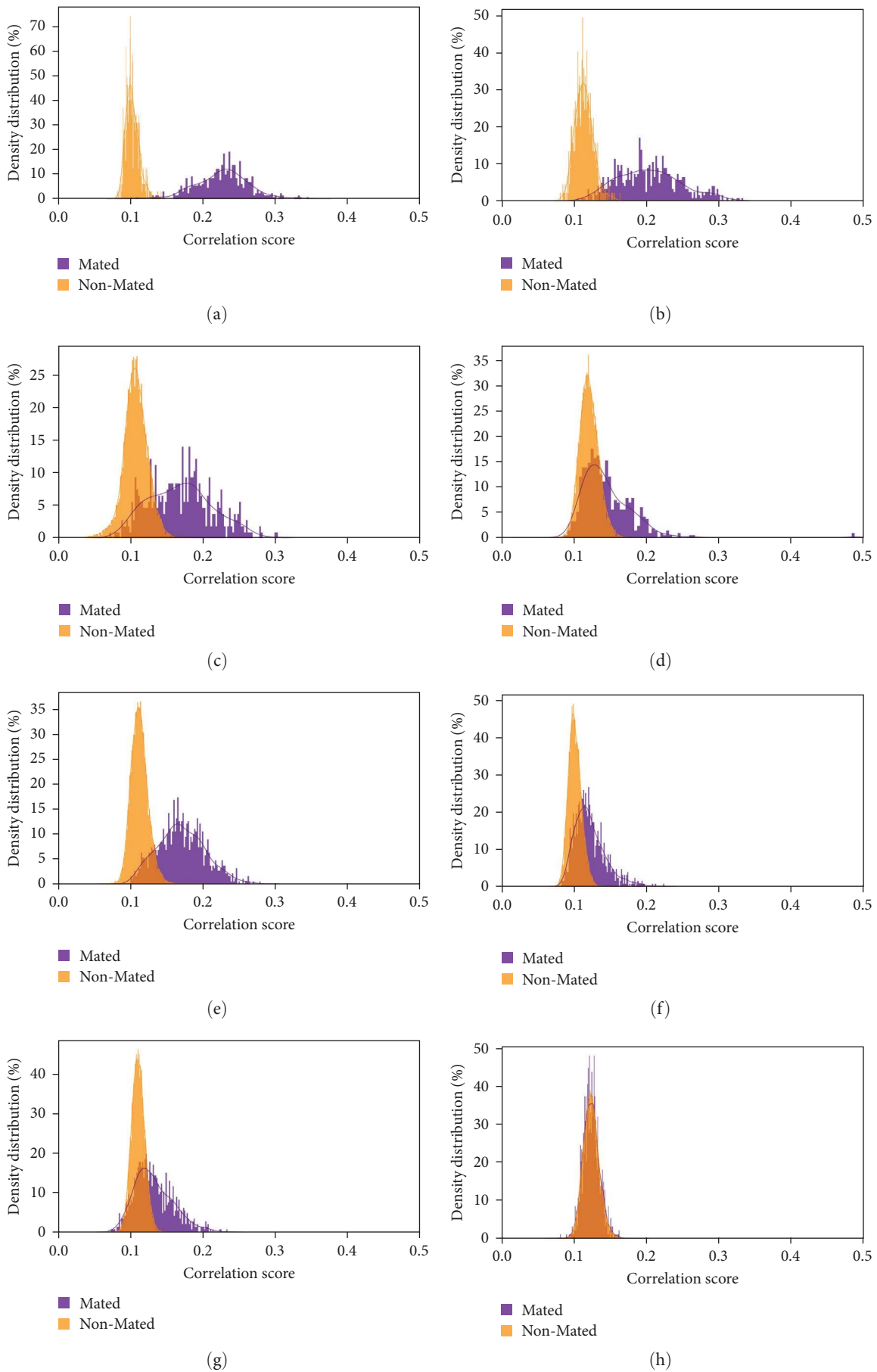


FIGURE 9: Correlation histograms of single-device pairings with Miura method: (a) UTFV-UTFV; (b) ZkTeco-ZkTeco; (c) IDIAP-IDIAP; (d) NTNU-NTNU; (e) UTFV-ZkTeco; (f) UTFV-IDIAP; (g) IDIAP-ZkTeco; (h) NTNU-ZkTeco.

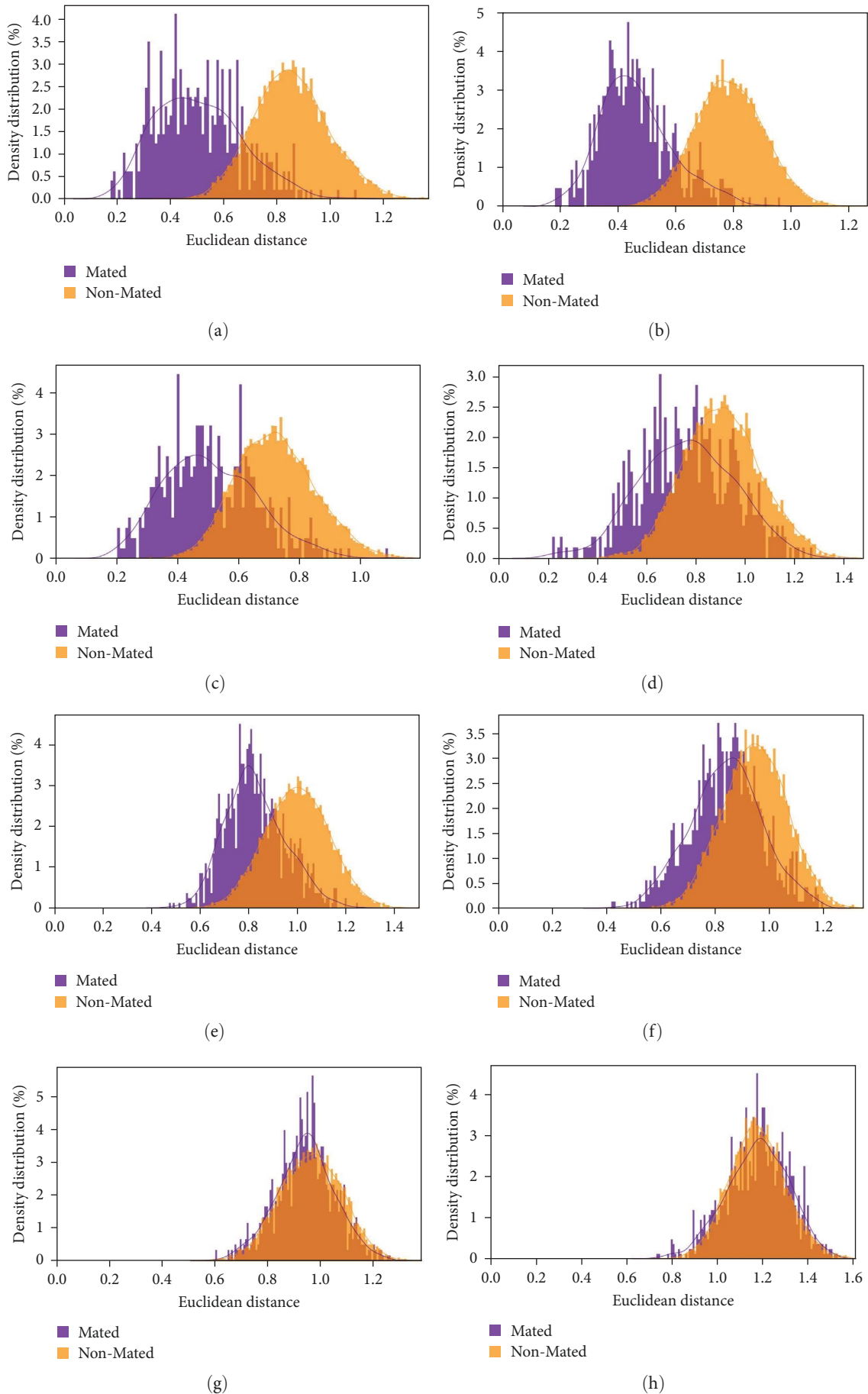


FIGURE 10: Euclidean distance histograms of single-device settings with CNN: (a) UTFV–UTFV; (b) ZkTeco–ZkTeco; (c) IDIAP–IDIAP; (d) NTNU–NTNU; (e) UTFV–ZkTeco; (f) UTFV–IDIAP; (g) IDIAP–ZkTeco; (h) NTNU–ZkTeco.

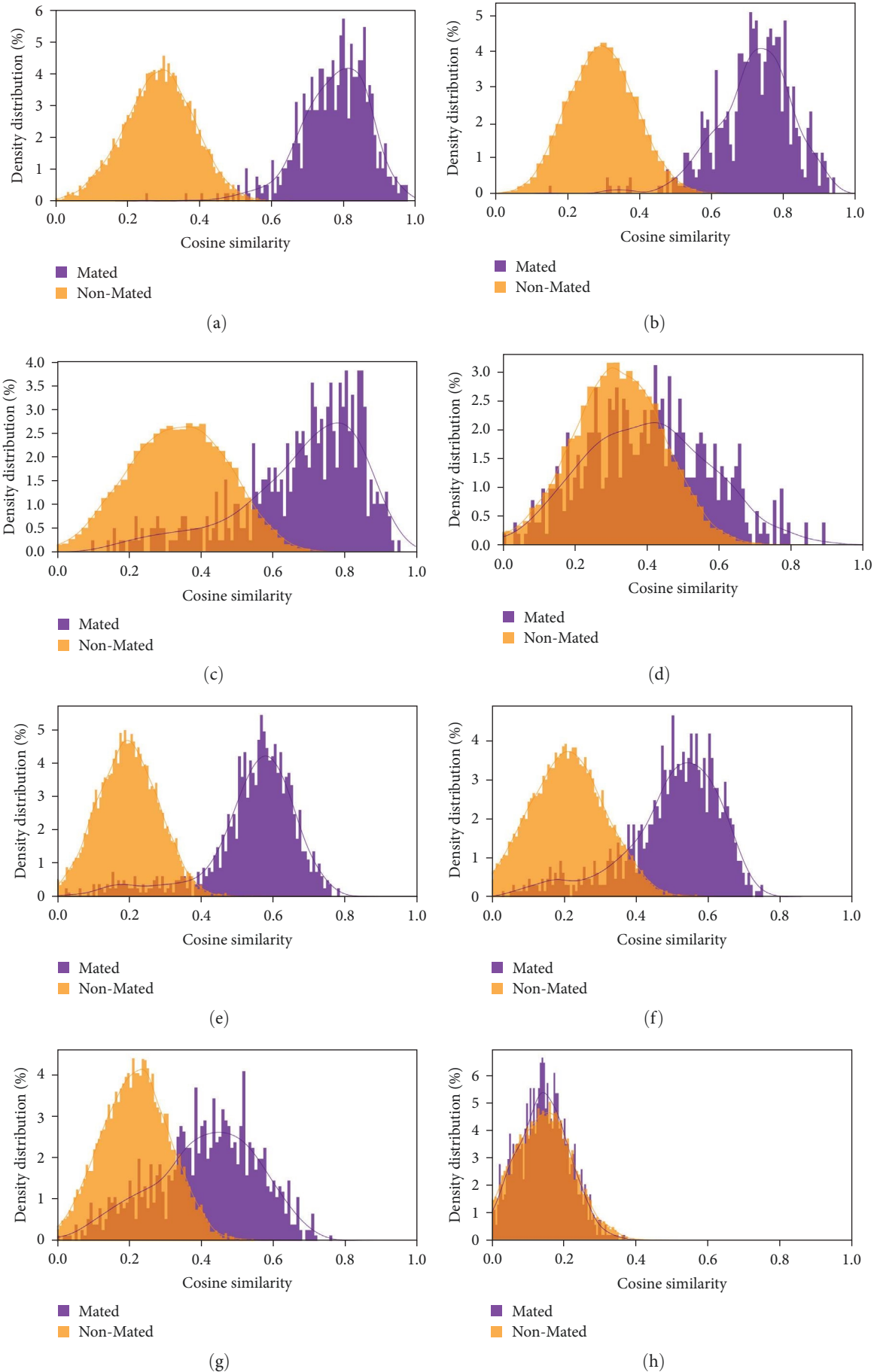


FIGURE 11: Cosine similarity histograms of single-device settings with CAE: (a) UTFV-UTFV; (b) ZkTeco-ZkTeco; (c) IDIAP-IDIAP; (d) NTNU-NTNU; (e) UTFV-ZkTeco; (f) UTFV-IDIAP; (g) IDIAP-ZkTeco; (h) NTNU-ZkTeco.

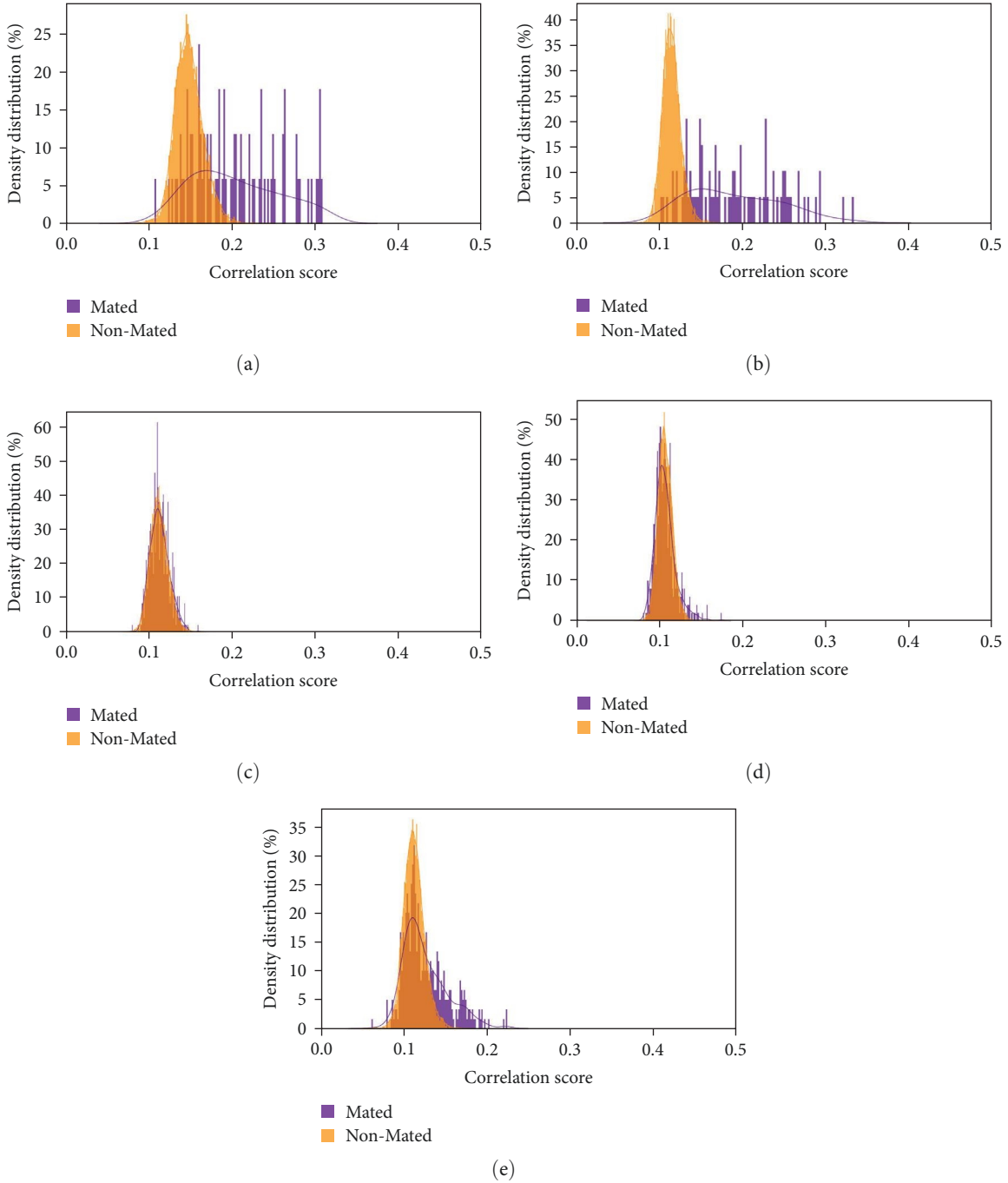


FIGURE 12: Correlation histograms of selected PFV device pairs: (a) PFV\_C-PFV\_C; (b) PFV\_L-PFV\_L; (c) UTFV-PFV\_C; (d) UTFV-PFV\_L; (e) PFV\_L-PFV\_C.

pair, while P-CAE achieves 7.5% EER on the same device pair. Moreover, on this device pair, the FMR1000 metric is substantially superior to the Miura method. This suggests that the P-CAE has fewer false non-match decisions for lower thresholds compared to the Miura method. The histogram plots of these two methods (Figures 9(e) and 11(e)) demonstrate that the P-CAE achieves a better separation of mated and non-mated cross-device pairs than the Miura method. On the other hand, the CNN indicates notably worse recognition performance compared to the other two methods. Especially on the

UTFV-ZkTeco pair, the difference is more eminent. While the P-CAE achieves 7.5% EER on this device pair, the CNN indicates 24.4% EER, which is more than three times worse compared to the P-CAE.

### 7. Discussion

Cross-device collected data poses several challenges for capturing device-agnostic vein image recognition. First and foremost, the data collection set-up lacks strict supervision due

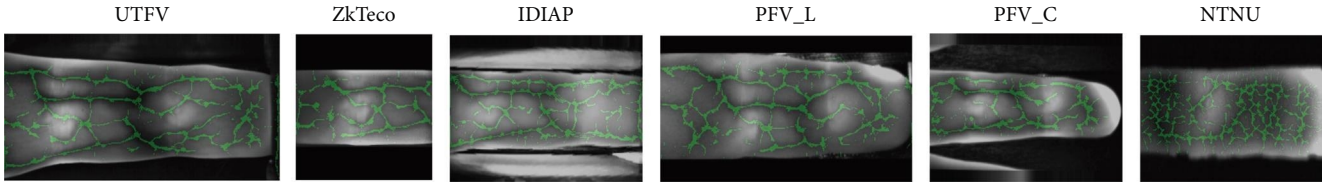


FIGURE 13: Finger vein images of different devices.

to the limited time allocated for the collection process, the number of participants, and the number of devices used for collection. Therefore, captured data involves translation and rotation in combination with uncontrolled ambient illumination. During the data collection, it was also observed that some subjects skipped some devices, intending to return to those devices later. However, as presented in Table 4, this intended return does not appear to have occurred for many subjects. Notably, the devices PFV\_L and PFV\_C exhibit clear instances of this phenomenon, with both the number of subjects and, particularly, the number of subjects returning for the second session are significantly lower than other devices. Moreover, the time limitations led to the oversight of the camera calibration step. Given the critical importance of camera calibration for cross-device comparisons, the omission of this step presents one of the primary challenges in cross-device comparisons within this dataset.

The challenges encountered during the data collection stage, particularly the lack of supervision, have a notable impact on the recognition performance across different devices. Devices such as PFV\_L and PFV\_C demonstrate noticeably poorer performance compared to what is reported in the literature. Previous studies report a recognition performance of 0.28%EER for the PFV\_L device [35] and 3.66% EER for PFV\_C [45] using the Miura method. Though the sample size for these devices is considerably smaller for these devices compared to the literature, the EER of 9.52% for PFV\_L and 22.6% for PFV\_C indicate the substantial impact of the data capturing process. Furthermore, Figure 13 illustrates that even when the same vein pattern is captured across different devices, the appearance of the vein patterns varies. For example, the image from the PFV\_C device exhibits on-axis rotation, resulting in a slightly different view of the same vein patterns. PFV\_C, being a contactless device, allows even a higher degree of on-axis rotations. Moreover, since this device lacks a finger rest, the distance between the finger and the camera varies between capturing sessions. On-axis rotations, together with fingers captured at varying distances, are found to be one of the contributing factors to the low mated pair scores for cross-device pairs. On the other hand, the image from the NTNU device does not exhibit a clear vein pattern compared to the images from the other devices. These observed differences serve as evidence to account for the variations in the comparison performances presented in Table 6, even though the same fingers are employed for comparison on each device.

Histogram plots (Figures 9–11) indicate that all three recognition methods face challenges in distinguishing between mated pairs and non-mated ones. The FNMR1000 metrics for cross-device comparisons, compared to single-device ones,

suggest that the recognition methods tend to have a significantly higher number of false non-match pairs compared to single-device scenarios. The following subsections discuss the factors causing this phenomenon for each recognition method.

*7.1. Miura Method.* This section discusses the factors affecting the comparisons with the Miura method. Figure 14(a) displays a good mated pair on the UTFV–UTFV device pair, while the remaining images in Figure 14 represent the same image pair on different device pairs. It is evident from Figure 14 that when the probe device is different from the reference device, comparisons are significantly affected despite the similarities between vein patterns. On the UTFV–ZkTeco (Figure 14(b)) pair, differences in the captured finger shape make it challenging to find a proper scaling factor for this device pair. On the UTFV device, fingers appear in their natural shape, which is wider at the root and narrower at the tip. In contrast, the ZkTeco device captures a fixed rectangular area on the finger. If the UTFV finger is wide, then the extracted vein patterns on the UTFV-ZkTeco device pair exhibit differences in terms of dimensions, which reduces the correlation between the image pairs. Similarly, the PFV\_C device, due to its contactless feature, captures fingers at varying distances, while the distance from the camera of a finger is fixed for the UTFV device. This affects the dimensions of the captured finger image on this device pair (Figure 14(e)). On-axis finger rotation, observed on the UTFV–IDIAP (Figure 14(c)) and the UTFV–PFV\_L (Figure 14(d)) device pairs, is another cause of low mated scores with the Miura method. On-axis rotation affects the vein appearance on the captured images due to changes in perspective. In such cases, the Miura method fails to find a high correlation despite the similarities between vein patterns.

Despite the high-quality images acquired with both PFV devices, their cross-device performances are noticeably worse compared to the other device pairs. Figure 12 illustrates how image pair correlation scores change with different PFV\_L and PFV\_C device pairs. Especially on the UTFV–PFV\_C device pair (Figure 12(c)), it is impossible to distinguish mated and non-mated pairs. Due to the contactless feature of the PFV\_C device, cross-device pairs involving the PFV\_C device exhibit differences in dimensions. Consequently, the Miura match results in poor comparisons for these image pairs, even though there are clear similarities between vein pairs.

*7.2. CNN.* Kuzu et al. [13] present an impressive recognition performance on the SDUMLA-HMT dataset using the CNN model. However, in this study, it is observed that the CNN is the least competitive recognition method among the three



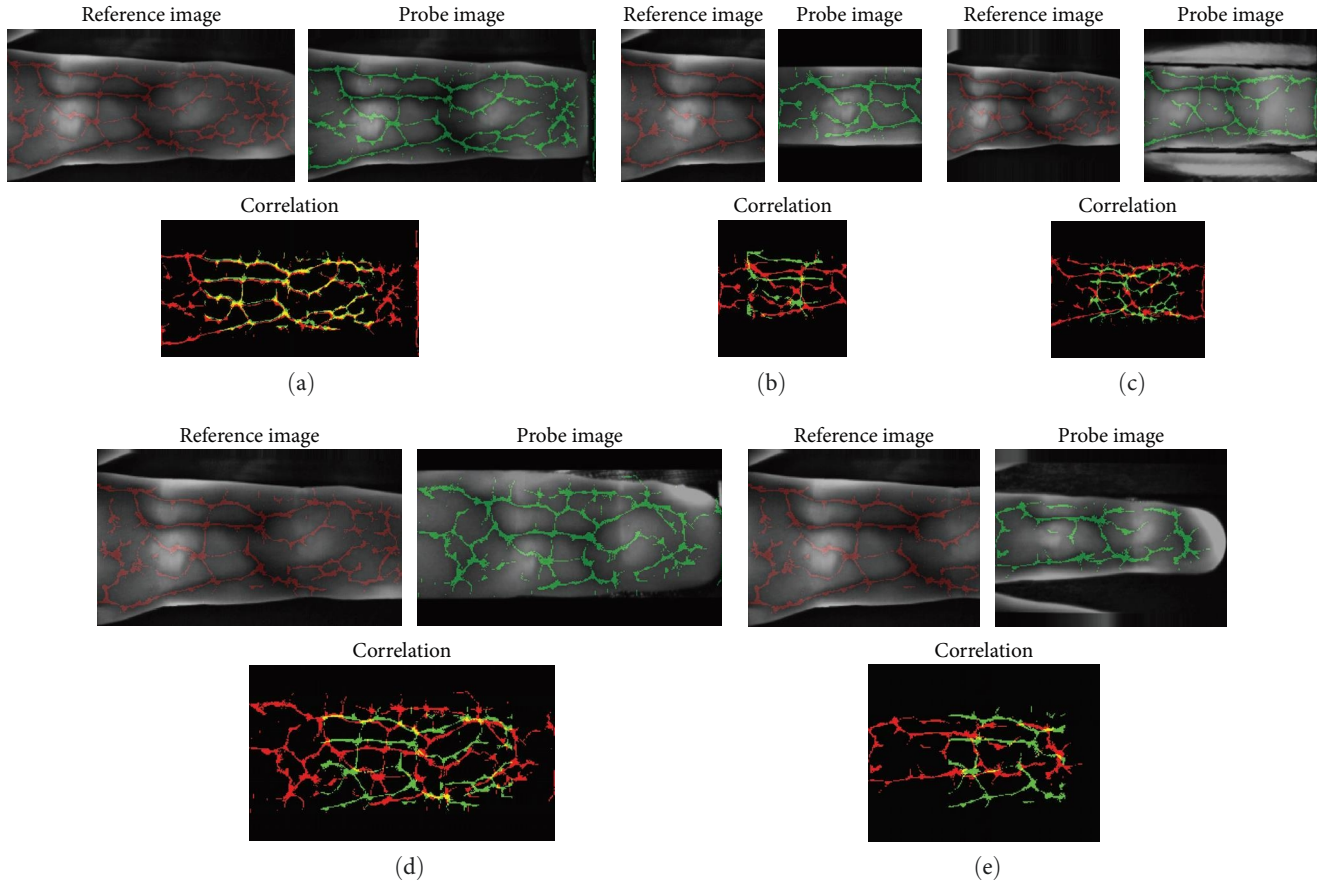


FIGURE 14: Mated pairs of different device pairs with Miura method: (a) UTFV-UTFV; (b) UTFV-ZkTeco; (c) UTFV-IDIAP; (d) UTFV-PFV\_L; (e) UTFV-PFV\_C.

TABLE 7: Impact of the training data on recognition performance of the CNN in EER (%).

|            | UTFV | ZkTeco | IDIAP | PFV_L | PFV_C | NTNU |
|------------|------|--------|-------|-------|-------|------|
| SDUMLA-HMT | 12.9 | 10.5   | 23.2  | 25.2  | 26.9  | 34.3 |
| UTFVP      | 8.4  | 9.8    | 26.0  | 21.3  | 24.9  | 32.6 |

methods. This difference in performance may be attributed to the training data not being representative enough of the evaluation set. As stated in the literature [11, 20], when the training data lacks similar properties to the evaluation data, the model fails to achieve satisfactory results. Tang et al. [11] conduct a study demonstrating that in such scenarios, the recognition performance can decrease by up to 6% in terms of EER. The CNN model used in this study accepts images at a resolution of  $228 \times 228$  pixels, while the resolution of SDUMLA-HMT images is  $240 \times 320$  pixels, which is considerably lower than the majority of the images in this study. This difference in compression ratio may be one of the reasons behind the poor performance of the CNN model. To assess the impact of training data on the recognition performance of the CNN, the model is trained with the UTFVP dataset. This dataset not only has higher-quality images but is also collected using the UTFV device. Table 7 compares the recognition performance when the CNN model is trained on the SDUMLA-HMT and the UTFVP datasets. Although

there is an improvement observed with the UTFVP dataset to the SDUMLA-HMT, the recognition performance on the UTFV device is still far from being competitive. When false non-match pairs of the UTFV device are examined, it is observed that they are affected by translations and illumination variations. This highlights the need for more cautious use of CNNs in challenging conditions like cross-device finger vein recognition compared to the other methods.

7.3. *P-CAE*. Although the Miura method outperforms the P-CAE in single-device settings, the CAE surpasses the Miura method on certain cross-device pairs, such as UTFV, ZkTeco, and IDIAP. Moreover, when considering the FMR1000 metric, the P-CAE reduces the false non-match rate by almost 50% compared to the Miura method on some device pairs. Figure 15(a) shows the comparison of the same mated pair with Figure 14(a) but with the P-CAE. Similar to the Miura method, the P-CAE faces challenges in comparing the image pairs involving the PFV\_L (Figure 15(d)), and the

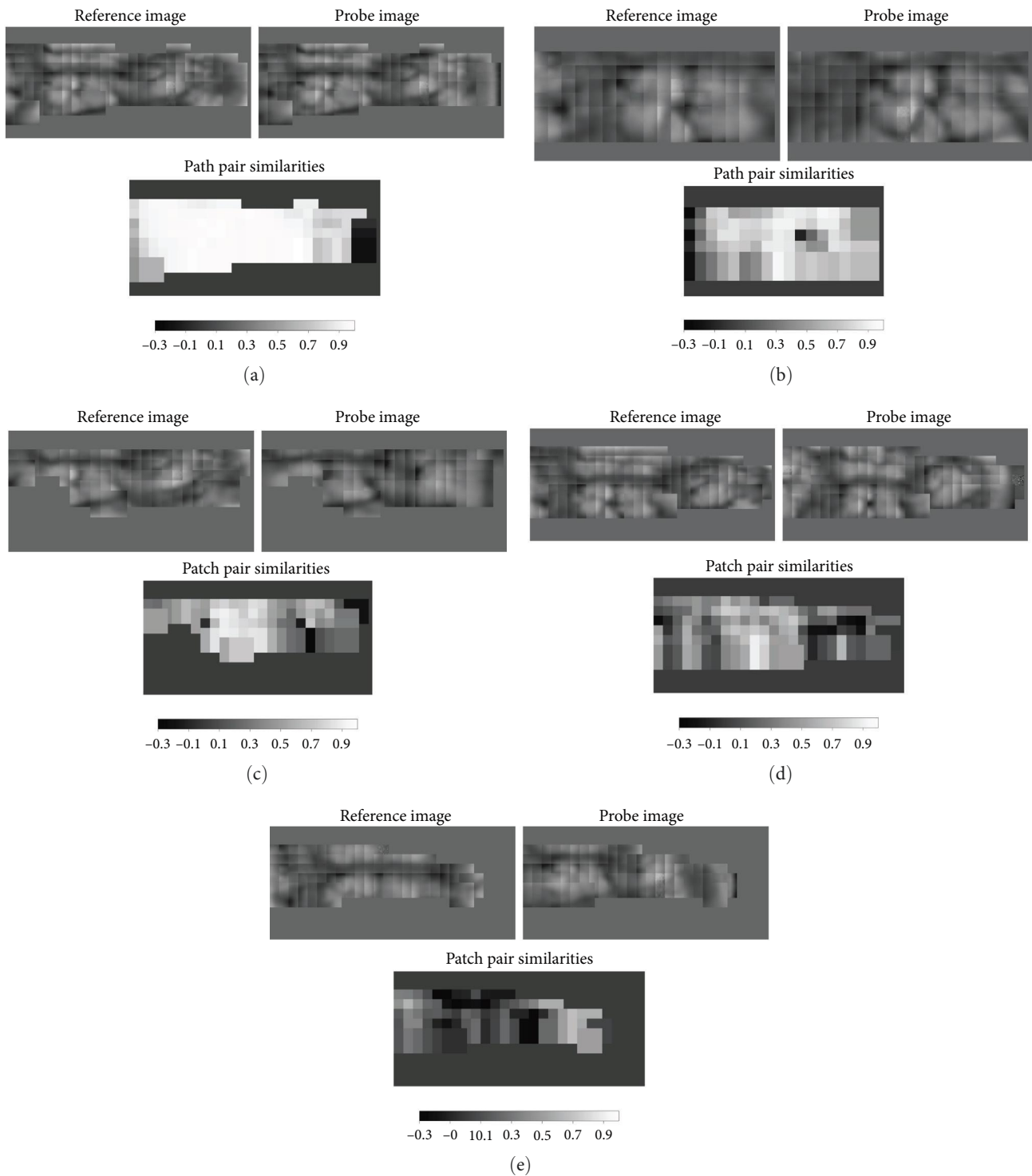


FIGURE 15: Mated pairs of different device pairs with CAE: (a) UTFV-UTFV; (b) UTFV-ZkTeco; (c) UTFV-IDIAP; (d) UTFV-PFV\_L; (e) UTFV-PFV\_C.

PFV\_C (Figure 15(e)) devices due to the translations observed on the image pairs. The patch-based approach is sensitive to translation errors. On the UTFV-PFV\_L pair (Figure 15(d)), slight alignment error leads to unsatisfactory comparisons, particularly at the fingertip. On the UTFV-PFV\_C pair, the differences in finger dimensions result in poor comparisons.

On the other hand, unlike the Miura method, the P-CAE presents a less sensitive behaviour to slight variations in finger vein patterns. On both UTFV-ZkTeco (Figure 15(b)) and UTFV-IDIAP (Figure 15(c)) pairs, despite the slight variations in the vein patterns, the P-CAE is still able to recognise the similarities between the image pairs. The

TABLE 8: Impact of the training data on the recognition performance of the P-CAE in EER (%).

|            | UTFV | ZkTeco | IDIAP | PFV_L | PFV_C | NTNU  |
|------------|------|--------|-------|-------|-------|-------|
| UTFVP      | 1.25 | 2.41   | 14.47 | 8.62  | 14.29 | 40.72 |
| SDUMLA-HMT | 1.52 | 5.93   | 25.38 | 11.0  | 23.04 | 38.71 |

strength of the P-CAE lies in its ability to learn representations not only of the vein structures but also of the finger background information. As a result, the P-CAE is more effective at handling slight variations in vein structures within patches compared to the Miura method. Moreover, the patch-based CAE provides more interpretable comparisons compared to the CNN. By examining the similarities between the patch pairs, it becomes relatively straightforward to explain matches and non-matches. This enables the identification of the areas that pose challenges for an accurate comparison.

In order to investigate the impact of the training data on the recognition performance of the P-CAE model, the datasets are evaluated on a model trained on the SDUMLA-HMT dataset. Table 8 indicates that almost all datasets performed worse with the model trained on SDUMLA-HMT. As mentioned before, the SDUMLA-HMT dataset involves low-quality images that mostly do not involve fine and detailed vein structures. Since the model cannot learn proper representations of fine vein structures, they may not be well encoded in the latent vector, even if they are present in the finger vein images such as UTFV or PFV\_L. The impact is even more on the lower-quality datasets such as IDIAP. Despite the poor performance, the results presented in Table 8 highlight the importance of the train dataset characteristics for the P-CAE. Despite its relatively flexible nature, the higher quality of the training data yields better recognition performance even on relatively low-quality datasets such as IDIAP.

## 8. Conclusion and Future Work

This study provides a comparative analysis of cross-device finger vein recognition across six different acquisition devices using a classical and two deep learning methods. It examines the factors influencing cross-device recognition performance. The findings of this study bring attention to the challenges facing cross-device finger vein recognition and underscore the need to employ standards for finger vein acquisition and sample quality assessment, similar to those implemented for fingerprint or iris recognition. Furthermore, the challenges presented in this study are expected to lead to new research areas on finger vein recognition.

The results on single-device pairs suggest that biometric sample capturing conditions and device properties have a substantial impact on the recognition performance. The recognition performance of the Miura method ranges from 0.57% EER (UTFV) to 32.6% EER (NTNU) on the same finger vein images. Additionally, the performances achieved on PFV\_L (9.52% EER) and PFV\_C (22.6% EER) are notably different from the performance presented in the literature (0.28% EER and 3.66% EER, respectively) [35, 45]. The dataset used in this study is acquired under limited supervision allowing us to explore a wider range of

variations in the captured images. The substantial differences observed in the recognition performances of different devices highlight the sensitivity of the recognition methods to the data acquisition conditions.

The findings on cross-device comparisons support the results presented in the literature [19, 20] and emphasise the need for well-established standards for finger vein sample quality assessment. The results indicate that the recognition methods cannot discriminate mated cross-device pairs from non-mated ones. Further examination of such mated pairs reveals that this problem stems from the differences between acquisition devices. For example, the rectangular shape of ZkTeco images makes it difficult to find a common scaling factor for the image pairs with the other devices. On-axis rotations and translation errors are found to be another main reason for poor cross-device recognition performance. Due to different device properties, variations observed on the cross-device pairs are higher than in single-device cases. Also, due to different contact features of the devices, translation errors are introduced by nature on some device pairs. Though none of the challenges presented in this work are new when it comes to finger vein recognition, cross-device recognition amplifies the impact and requires a different approach when comparing images from different devices.

Contrary to its superior performance reported in the literature, the CNN fails to achieve the same competitive results on the cross-device datasets. Further experiments imply that the performance of the CNN may depend on the training data characteristics, and the model used for authentication may need to be trained with a dataset containing image pairs more similar to those collected by the acquisition devices, which poses another challenge for cross-device finger vein recognition with this method. Moreover, interpreting the CNN outputs to explain mated and non-mated pairs is rather difficult compared to the other two methods. In light of these findings, the CNN model may necessitate additional attention to cross-device finger vein recognition. These findings imply the need for heightened focus on the CNN under challenging conditions, such as cross-device recognition.

Despite being trained on a different dataset than the evaluation data, the P-CAE outperforms the classical baseline method on some of the cross-device pairs and achieves comparable results on the others. For instance, in the UTFV-ZkTeco pair, the P-CAE indicates 7.95% EER and 18.47% FNMR1000, which is approximately 30% better than the baseline and three times better than the results presented in [19]. Though the patch-based approach is susceptible to translation errors, the analyses of image pairs suggest that the P-CAE exhibits greater robustness to slight variations in vein structures compared to the Miura method. Since each patch

is evaluated individually, the patch-based approach offers an enhanced explainability of mated and non-mated pairs in comparison to the CNN. Furthermore, the recognition performances indicate that the P-CAE provides a higher generalisation in finger vein comparison compared to the CNN. Considering the advantages, with an improved alignment approach, the P-CAE presents competitive and promising results, particularly under challenging conditions such as cross-device comparisons.

To summarise, finger vein images possess features that can be recognised across devices; however, realising the potential of cross-finger vein recognition is challenging without active implementation of standards such as ANSI/NIST-ITL [32], ISO/IEC 19784 [33], or ISO/IEC 39794-9 [34]. Hence, their practical integration is imperative inter-operable finger vein recognition. Cross-device comparisons amplify the impact of existing challenges on finger vein recognition, requiring further advancements in the recognition methods. On the other hand, despite the difficulties associated with cross-device finger vein recognition, the proposed patch-based P-CAE architecture shows promising potential in addressing some of the challenges inherent to cross-device comparisons. In light of these findings, this study holds significant promise for enhancing finger vein recognition systems and achieving inter-operability.

## Data Availability

The cross-device finger vein dataset is publicly available at <https://www.utwente.nl/en/eemcs/dmb/downloads/utcdfvp/>

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The PhD is funded by the Turkish Government Ministry of Education.

## References

- [1] N. Miura, A. Nagasaka, and T. Miyatake, "Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification," *Machine Vision and Applications*, vol. 15, no. 4, pp. 194–203, 2004.
- [2] N. Miura, A. Nagasaka, and T. Miyatake, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 8, pp. 1185–1194, 2007.
- [3] J. H. Choi, W. Song, T. Kim, S.-R. Lee, and H. C. Kim, "Finger vein extraction using gradient normalization and principal curvature," in *Image Process.: Machine Vision and Applications II*, vol. 7251, pp. 359–367, SPIE, 2009.
- [4] B. A. Rosdi, C. W. Shing, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, pp. 11357–11371, 2011.
- [5] N. Hu, H. Ma, and T. Zhan, "A new finger vein recognition method based on lbp and 2dpc," in *2018 37th Chinese Control Conference (CCC)*, pp. 9267–9272, IEEE, 2018.
- [6] Y. Matsuda, N. Miura, A. Nagasaka, H. Kiyomizu, and T. Miyatake, "Finger-vein authentication based on deformation-tolerant feature-point matching," *Machine Vision and Applications*, vol. 27, no. 2, pp. 237–250, 2016.
- [7] F. Liu, G. Yang, Y. Yin, and S. Wang, "Singular value decomposition based minutiae matching method for finger vein recognition," *Neurocomputing*, vol. 145, pp. 75–89, 2014.
- [8] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1816–1829, 2017.
- [9] C. Xie and A. Kumar, "Finger vein identification using convolutional neural network and supervised discrete hashing," *Deep Learning for Biometrics*, pp. 109–132, 2017.
- [10] S. Fairuz, M. H. Habaebi, E. M. A. Elsheikh, and A. J. Chebil, "Convolutional neural network-based finger vein recognition using near infrared images," in *2018 7th International Conference on Computer and Communication Engineering (ICCCCE)*, pp. 453–458, IEEE, 2018.
- [11] S. Tang, S. Zhou, W. Kang, Q. Wu, and F. Deng, "Finger vein verification using a Siamese CNN," *IET Biometrics*, vol. 8, no. 5, pp. 306–315, 2019.
- [12] R. S. Kuzu, E. Maiorana, and P. Campisi, "Vein-based biometric verification using transfer learning," in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 403–409, IEEE, 2020.
- [13] R. S. Kuzu, E. Maiorana, and P. Campisi, "Loss functions for cnn-based biometric vein recognition," in *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 750–754, IEEE, 2021.
- [14] B. Hou and R. Yan, "Convolutional auto-encoder based deep feature learning for finger-vein verification," in *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pp. 1–5, IEEE, 2018.
- [15] B. Hou and R. Yan, "Convolutional autoencoder model for finger-vein verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2067–2074, 2019.
- [16] M. Agnihotri, A. Rathod, D. Thapar, G. Jaswal, K. Tiwari, and A. Nigam, "Learning domain specific features using convolutional autoencoder: a vein authentication case study using siamese triplet loss network," in *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019)*, pp. 778–785, Science and Technology Publications, Lda, 2019.
- [17] L. Lin, H. Liu, W. Zhang, F. Liu, and Z. Lai, "Finger vein verification using intrinsic and extrinsic features," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–7, IEEE, 2021.
- [18] V. Bros, K. Kotwal, and S. Marcel, "Vein enhancement with deep auto-encoders to improve finger vein recognition," in *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, 2021.
- [19] T. Arican, R. Veldhuis, and L. Spreeuwiers, "Fingers crossed: An analysis of cross-device finger vein recognition," in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, 2022.
- [20] B. Prommegger, G. Wimmer, and A. Uhl, "Cross sensor finger vein recognition," in *2022 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, 2022.
- [21] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li, "Finger-vein authentication based on wide line detector and pattern normalization," in *2010 20th International Conference on Pattern Recognition*, pp. 1269–1272, IEEE, 2010.
- [22] L. Yang, G. Yang, Y. Yin, and X. Xi, "Finger vein recognition with anatomy structure analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1892–1905, 2018.

- [23] H. Qin, X. He, X. Yao, and H. Li, "Finger-vein verification based on the curvature in radon space," *Expert Systems with Applications*, vol. 82, pp. 151–161, 2017.
- [24] J. Zeng, Y. Chen, C. Qin et al., "A novel method for finger vein recognition," in *Biometric Recognition. CCBR 2019*, Z. Sun, R. He, J. Feng, S. Shan, and Z. Guo, Eds., vol. 11818 of *Lecture Notes in Computer Science*, pp. 46–54, Springer, 2019.
- [25] J. M. Song, W. Kim, and K. R. Park, "Finger-vein recognition based on deep densenet using composite image," *IEEE Access*, vol. 7, pp. 66845–66863, 2019.
- [26] X. Wang, H. Wang, Y. He, Y. Ding, and L. Tao, "Novel algorithm for finger vein recognition based on inception-resnet module," in *Eleventh International Conference on Digital Image Processing (ICDIP 2019)*, vol. 11179, pp. 367–375, Proceedings of the SPIE, 2019.
- [27] B. Prommegger, D. Söllinger, G. Wimmer, and A. Uhl, "Cnn based finger region segmentation for finger vein recognition," in *2022 International Workshop on Biometrics and Forensics (IWBF)*, pp. 1–6, IEEE, 2022.
- [28] W.-F. Ou, L.-M. Po, C. Zhou, P.-F. Xian, and J.-J. Xiong, "Gan-based inter-class sample generation for contrastive learning of vein image representations," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 2, pp. 249–262, 2022.
- [29] J. Huang, W. Luo, W. Yang, A. Zheng, F. Lian, and W. Kang, "Fvt: finger vein transformer for authentication," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.
- [30] Z. Chen, J. Liu, C. Cao, C. Jin, and H. Kim, "Fv-upatches: enhancing universality in finger vein recognition," 2022.
- [31] Z. Pan, J. Wang, Z. Shen, and S. Han, "Disentangled representation and enhancement network for vein recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 8, pp. 4164–4176, 2023.
- [32] K. Mangold, "Data format for the interchange of fingerprint, facial & other biometric information ansi/nist-1-2011 nist special publication 500-290," 2016.
- [33] "Information technology—biometric application programming interface—part 1: BioAPI specification," ISO/IEC, 2018, <https://www.iso.org/standard/70866.html>.
- [34] ISO/IEC JTC1 SC37 Biometrics, "ISO/IEC 39794-9: 2021 information technology extensible biometric data interchange formats—part 4: vascular image data," International Organization for Standardization, 2021.
- [35] C. Kauba, B. Prommegger, and A. Uhl, "Focussing the beam—a new laser illumination based data set providing insights to finger-vein recognition," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–9, IEEE, 2018.
- [36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, IEEE, 2009.
- [37] Y. Yin, L. Liu, and X. Sun, "Sdumla-hmt: a multimodal biometric database," in *Biometric Recognition. CCBR 2011*, Z. Sun, J. Lai, X. Chen, and T. Tan, Eds., vol. 7098 of *Lecture Notes in Computer Science*, pp. 260–268, Springer, Beijing, China, 2011.
- [38] T. Arican, R. Veldhuis, and L. Spreeuwens, "Finger vein verification with a convolutional auto-encoder," in *Proceedings of the 2021 Symposium on Information Theory and Signal Processing in the Benelux*, pp. 43–51, Werkgemeenschap voor Informatie- en Communicatietheorie (WIC), 2021.
- [39] T. Arican, R. Veldhuis, and L. Spreeuwens, "Unraveling finger veins: an improvement to unsupervised finger vein recognition with a convolutional autoencoder," in *Pre-Proceedings of the 2022 Symposium on Information Theory and Signal Processing in the Benelux*, pp. 43–51, Werkgemeenschap voor Informatie- en Communicatietheorie (WIC), 2022.
- [40] B. T. Ton and R. N. J. Veldhuis, "A high quality finger vascular pattern dataset collected using a custom designed capturing device," in *2013 International Conference on Biometrics (ICB)*, pp. 1–5, IEEE, 2013.
- [41] Zkteco, "FVP10r fingerprint and finger vein combined sensor," <https://zkteco.technology/en/product/fpv10r/>.
- [42] L. S. Momo and L. C. Torres, "Method and device for biometric vascular recognition and/or identification," 2019, Patent No. WO/2019/15254, Global ID SA <https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2019150254>.
- [43] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "A hough transform-based method for radial lens distortion correction," in *12th International Conference on Image Analysis and Processing, 2003. Proceedings*, pp. 182–187, IEEE, 2003.
- [44] R. Ramachandra, K. B. Raja, S. K. Venkatesh, and C. Busch, "Design and development of low-cost sensor to capture ventral and dorsal finger vein for biometric authentication," *IEEE Sensors Journal*, vol. 19, no. 15, pp. 6102–6111, 2019.
- [45] C. Kauba, B. Prommegger, and A. Uhl, "Combined fully contactless finger and hand vein capturing device with a corresponding dataset," *Sensors*, vol. 19, no. 22, Article ID 5014, 2019.
- [46] B. R. Hunt, "Super-resolution of images: algorithms, principles, performance," *International Journal of Imaging Systems and Technology*, vol. 6, no. 4, pp. 297–304, 1995.
- [47] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.
- [48] P. Normakristagaluh, L. J. Spreeuwens, and R. N. J. Veldhuis, "Finger-vein pattern recognition based on ICP on contours," in *40th WIC Symposium on Information Theory in the Benelux 2019*, pp. 97–101, Katholieke Universiteit Leuven, 2019.
- [49] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes. Sensor fusion iv: control paradigms and data structures," *The International Society for Optics and Photonics*, vol. 1611, pp. 586–606, 1992.
- [50] M. van der Spek and L. Spreeuwens, "Identification through finger bone structure biometrics," in *Proceedings of the 2021 Symposium on Information Theory and Signal Processing in the Benelux*, pp. 56–63, Werkgemeenschap voor Informatie- en Communicatietheorie (WIC), 2021.
- [51] S. Qiu, Y. Liu, Y. Zhou, J. Huang, and Y. Nie, "Finger-vein recognition based on dual-sliding window localization and pseudo-elliptical transformer," *Expert Systems with Applications*, vol. 64, pp. 618–632, 2016.