Towards Protecting and Enhancing Vascular Biometric Recognition methods via Biohashing and Deep Neural Networks

Hatef Otroshi Shahreza and Sébastien Marcel

Abstract—Biometric template protection has been a crucial concern in biometric recognition systems. This is because biometric characteristics are irreplaceable, so the compromised templates can also be used in other applications. On the other side, deploying template protection algorithms often affects the performance of biometric systems. In this paper, we consider both raw and pre-processed finger vein images and propose a novel deep-learning-based framework to protect biometric templates and enhance recognition performance. We use a deep convolutional auto-encoder structure to reduce the dimension of the feature space, and then secure templates by applying the Biohashing algorithm on the features extracted at the bottleneck layer of our auto-encoder. The experimental results indicate that the protected templates through our framework achieve superior performance than Biohash protected templates of the raw features in the normal scenario. In the stolen scenario, where the Biohashing key is stolen, our model yields far better performance than Biohashing of raw features extracted by previous recognition methods. We also evaluate the generalization of our proposed framework on other vascular biometric modalities. It is worth mentioning that we provide an open-source implementation of our framework so that other researchers can verify our findings and build upon our work.

Index Terms—Auto-encoder, Biohashing, biometrics, deep neural network, finger vein, template protection, vascular biometrics.

1 Introduction

Biometrics generally refers to establishing identity of a person based on their physiological (e.g., face, fingerprint, finger vein), behavioral (e.g., signature, speech), or chemical (e.g., DNA) attributes. While biometric authentication systems offer great convenience for the user, conventional authentication systems which use PINs, passwords, tokens, etc., are always in danger of being forgotten, disclosed or stolen. [1], [2]. On the other hand, the biometric attributes are ideally a consistent property of each person. Therefore, providing the credential by the biometric characteristics can bring strong assurance of authentication. However, an important challenge in biometric authentication systems is that, unlike conventional authentication credentials, the biometric samples of each individual are often subject to variations. This is particularly because the biometric data are measured by a sensor under inconsistent environmental conditions [1]. In addition, biometric credential underlies the privacy concerns regarding the applications of biometrics, and therefore causes challenges to the privacy of biometric authentication systems to prevent the compromise of biometric templates [3]. Hence, security and privacy are both two major challenges in each biometric authentication system.

Generally, in biometric recognition system, the user's biometric features are usually compared with the reference templates which are stored in the database during the enrollment stage. Then, upon the similarity score between the extracted features and reference template, the authentication decision is made. However, a critical privacy concern for such authentication systems is to protect the biometric templates stored in the database from being compromised. Indeed, the hacker can use the stolen template to reverse and find the finger image. The hacker can then use the finger

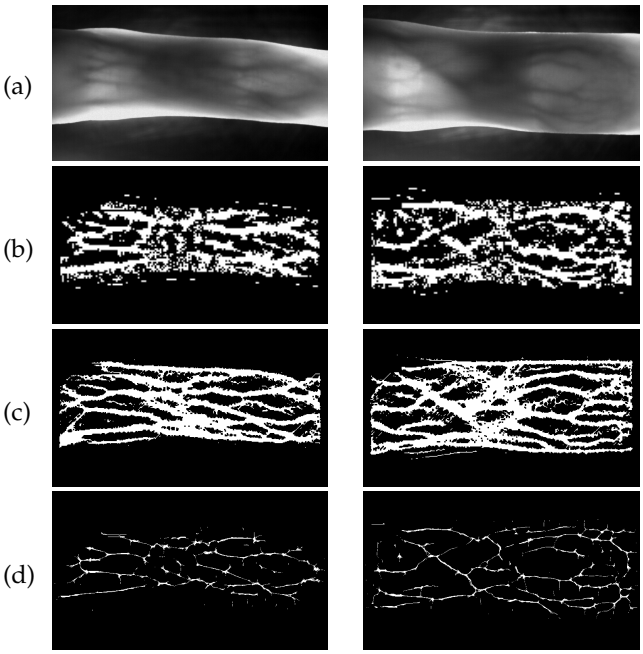


Fig. 1: Two sample finger vein images from two individuals in UTFVP dataset [4] and their corresponding Wide Line Detector (WLD) [5], Repeated Line Tracking (RLT) [6], and Maximum Curvature (MC) [7] features: (a) Finger vein image, (b) WLD, (c) RLT, (d) MC.

• *Authors are with the Biometrics Security and Privacy Group of Idiap Research Institute, Martigny, Switzerland. Hatef Otroshi Shahreza (hatef.otroshi@epfl.ch) is also affiliated with École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.*

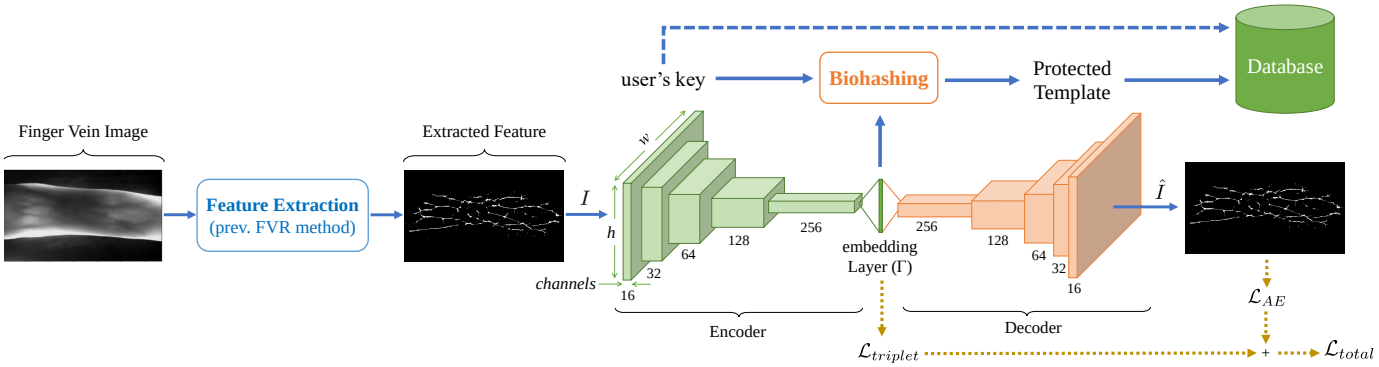


Fig. 2: Block-diagram of the proposed framework

image to impersonate the corresponding user even in other applications. Therefore, unprotected biometric templates heavily jeopardize the user's privacy, since the original biometric data is irrevocable if they are compromised [1].

To tackle the privacy challenges of such systems, several biometric template protection algorithms are proposed in the literature [1], [2], [8]. It is worth mentioning that there is often a trade-off between performance and privacy while deploying biometric template protection (BTP) schemes. In addition to security and performance challenges, BTP algorithms need to be cancelable as well. Indeed, if the protected biometric template is compromised, the template protection algorithms should be able to generate a new substitute protected template from the original data [9]. In cancelable template protection methods, a transformation function is often utilized which is dependent on a *key* [8]. Therefore, by changing the key, a new protected template can be generated for the same biometric feature. Among cancelable BTP algorithms, Biohashing [10] is one of the most well-known and widely studied methods [11]. In particular, Biohashing has been shown to be successful on different biometric characteristics (e.g. fingerprints [10], finger vein [11], iris [12], face [13], palm prints [14]).

In recent years, deep neural networks have demonstrated significant performance in many applications. In particular, deep learning methods have shown remarkable performance in biometric systems [15], [16]. In this paper, we consider finger vein images as biometric data, and propose a novel deep-learning-based framework to protect and enhance the previous finger vein recognition (FVR) methods in the literature. It is particularly noticeable since deploying BTP schemes, e.g., Biohashing [10], affects the performance of FVR systems [11]. For this end, in the proposed framework, we use the features extracted by the traditional FVR methods (e.g., figure 1), and train a deep convolutional auto-encoder with these features. Next, the trained auto-encoder is used to generate reduced-dimension features which are then given to Biohashing algorithm [10]. Figure 2 depicts the block-diagram of the proposed framework. In this paper, we use three well-known FVR approach, Wide Line Detector (WLD) [5], Repeated Line Tracking (RLT) [6], and Maximum Curvature (MC) [7], and use our proposed framework to protect them and enhance their performance. The simulation results on the UTFVP finger vein dataset [4] indicate that the protected templates generated by our framework achieve superior performance than Biohash protected templates of the raw features in the normal scenario. Furthermore, in the scenario where the Biohashing key is stolen, called *stolen* scenario, our framework achieves have far better performance than Biohashing protected templates from the raw features. In addition, we deploy raw finger vein images in our framework and use our framework as a secure FVR method. We also deploy our proposed framework using previous feature extractor on PUT Vein database [17] to evaluate the generalization of our framework on other vascular biometric modalities (i.e., palm and wrist.)

To better elucidate the contributions of this paper, we list them hereunder:

- We propose to use a deep convolutional auto-encoder to learn deep features in a reduced-dimension space from feature maps (pre-processed finger vein images) produced by traditional FVR methods, and then to apply Biohashing to these deep features to generate protected templates.
- We also propose to adopt a multi-term loss function, combining an auto-encoder loss and a triplet loss, to enhance the recognition accuracy despite the template protection.
- Besides, we use raw finger vein images (without any pre-processing) in our framework as a secure finger vein recognition method.

In the following sections, we review the related works in section 2. Next, we describe our proposed framework in section 3, and provide experimental analysis in section 4. Finally, the paper is concluded in section 5.

2 Related Works

We review papers in three relevant areas: Finger Vein Recognition (FVR), Finger Vein Template Protection, Deep Learning for Template Protection.

2.1 Finger Vein Recognition (FVR)

Generally, FVR systems rely on the structure of vascular patterns which are constructed by the blood vessels in the finger. Hence, features representing these patterns can be used in biometric systems. Most FVR systems use feature extraction methods that yield binary images for vascular patterns, e.g., [5], [6], [7], [18], [19], [20], [21]. These features

are often compared in a recognition system using Miura matcher [7].

Recently, some deep-learning-based approaches were proposed for FVR [22], [23], [24], [25]. In [22], a convolutional neural network (CNN) containing 5 convolutional layers is proposed which predicts the identity of users as separate classes in its output. In [23], authors propose a CNN including 9 convolutional layers and 500-neuron layer at the output, and it is trained with a triplet similarity loss function. Then, a supervised discrete hashing (SDH) algorithm is deployed on the extracted features to achieve fast retrieval. In [24], authors used a convolutional auto-encoder and extract features in the bottleneck layer. Then they use a support vector machine (SVM) classifier to classify the identity of users as the task of finger vein verification. In [25], a CNN was proposed which integrates both tasks of FVR and finger vein antispoofing into a single network using a multi-task learning approach. Their proposed CNN consists of two branches: the main branch is used for recognition and the auxiliary branch is used for presentation attack detection (PAD).

It is worth mentioning that since no template protection scheme is used in these methods, they are vulnerable to direct and indirect attacks [26]. In this paper, we consider some of the classical feature extractors which extract binary features from finger vein image. In particular, we use Wide Line Detector (WLD) [5], Repeated Line Tracking (RLT) [6], and Maximum Curvature (MC) [7] methods. Figure 1 illustrates example images of binary feature extracted by these methods.

2.2 Finger Vein Template Protection

Additional works try to increase protection of FVR systems. In [11], authors explore the effect of Biohashing on the performance of WLD, RLT, and MC feature extractors. Kirchgasser *et. al.* also proposed an alignment-free template protection method using Index-of-Maximum (IoM) [27] hashing [28], [29]. In [30], authors proposed a cancelable finger vein bio-cryptosystem. They extracted features from finger vein images using Gabor filter and linear discriminate analysis (LDA) techniques. The extracted features were then biohashed. Next, fuzzy commitment and fuzzy vault schemes were fused through AND fusion and OR fusion. In [31], authors use a user-specific random projection on the extracted biometric features to reduce the features dimension and generate protected templates. Then, they train a deep belief networks to match the protected templates. In [32], authors use Biohashing for the extracted features, and then apply a binary transformation on the Biohashing output. Finally, the results are given to a multilayer extreme learning machine (ML-ELM) for training and classification. In [33], a cancelable multi-biometric system, including fingerprint and finger-vein, was proposed. After extracting features from each mode, authors used a feature-level fusion strategy which includes a non-invertible transformation based on the Enhanced Partial Discrete Fourier Transform (EP-DFT) and three different fusion options.

Generally, there are four main criteria for evaluating biometric template protection schemes [34], [35], [36], [37]:

- *Cancelability*: Each biometric template protection scheme should be able to generate a new protected template and to revoke the enrolled protected templates if they are compromised.
- *Unlinkability*: Considering the cancelability criteria of a biometric template protection scheme, various protected templates could be generated from the same biometric data. Therefore, there should be no correlation between the generated templates (i.e., independent).
- *Non-invertibility*: It should be computationally impossible to invert the original biometric template from the protected template or to reconstruct the original biometric data.
- *Recognition Performance*: The performance of the recognition system can be affected by biometric template protection. Indeed, each template protection scheme needs to have accurate recognition.

2.3 Deep Learning for Template Protection

Deep neural networks are also used to generate protected biometric templates for other biometric characteristics. For example, in [38], authors propose a CNN with initial random weights and without any training process to extract features from finger dorsal images and then protect such features with Biohashing. In [39], authors trained a CNN as a classifier for knuckle recognition. Then, they deployed Biohashing on the output layer and generated protected templates. In [40], authors used a CNN which to extract features from face and iris images and then fuses these features in a fully connected layer, called joint representation layer, prior to the output layer that was used for the classification. Next, they used features in the joint representation layer and generated Biohash codes. In [41], authors use deep CNNs to map face images to maximum entropy binary (MEB) codes. Next, these codes are hashed by deploying a hash function. The main drawback of this system is that for enrolling a new individual a new training procedure is required. In [42], a deep CNN is trained to maximize the inter-class variations and minimize the intra-class variations. Next, by thresholding the network output, binarized codes are generated which are finally protected using a cryptographic hashing. In [34], a randomized network is utilized to generate protected templates using a user-specific key and the image of user's face. The user's key is not stored in the system, however it is used to generate a secure sketch, which is stored in the system, using an encoder during enrollment. In the recognition stage, the secure sketch along with a random partition of the biometric features are used to estimate the user's key by a decoder. Then, the estimated key is used to configure a randomized network with random activation and random permutation-flip to generate protected templates.

In this paper, we use the feature produced by previous FVR methods and give them to a deep convolutional auto-encoder to learn deep features in the bottleneck layer. Then, we apply Biohashing on these deep features to generate protected templates. Experiments show that the proposed framework enhance the performance of the previous FVR methods in addition to privacy protection. To our knowledge, this approach is original, and it has been never proposed to reduce the dimensionality of the features extracted by FVR methods with a deep AE prior standard Biohashing,

hence avoiding the enormous dimensionality reduction gap achieved by applying directly Biohashing to pre-processed images. In addition to protecting and enhancing previous FVR methods, we deploy the proposed convolutional auto-encoder on raw finger vein images (without prior feature extraction) and use our framework as a secure FVR method.

3 Proposed Framework

3.1 Overview

As explained in Section 1 and illustrated in the figure 2, our proposed deep-learning-based framework consists of a deep convolutional auto-encoder which extracts deep features at its bottleneck layer (so called embedding). In 3.2, we describe in further details the network structure, our multi-term loss function and the training process. After deep features are calculated at the embedding layer, we use the Biohashing algorithm to generate protected templates. This algorithm is also explained in section 3.3. Finally, for the recognition stage, the Biohash templates should be compared and scored, which is described in 3.4. It is noteworthy that the proposed framework is cancelable as Biohashing is cancelable [10], [11], [43]. Indeed, a new protected template can be generated any time using Biohashing with a new key.

We use the proposed framework to protect and enhance previous FVR methods. For this end, we train our proposed convolutional auto-encoder with the features extracted by the corresponding FVR methods. Indeed, by using the auto-encoder structure, we can considerably reduce the dimension of extracted features without loosing significant information. The new deep features can be protected by Biohashing efficiently since the dimension of features is small enough.

In addition, we use the proposed framework to introduce a new secure FVR method. For this end, we use raw finger vein images to train the proposed convolutional auto-encoder and directly extract features from finger vein images.

3.2 Auto-encoder

3.2.1 Network Structure

We use a convolutional auto-encoder that reduces the size of its input to the bottleneck layer (encoder), and then reconstruct the image (decoder). The encoder network consists of five convolutional layers with 16, 32, 64, 128, 256 filters, respectively. We use 3×3 kernel with stride 2 in each layer, which divides the spatial size by factor 2. Additionally, we use Batch normalization [44] after each convolution operation. Finally, we use a fully connected layer to get the embedding layer. For the decoder network, we use the transpose convolution layers. Except for the final layer, which has `sigmoid` function, we use the rectified linear unit (ReLU) for the other layers.

We should note that the size of the input image given to the network is the size of the extracted features by the corresponding FVR model. Further information about the size of the features extracted by different FVR methods are reported in section 4.

3.2.2 Multi-term Loss Function

To train the proposed network, we use a multi-term loss function. Let's consider I, $\hat{I}$, Γ as the input image, the reconstructed image, and the embedding layer, respectively. The total loss is

$$\mathcal{L}_{total} = (1-\alpha)\mathcal{L}_{AE} + \alpha\mathcal{L}_{triplet}, \tag{1}$$

where α is a hyper-parameter (in $[0,1]$ interval) to control the contribution of $\mathcal{L}_{AE}$ and $\mathcal{L}_{triplet}$, where $\mathcal{L}_{AE}$ and $\mathcal{L}_{triplet}$ are the auto-encoder loss, and the embedding triplet loss, respectively. For the auto-encoder loss, we use different loss functions considering the input of our framework. If the proposed framework is used to protect and enhance the performance of a FVR method which has binary features, we use Binary Cross Entropy (BCE) as the auto-encoder loss:

$$\mathcal{L}_{AE} = \mathcal{L}_{BCE} = -\sum_{i=1}^{w}\sum_{j=1}^{h} I(i,j)\log\hat{I}(i,j), \tag{2}$$

where w and h are the width and height of the input image, I, respectively. However, if the proposed framework is used as our proposed FVR, we use a different term which works with continues values of finger vein images. In this case, we use the l_2 norm of the auto-encoder error:

$$\mathcal{L}_{AE} = \|I - \hat{I}\|_2. \tag{3}$$

Furthermore, the embedding triplet loss is defined as:

$$\mathcal{L}_{triplet} = \left[||\Gamma_a - \Gamma_p||^2 - ||\Gamma_a - \Gamma_n||^2 + \beta\right]_+ \tag{4}$$

where Γ_a, Γ_p, and Γ_n are the values of the embedding layer for anchor, positive, and negative images, respectively [45], and β is also a margin which is enforced between positive and negative pairs which is set to 1 in our experiments.

3.2.3 Training Process

To train the proposed auto-encoder with our multi-term loss function, we use Adam [46] optimizer. We use the initial learning rate of 10^{-3}, and decrease the learning rate every 10 epochs. We use the Pytorch framework for the experiments.

For our experiments, we use the UTFVP finger vein dataset [4] which contains 1440 finger vein images with 672×380 resolution that have been collected from 60 individuals. We apply data augmentation technique to the training set by randomly adjusting each finger vein image with a combination of the following transformations:

- rotation [range: < 7 degree]
- width shift [range: $< 0.025\times$ image width]
- height shift [range: $< 0.025\times$ image height]
- channel shift (i.e., offset) [range: < 0.075]
- zoom [range: (0.95,1.05)]

3.3 Biohashing algorithm for Template Protection

As mentioned earlier, we use the Biohashing algorithm [10] to generate the protected templates. Let's consider Γ, indicating an unprotected biometric template calculated at the embedding layer of our auto-encoder. Then, the protected template, B, can be generated by algorithm 1 using Γ and user's key, k. The Biohash templates, B, and the user's key should be eventually stored at the system database during enrollment.

Algorithm 1 Biohashing algorithm for template protection

Inputs:
Γ : unprotected biometric template
M : length of the unprotected template (Γ)
m : length of the protected template
k : user's seed
Output: $B = \{b_i | i = 1, 2, ..., m\}$ binary BioHash protected template
Procedure:
Generate a set of pseudo-random vectors, $\{r_i \in \mathbb{R}^M | i = 1, 2, ..., m\}$, based on the user's seed, k.
Apply the Gram-Schmidt process to transform the generated pseudo-random vectors $\{r_i \in \mathbb{R}^M | i = 1, ..., m\}$ into an orthonormal set of matrices $\{r_{\perp i} \in | i = 1, ..., m\}$
Compute $\{\langle \Gamma, r_{\perp i} \rangle \in \mathbb{R} | i = 1, ..., m\}$ where $\langle ., . \rangle$ indicates inner product operation.
Compute m bits BioHash $\{b_i | i = 1, 2, ..., m\}$ from

$$b_i = \begin{cases} 0 & \text{if } \langle \Gamma, r_{\perp i} \rangle \leq \tau \\ 1 & \text{if } \langle \Gamma, r_{\perp i} \rangle > \tau \end{cases}, i = 1, ..., m,$$

where τ is a preset threshold.
End Procedure

3.4 Scoring and Comparing Biohash Templates

In the subsequent experiments, we will consider that FVR operated in verification mode only. In the enrolling stage, the protected templates for every individual are stored at the system database. For the verification stage, either verification or identification, the probe templates should be compared with the templates in the database. To find the score between the probe template and the model template, we use the Hamming distance between the Biohash templates.

4 Experiments and Discussion

4.1 Experiment Setup

As mentioned earlier, we use the publicly available finger vein UTFVP dataset [4] in our experiments. This dataset contains in total 1440 finger vein images which have been collected from 60 subjects. We used the training (subjects 1-10, 240 images), development (subjects 11-28, 432 images) and evaluation (subjects 29-60, 768 images) subsets of this dataset[1] as used in [11]. The training subset is used for training the convolutional auto-encoder in our framework, the development subset is used for threshold estimation in the recognition stage, and the evaluation subset is used for reporting the final results and further comparisons. We implemented Wide Line Detector (WLD) [5], Repeated Line Tracking (RLT) [6], and Maximum Curvature (MC) [7] algorithms to extract biometric features from finger vein images, and then apply our framework on the features extracted by these methods. Figure 1 shows two sample finger vein images and the corresponding WLD, RLT, and MC features for two individuals in the UTFVP dataset. Furthermore, table 1 reports the size of these features and also the execution time on a system with an Intel i7-7700K 4.2 GHz to extract these features from each image in UTFVP dataset.

1. The implementation of this division for the UTFVP dataset is available under NOM protocol at https://gitlab.idiap.ch/bob/bob.db.utfvp

TABLE 1: Size of the features extracted by WLD, RLT, and MC methods and their execution times on UTFVP dataset

	WLD	**RLT**	**MC**
Feature size	164×94	409×234	682×390
Execution Time	0.17	22.6	3.25

As mentioned in Section 1, we consider two scenarios in our experiments: the *normal* scenario and the *stolen* scenario. In the normal scenario, which is the expected scenario for most cases, each user's key is considered to be secret and not been disclosed. However, in the stolen scenario, the impostor has access to the genuine user's secret key and use it with the impostor's own finger vein template. While such a scenario is expected to happen rarely in practice, the system's vulnerability relies on the leakage of the user's secret key. To implement the stolen scenario, in the verification stage, we calculate the Biohash code of other users in the database using the same key as the genuine's key.

It is worth mentioning that we evaluate the performance of our proposed framework in the normal scenario and the stolen scenario, and we do not evaluate cancelability, unlinkablity, and non-invertibility characteristics. Because, our framework relies on Biohashing algorithm, and evaluation of these characteristics of Biohashing have been addressed already by [10], [11], [43], [47], [47], [48], [49], [50].

In our experiments, we have three different hyper-parameters including the length of the embedding layer in the AE ($L_{embedding}$), the length of Biohash templates ($L_{Biohash}$), and the value of α in equation 1 for controlling the contribution of different loss terms. For simplicity in our experiments, we consider $L_{embedding}$, $L_{Biohash}$, and α equal with 100, 100, and 0.1, respectively. Afterwards, we provide an ablation study to evaluate the effect of each of these hyper-parameters.

After evaluating the performance of our framework on previous FVR methods, in another experiment, we deploy our framework on the raw finger vein images (without any preprocessing) of UTFVP dataset and compare the results with enhanced versions of previous FVR methods.

In addition, in another experimental setup, we evaluate the performance of our framework on other vascular biometric modalities. For this end, we use PUT Vein dataset [17] which includes palm vein and wrist vein images. This dataset consists of 2400 images, where half of images contains palm vein images (1200 images) and another half contains wrist vein images (another 1200 images) which were acquired from both hands of 50 individuals. We consider the images from "right" hands of this dataset and divide it into two part, the first part (subject 1-25, 600 images) for training and development, and the second part (subject 26-50, 600 images) for evaluation[2].

We use Bob[3] toolbox [51], [52] and the open-source implementation[4] of Biohash protected finger vein verification systems in [11] for our experiments. The source code and the

2. The implementation of this division for the PUT Vein dataset is available under R_1 protocol at https://gitlab.idiap.ch/bob/bob.db.putvein

3. https://www.idiap.ch/software/bob/

4. https://gitlab.idiap.ch/bob/bob.chapter.fingerveins_biohashing

trained models from our experiments are publicly available to allow to reproduce our results[5]

4.2 Performance Evaluation for previous FVR methods

Figure 3 compares the receiver operating characteristic (ROC) curves of the protected and enhanced version WLD, RLT, and MC methods via our framework, namely WLD+AE+Biohash, RLT+AE+Biohash, and MC+AE+Biohash, respectively, against the corresponding Biohash protected templates of these methods, namely WLD+Biohash, RLT+Biohash, and MC+Biohash, respectively, in the normal and the stolen scenarios on the evaluation subset of the UTFVP dataset. In all ROC curves of this paper, the marked points which are connected with the dashed lines correspond to the threshold that leads to False Match Rate (FMR)=10^{-3} on the development subset. In figure 3, we also compare the mentioned ROC curves to WLD, RLT, and MC methods alone without template protection in the normal scenario. Furthermore, since the AE does the dimensionality reduction in our framework, we also compare the performance of our framework with the Principal Component Analysis (PCA) as a traditional dimensionality reduction technique. For this end, we use the features extracted by WLD, RLT, and MC methods and use the PCA algorithm prior to Biohashing, namely WLD+PCA+Biohash, RLT+PCA+Biohash, and MC+PCA+Biohash, respectively. We should note that for a fair comparison, we considered the dimension reduced by PCA to be consistent with that of AE. As this figure shows, the proposed framework achieves superior performance than Biohash protected versions of WLD, RLT, and MC methods, both in normal and in stolen scenarios. In the case of WLD and RLT methods, our method even outperforms unprotected versions in the normal scenario. Comparing our method with the PCA algorithm, while our method has competitive performance with PCA algorithm in the stolen scenario, our method achieves far better performance in the normal scenario.

In addition to the ROC curves, we compare the performance of the aforementioned methods in terms of False Match Rate (FMR), False Non-Match Rate (FNMR), and Equal Error Rate (EER) for the evaluation subset of the UTFVP dataset in normal scenario and stolen scenario, which is reported in the table 2. Please note that for the values in this table, the threshold for each method is selected individually in the way that we achieve minimum EER on the development subset. This table also shows that our method achieves the best performance in the stolen scenario in terms of FMR, FNMR, and EER. However, in the normal scenario, our method has competitive performance with Biohash protected versions of the mentioned FVR methods.

4.3 Ablation Study

In section 4.2, we evaluated the performance of the proposed framework in protecting and enhancing the performance of previous FVR methods. We considered values of the hyper-parameters (including the length of the embedding layer in the AE ($L_{embedding}$), the length of Biohash templates ($L_{Biohash}$), and the value of α in equation 1) to be fixed. In this section, we evaluate the effect of each of these hyper-parameters on the performance of our framework by investigating the ROC curves in the normal and the stolen scenarios on the evaluation subset of the UTFVP dataset. For this end, we consider WLD method, which has a smaller feature size and is faster to be calculated (see table 1), and build our experiments upon WLD.

5. Source code: https://gitlab.idiap.ch/bob/bob.paper.tbiom2021_protect_vascular_dnn_biohash

TABLE 2: Comparing the performance of previous FVR methods with their Biohash protected templates and their enhanced version via our proposed framework in normal scenario and stolen scenario in terms of FMR, FNMR, and EER on the evaluation subset of UTFVP dataset. (Note that the best performance is **emboldened**)

method	Normal Scenario			Stolen Scenario		
	FMR	FNMR	EER	FMR	FNMR	EER
WLD	3.8%	3.8%	3.8%	-	-	-
WLD + Biohash	1.4%	1.7%	1.5%	34.3%	44.0%	39.1%
WLD + PCA + Biohash	6.8%	8.7%	7.8%	12.4%	11.1%	11.7%
WLD + AE + Biohash	**0.8%**	**1.3%**	**1.1%**	**8.5%**	**10.3%**	**9.4%**
RLT	4.6%	4.6%	4.6%	-	-	-
RLT + Biohash	**0.5%**	**0.7%**	**0.6%**	46.8%	34.4%	40.6%
RLT + PCA + Biohash	9.7%	10.0%	9.9%	13.5%	13.2%	13.3%
RLT + AE + Biohash	**0.5%**	0.9%	0.7%	**12.7%**	**13.5%**	**13.1%**
MC	**2.3%**	2.3%	**2.3%**	-	-	-
MC + Biohash	2.9%	2.3%	2.6%	42.1%	53.0%	47.5%
MC + PCA + Biohash	24.1%	22.4%	23.3%	21.8%	22.7%	22.2%
MC + AE + Biohash	**2.3%**	**2.2%**	**2.3%**	**14.2%**	**13.4%**	**13.8%**

4.3.1 Evaluating the effect of $L_{embedding}$

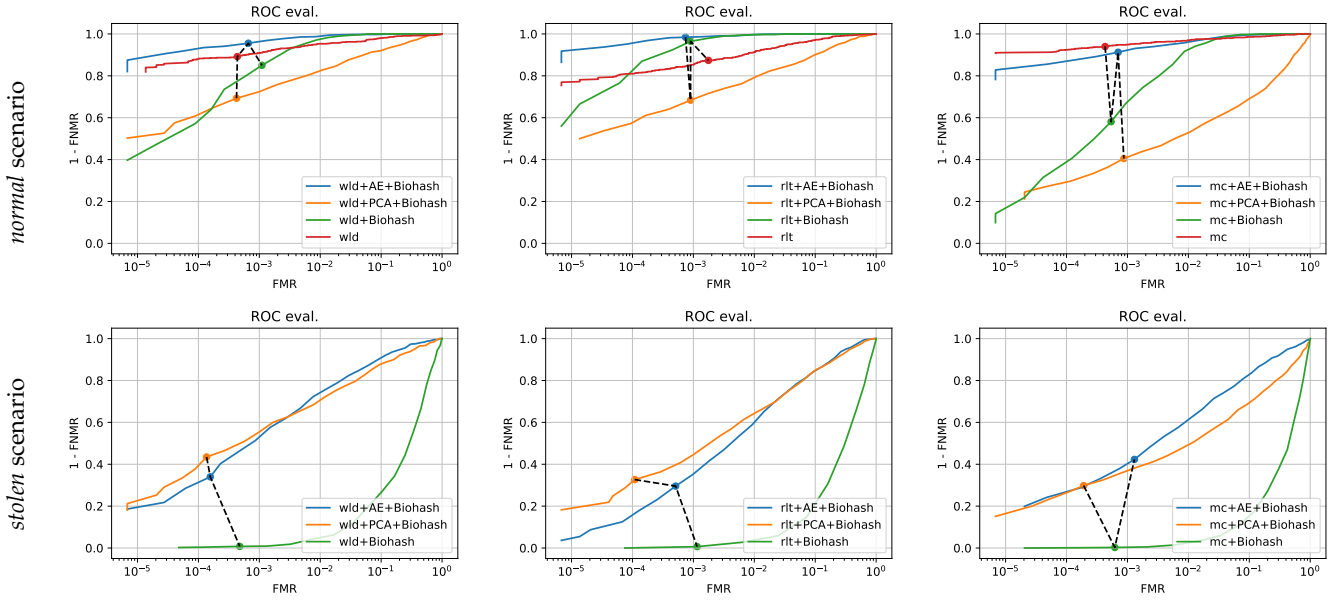
To evaluate the effect of the length of the embedding layer in the AE ($L_{embedding}$), we evaluate the performance of our framework with values 50, 100, 200, 500, and 1000 for $L_{embedding}$. Figure 4, represents the ROC curves of our framework for different values of $L_{embedding}$. As this figure shows, generally, a higher $L_{embedding}$ leads to superior performance in the normal scenario but inferior performance in the stolen scenario except for $L_{embedding} = 50$ and 100. This figure also shows that for $L_{embedding} = 100$, we achieve superior performance for both normal and stolen scenarios. Meanwhile, the performance for $L_{embedding} = 50$ is subordinate for both normal and stolen scenarios.This might be due to the fact that embedding layer of length 50 is not enough to represent finger vein features.

4.3.2 Evaluating the effect of $L_{Biohash}$

Similar to $L_{embedding}$, we evaluate the effect of the length of Biohash templates ($L_{Biohash}$) by varying its value between 30, 50, 80, 100, 500, and 1000. Figure 5, illustrates the ROC curves of our framework for different values of $L_{Biohash}$. As this figure shows, increasing the value of $L_{Biohash}$ above 100 does not significantly change the performance of our method neither in the normal scenario nor in the stolen scenario. However, decreasing the value of $L_{Biohash}$ lower than 100 degrades the performance of our framework in both normal and stolen scenarios.

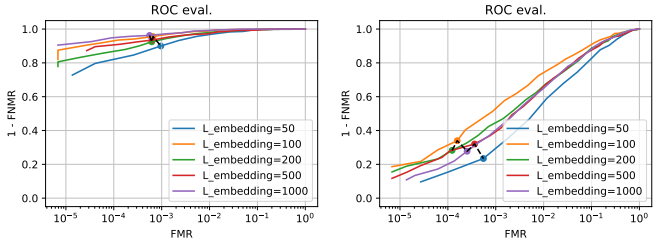
4.3.3 Evaluating the effect of α in equation 1

The hyper-parameter α in equation 1 controls the contribution of different terms in the loss function. To evaluate the effect of each loss term on the performance of our framework, we vary the value of α in $[0, 1]$ interval. Figure 6, illustrates the ROC curves of our framework for different values of α.



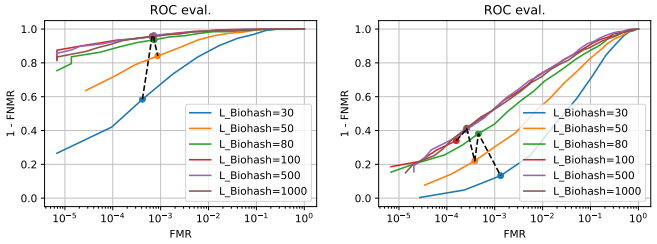
(a) WLD (b) RLT (c) MC

Fig. 3: Comparison of ROC curves of previous FVR methods with their Biohash protected templates, Biohash protected of their PCA transformation, and their protected version via our proposed framework in normal scenario (first row) and stolen scenario (second row): a)WLD, b)RLT, c)MC. The marked points which are connected with the dashed lines in each plot correspond to the threshold that leads to FMR=10^{-3} on the development subset.



(a) *normal* scenario (b) *stolen* scenario

Fig. 4: Evaluating the effect of $L_{embedding}$: a) normal scenario, b) stolen scenario. The marked points which are connected with the dashed lines in each plot correspond to the threshold that leads to FMR=10^{-3} on the development subset.



(a) *normal* scenario (b) *stolen* scenario

Fig. 5: Evaluating the effect of $L_{Biohash}$: a) normal scenario, b) stolen scenario. The marked points which are connected with the dashed lines in each plot correspond to the threshold that leads to FMR=10^{-3} on the development subset.

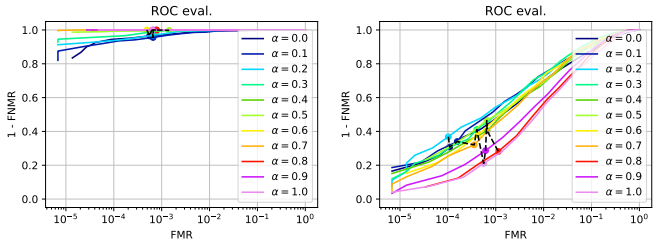
As this figure indicates, in the normal scenario, increasing the value of α enhances the performance of our framework. On the other hand, in the stolen scenario, increasing the value of α decreases the performance of our method. Therefore, the value of α which leads to the best performance in the normal scenario has the worst performance in the stolen scenario. We should also note that $\alpha = 1.0$ practically eliminates the effect of the decoder part of our auto-encoder network in the training process. Therefore, as depicted in figure 6, it leads to high performance in the normal scenario, but very poor performance in the stolen scenario.

4.4 Using our framework as a FVR method

In another experiment, we use raw finger vein images as the input to our convolutional auto-encoder and used the extracted features in the embedding layer for generating protected templates using Biohashing. Figure 7 compares the ROC curves of this setup, namely img+AE+Biohash, with protected and enhanced version WLD, RLT, and MC methods via our framework. As this figure shows, using raw finger vein images leads to superior performance in the normal scenario, but competitively inferior performance in the stolen scenario.

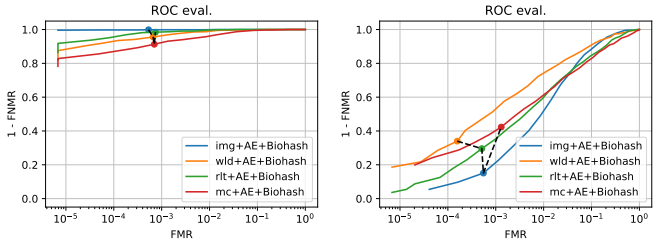
4.5 Palm and Wrist Vein Recognition

As mentioned earlier, to evaluate the generalization of our framework for other modalities, in another experimental setup, we use our framework for palm and wrist images from the PUT Vein database. Table 3 compares the performance of the proposed for WLD, RLT, MC methods as well as raw images on the evaluation subset of PUT Vein dataset in terms of FMR, FNMR, and EER. As this table shows, in general, our framework enhances the performance



(a) *normal* scenario (b) *stolen* scenario

Fig. 6: Evaluating the effect of α: a) normal scenario, b) stolen scenario. The marked points which are connected with the dashed lines in each plot correspond to the threshold that leads to FMR=10^{-3} on the development subset.



(a) *normal* scenario (b) *stolen* scenario

Fig. 7: Comparison of ROC curves of our framework in two modes: 1) given the raw finger vein images as the input of auto-encoder, and 2)given the features extracted from WLD, RLT, and MC as the input: a) normal scenario, b) stolen scenario. The marked points which are connected with the dashed lines in each plot correspond to the threshold that leads to FMR=10^{-3} on the development subset.

of Biohash protected versions of WLD, RLT, MC methods for palm and wrist data in both normal and stolen scenarios.

TABLE 3: Comparing the performance of the proposed framework on PUT Vein dataset. (Note that the best performance is **emboldened**)

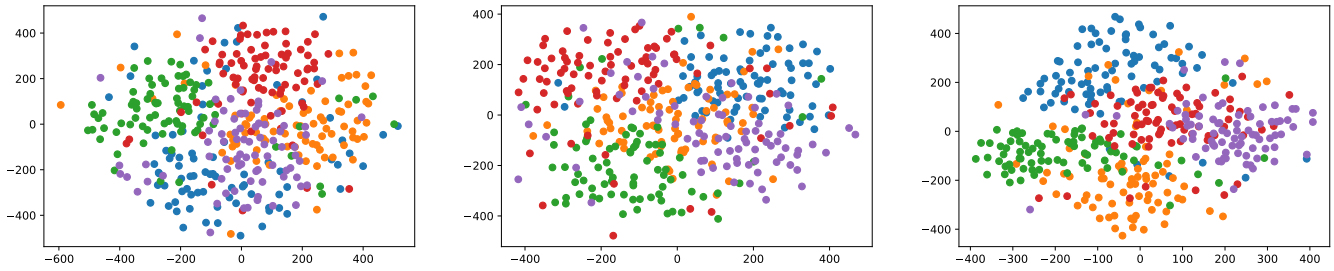
Data	method	Normal Scenario			Stolen Scenario		
		FMR	FNMR	EER	FMR	FNMR	EER
Palm	WLD	11.6%	11.6%	11.6%	-	-	-
	WLD + Biohash	3.5%	3.5%	3.5%	43.9%	50.0%	47.0%
	WLD + AE + Biohash	**1.7%**	**2.1%**	**1.9%**	**29.3%**	**30.0%**	**29.7%**
	RLT	37.5%	37.5%	37.5%	-	-	-
	RLT + Biohash	1.4%	1.1%	1.2%	41.3%	57.0%	49.1%
	RLT + AE + Biohash	**0.1%**	**0.0%**	**0.0%**	**33.9%**	**38.5%**	**36.2%**
	MC	34.5%	34.5%	34.5%	-	-	-
	MC + Biohash	2.1%	1.8%	1.9%	46.7%	51.4%	49.1%
	MC + AE + Biohash	**0.0%**	**0.1%**	**0.1%**	**43.0%**	**38.6%**	**40.8%**
	img + AE + Biohash	**0.0%**	**0.0%**	**0.0%**	40.5%	43.1%	41.8%
Wrist	WLD	27.0%	27.0%	27.0%	-	-	-
	WLD + Biohash	3.2%	4.5%	3.9%	44.8%	52.5%	48.7%
	WLD + AE + Biohash	**0.7%**	**0.5%**	**0.6%**	**30.7%**	**36.9%**	**33.8%**
	RLT	36.4%	36.4%	36.4%	-	-	-
	RLT + Biohash	1.4%	1.8%	1.6%	41.5%	53.9%	47.7%
	RLT + AE + Biohash	**0.0%**	**0.0%**	**0.0%**	**35.4%**	**37.6%**	**36.5%**
	MC	34.5%	34.5%	34.5%	-	-	-
	MC + Biohash	2.9%	2.0%	2.5%	**45.2%**	53.0%	49.1%
	MC + AE + Biohash	**0.1%**	**0.0%**	**0.0%**	46.1%	**39.2%**	**42.7%**
	img + AE + Biohash	0.7%	0.6%	0.7%	48.9%	46.1%	47.5%

4.6 Discussion

As shown in figure 3 and also observed in [11], Biohashing decreases the performance of WLD, RLT, and MC methods. In the normal scenario, we observe a considerable drop in the performance of Biohash protected versions of these FVR methods than their unprotected versions for low FMR thresholds. Besides, in the stolen scenario, the poor performance of Biohash protected templates indicates serious vulnerability of such systems to the reveal of users' keys. Comparing our proposed framework with Biohashing in the normal scenario, our framework achieves competitive performance for high FMR thresholds and also much superior performance for low FMR thresholds. In addition, in the stolen scenario, our method has far better performance than Biohash protected versions of the mentioned FVR methods. In fact, we reduced the dimension of extracted features through our framework that helps prevent the enormous dimensionality reduction gap caused by directly applying Biohashing to pre-processed images. While traditional dimensionality reduction techniques such as PCA can help to generate features in the lower dimension, experiments show the superiority of auto-encoder in the recognition performance. For instance, in the case of WLD, as shown in figure 3 and table 2, our framework achieves better performance than WLD, WLD+Biohash, and WLD+PCA+Biohash. To interpret the performance of our method, we use T-SNE technique to visualize the features prior to Biohashing in WLD+Biohash, WLD+PCA+Biohash, and WLD+AE+Biohash. Figure 8 illustrates the 2D representation of WLD, WLD+PCA, and WLD+AE of 5 different identities in the UTFVP dataset. As this figure shows, WLD could not completely separate the identity of finger vein images. Nonetheless, the features in WLD+PCA and WLD+AE could better determine the identity. In particular, the identities are better distinguised in WLD+AE. Therefore, it is expectable to achieve better performance with the features extracted in the embedding layer of our auto-encoder.

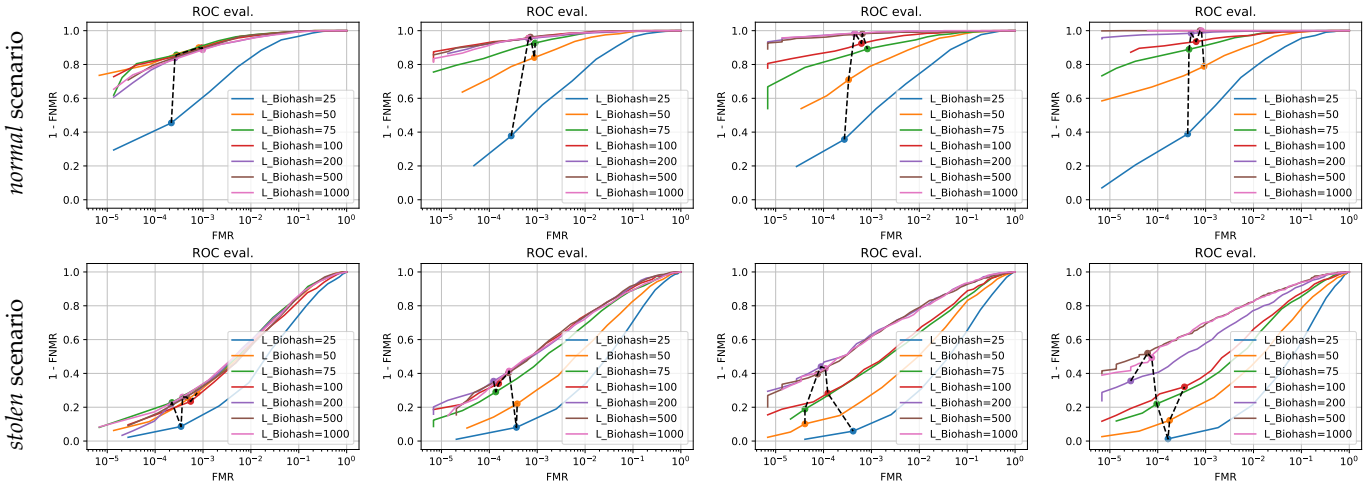
As seen in section 4.3, adapting hyper-parameters changes the performance of our proposed framework. Experiments show that adapting and choosing suitable values for $L_{embedding}$ and α is indeed a trade-off between the performance in the normal and stolen scenarios (see figure 4 and figure 6, respectively). Meanwhile, we notice that $L_{embedding}$ should be greater than 50 to achieve sufficient performance in both normal and stolen scenarios. However, our experiments in section 4.3.2 suggest that we can find a lower band for $L_{Biohash}$ where decreasing the value of $L_{Biohash}$ less than that value degrades the performance of our framework in both normal and stolen scenarios while increasing the value of $L_{Biohash}$ above that lower band does not significantly change the performance of our method neither in the normal scenario nor in the stolen scenario. To empirically find that lower band, in another experiment, we change $L_{embedding}$ between 50, 100, 200, and 500, and in each case, we evaluate the performance of our framework for different values of $L_{Biohash}$ between 25, 50, 75, 100, 200, 500, and 1000. Figure 9 represents the ROC curves of our framework for the aforementioned configurations[6]. As this figure demonstrates, for each $L_{embedding}$, opting the value of $L_{Biohash}$ less than $L_{embedding}$ degrades the performance of our framework in both normal and stolen scenarios. Therefore, this experiment shows that the lower band for $L_{Biohash}$ in our framework to have the higher performance

6. In this experiment, like section 4.3.2, we use WLD features extracted from images of UTFVP dataset ($\alpha = 0.1$).



(a) WLD (b) WLD+PCA (c) WLD+AE

Fig. 8: 2D representation of extracted features for 5 different identities with a)WLD, b)WLD+PCA, and c)WLD+AE methods. Different colors illustrate different identities. The axes denote the reduced dimensions to represent each feature using the T-SNE technique.



(a) $L_{embedding} = 50$ (b) $L_{embedding} = 100$ (c) $L_{embedding} = 200$ (d) $L_{embedding} = 500$

Fig. 9: Evaluating the performance of our framework with different values of $L_{embedding}$ and $L_{Biohash}$ in protecting WLD features on UTFVP dataset in normal scenario (first row) and stolen scenario (second row): a)$L_{embedding} = 50$, b)$L_{embedding} = 100$, c)$L_{embedding} = 200$, d)$L_{embedding} = 500$. The marked points which are connected with the dashed lines in each plot correspond to the threshold that leads to FMR=10^{-3} on the development subset.

with smaller length of Biohashing is $L_{embedding}$.

Our experiment in section 4.4 also shows that using raw image in the input of the auto-encoder leads to superior performance in the normal scenario. Nevertheless, in the stolen scenario, the pre-processed images achieve better performance when they are used as the input of our framework. Meanwhile, we should note that each of the pre-processing methods requires more execution time, as reported in table 1.

It is noteworthy that the execution of the encoder in our framework is very fast. Table 4 reports the execution time to get the embedding features from the encoder in our proposed framework and the required memory for the encoder network in our framework for WLD, RLT, and MC methods on the images from UTFVP dataset[7]. Comparing the execution times in table 4 with the required time to extract each feature reported in table 1, we can conclude that our proposed framework is quite fast and its execution time is almost negligible respecting the run time for feature extraction in mentioned FVR methods. In a nutshell, considering the enhancement in both normal and stolen scenarios and the high speed of our framework, it is much worth applying our framework to previous FVR methods.

Last but not least, experiments in section 4.5 show that our framework enhanced the performance of protected versions of WLD, RLT, and MC methods in both normal and stolen scenarios. Moreover, this experiment confirms the generalization capability of our framework for other

TABLE 4: The average execution time (second) to get embedding features from the encoder and the required memory for encoder network in our framework

	WLD	**RLT**	**MC**
Encoder's Exe. Time*	0.003 (0.0002)	0.02 (0.0008)	0.06 (0.004)
Encoder's req. Memory	3.1 MB	10.9 MB	27.5 MB

*The values are for the CPU (GPU) implementation.

7. The execution times reported in table 1 and 4 are achieved by a system equipped with an Intel i7-7700K 4.2 GHz CPU and an NVIDIA 1080 Ti GPU.

vascular biometric modalities such as palm and vein images.

5 Conclusion

In this paper, we considered finger vein images as biometric data and proposed a deep-learning-based framework to protect and enhance the previous finger vein recognition methods by reducing the dimension of biometric features using a DNN and then protecting the reduced-dimension features with Biohashing. To this end, we used the raw finger vein images and the extracted features from previous FVR methods to train a deep convolutional auto-encoder with a multi-term loss function. We used the auto-encoder to extract reduced-dimension features in the bottleneck layer (embedding layer). Finally, we generated protected templates from deep features using Biohash. The simulation results indicate that the protected templates generated by our framework achieve superior performance than both Biohash protected templates of the raw features in the normal scenario. Furthermore, in the stolen scenario, our framework has far better performance than Biohash protected templates of the raw features. In addition, we provided ablation studies to verify the impact of different hyper-parameters, including the length of the embedding layer in the AE ($L_{embedding}$), the length of Biohash templates ($L_{Biohash}$), and the value of α in equation 1 for controlling the contribution of different terms in the loss function. We also evaluate the generalization of our proposed framework on other vascular biometric modalities (i.e., palm and wrist).

Acknowledgments

This research is based upon work supported by the H2020 TReSPAsS-ETN Marie Sklodowska-Curie early training network (grant agreement 860813).

References

[1] M. Lim, A. B. J. Teoh, and J. Kim, "Biometric feature-type transformation: Making templates compatible for secret protection," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 77–87, 2015.

[2] A. Sarkar and B. K. Singh, "A review on performance, security and various biometric template protection schemes for biometric authentication systems," *Multimedia Tools and Applications*, pp. 1–56, 2020.

[3] *ISO/IEC 24745:2011(E) Information technology – Security techniques – Biometric information protection*, International Organization for Standardization International Standard, Jun. 2011.

[4] B. T. Ton and R. N. J. Veldhuis, "A high quality finger vascular pattern dataset collected using a custom designed capturing device," in *Proceedings of the 2013 International Conference on Biometrics (ICB)*, Madrid, Spain, Jun. 2013, pp. 1–5.

[5] 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*. IEEE, 2010, pp. 1269–1272.

[6] 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.

[7] ——, "Extraction of finger-vein patterns using maximum curvature points in image profiles," *IEICE TRANSACTIONS on Information and Systems*, vol. 90, no. 8, pp. 1185–1194, 2007.

[8] M. Sandhya and M. V. Prasad, "Biometric template protection: A systematic literature review of approaches and modalities," in *Biometric Security and Privacy*. Springer, 2017, pp. 323–370.

[9] R. K. Pandey, Y. Zhou, B. U. Kota, and V. Govindaraju, "Learning representations for cryptographic hash based face template protection," in *Deep learning for biometrics*. Springer, 2017, pp. 259–285.

[10] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245–2255, 2004.

[11] V. Krivokuća and S. Marcel, "On the recognition performance of biohash-protected finger vein templates," in *Handbook of Vascular Biometrics*. Springer, Cham, 2020, pp. 465–480.

[12] C. S. Chin, A. T. B. Jin, and D. N. C. Ling, "High security iris verification system based on random secret integration," *Computer Vision and Image Understanding*, vol. 102, no. 2, pp. 169–177, 2006.

[13] A. Goh and D. C. Ngo, "Computation of cryptographic keys from face biometrics," in *IFIP International Conference on Communications and Multimedia Security*. Springer, 2003, pp. 1–13.

[14] T. Connie, A. Teoh, M. Goh, and D. Ngo, "Palmhashing: a novel approach for cancelable biometrics," *Information Processing Letters*, vol. 93, no. 1, pp. 1–5, 2005.

[15] K. Sundararajan and D. L. Woodard, "Deep learning for biometrics: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–34, 2018.

[16] B. Bhanu and A. Kumar, *Deep learning for biometrics*. Springer, 2017.

[17] R. Kabaciński and M. Kowalski, "Vein pattern database and benchmark results," *Electronics Letters*, vol. 47, no. 20, pp. 1127–1128, 2011.

[18] 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 Processing: Machine Vision Applications II*, vol. 7251. International Society for Optics and Photonics, 2009, p. 725111.

[19] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2228–2244, 2011.

[20] J.-L. Starck, J. Fadili, and F. Murtagh, "The undecimated wavelet decomposition and its reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 297–309, 2007.

[21] S.-J. Chuang, "Vein recognition based on minutiae features in the dorsal venous network of the hand," *Signal, Image and Video Processing*, vol. 12, no. 3, pp. 573–581, 2018.

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

[23] C. Xie and A. Kumar, "Finger vein identification using convolutional neural network and supervised discrete hashing," in *Deep Learning for Biometrics*. Springer, 2017, pp. 109–132.

[24] 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.

[25] W. Yang, W. Luo, W. Kang, Z. Huang, and Q. Wu, "Fvras-net: An embedded finger-vein recognition and antispoofing system using a unified cnn," *IEEE Transactions on Instrumentation and Measurement*, 2020.

[26] S. Marcel, M. S. Nixon, and S. Z. Li, *Handbook of biometric anti-spoofing*, 1st ed. Springer, 2014.

[27] Z. Jin, J. Y. Hwang, Y.-L. Lai, S. Kim, and A. B. J. Teoh, "Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 393–407, 2017.

[28] S. Kirchgasser, C. Kauba, Y.-L. Lai, J. Zhe, and A. Uhl, "Finger vein template protection based on alignment-robust feature description and index-of-maximum hashing," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2020.

[29] S. Kirchgasser, Y.-L. Lai, J. Zhe, and A. Uhl, "Finger-vein template protection based on alignment-free hashing," in *Proceedings of the IEEE 10th International Conference on Biometrics: Theory, Applications, and Systems (BTAS2019)*, 2019, pp. 1–9.

[30] W. Yang, J. Hu, and S. Wang, "A finger-vein based cancellable bio-cryptosystem," in *International Conference on Network and System Security*. Springer, 2013, pp. 784–790.

[31] Y. Liu, J. Ling, Z. Liu, J. Shen, and C. Gao, "Finger vein secure biometric template generation based on deep learning," *Soft Computing*, vol. 22, no. 7, pp. 2257–2265, 2018.

[32] W. Yang, S. Wang, J. Hu, G. Zheng, J. Yang, and C. Valli, "Securing deep learning based edge finger vein biometrics with binary decision diagram," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4244–4253, 2019.

[33] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Valli, "A fingerprint and finger-vein based cancelable multi-biometric system," *Pattern Recognition*, vol. 78, pp. 242–251, 2018.
[34] G. Mai, K. Cao, X. Lan, and P. C. Yuen, "Secureface: Face template protection," *IEEE Transactions on Information Forensics and Security*, 2020.
[35] K. Nandakumar and A. K. Jain, "Biometric template protection: Bridging the performance gap between theory and practice," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 88–100, 2015.
[36] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," *Pattern Recognition Letters*, vol. 79, pp. 80–105, 2016.
[37] T. M. Dang, L. Tran, T. D. Nguyen, and D. Choi, "Fehash: Full entropy hash for face template protection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 810–811.
[38] A. Singh, A. Arora, S. H. Patel, G. Jaswal, and A. Nigam, "Fdfnet: A secure cancelable deep finger dorsal template generation network secured via. bio-hashing," in *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*. IEEE, 2019, pp. 1–9.
[39] A. Singh, S. Hasmukh Patel, and A. Nigam, "Cancelable knuckle template generation based on lbp-cnn," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
[40] V. Talreja, M. C. Valenti, and N. M. Nasrabadi, "Multibiometric secure system based on deep learning," in *2017 IEEE Global conference on signal and information processing (globalSIP)*. IEEE, 2017, pp. 298–302.
[41] R. K. Pandey, Y. Zhou, B. U. Kota, and V. Govindaraju, "Deep secure encoding for face template protection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2016, pp. 77–83.
[42] A. Kumar Jindal, S. Chalamala, and S. Kumar Jami, "Face template protection using deep convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 462–470.
[43] K. H. Cheung, A. W.-K. Kong, J. You, D. Zhang *et al.*, "An analysis on invertibility of cancelable biometrics based on biohashing." in *CISST*, vol. 2005. Citeseer, 2005, pp. 40–45.
[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, Jul. 2015, pp. 448–456.
[45] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2015, pp. 815–823.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, California., USA, May 2015.
[47] R. Belguechi, E. Cherrier, and C. Rosenberger, "How to evaluate transformation based cancelable biometric systems?" in *NIST International Biometric Performance Testing Conference (IBPC)*, 2012.
[48] A. Nagar, K. Nandakumar, and A. K. Jain, "Biometric template transformation: a security analysis," in *Media Forensics and Security II*, vol. 7541. International Society for Optics and Photonics, 2010, p. 75410O.
[49] Y. Lee, Y. Chung, and K. Moon, "Inverse operation and preimage attack on biohashing," in *IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications*. IEEE, 2009, pp. 92–97.
[50] X. Dong, Z. Jin, and A. T. B. Jin, "A genetic algorithm enabled similarity-based attack on cancellable biometrics," in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 2019, pp. 1–8.
[51] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel, "Bob: a free signal processing and machine learning toolbox for researchers," in *Proceedings of the 20th ACM Conference on Multimedia Systems (ACMMM)*, Oct. 2012.
[52] A. Anjos, M. Günther, T. de Freitas Pereira, P. Korshunov, A. Mohammadi, and S. Marcel, "Continuously reproducing toolchains in pattern recognition and machine learning experiments," in *Proceedings of the International Conference on Machine Learning (ICML)*, Aug. 2017.

Hatef Otroshi Shahreza is a Ph.D. student at École Polytechnique Fédérale de Lausanne (EPFL) and a Research Assistant at the Biometrics Security and Privacy Group of Idiap Research Institute (Switzerland) where he received H2020 Marie Sklodowska-Curie fellowship (TReSPAsS-ETN) for his doctoral program. Prior to his Ph.D., Hatef received his BSc. (Hons.) and MSc. in Electrical Engineering from University of Kashan and Sharif University of Technology (Iran) in 2016 and 2018, respectively. His research interests include but are not limited to deep learning, machine learning, computer vision, biometrics, and biometric template protection.

Sébastien Marcel heads the Biometrics Security and Privacy group at Idiap Research Institute (Switzerland) and conducts research on face recognition, speaker recognition, vein recognition, attack detection (presentation attacks, morphing attacks, deepfakes) and template protection. He received his Ph.D. degree in signal processing from Université de Rennes I in France (2000) at CNET, the research center of France Telecom (now Orange Labs). He is a lecturer at the École Polytechnique Fédérale de Lausanne and the University of Lausanne. He is also the Director of the Swiss Center for Biometrics Research and Testing, which conducts certifications of biometric products.