SoftmaxOut Transformation-Permutation Network for Facial Template Protection

Hakyoung Lee, Cheng Yaw Low and Andrew Beng Jin Teoh
School of Electrical and Electronic Engineering
College of Engineering, Yonsei University
Seoul, Korea
{discone9410, chengyawlow, bjteoh}@yonsei.ac.kr

***Abstract*—In this paper, we propose a data-driven cancellable biometrics scheme, referred to as SoftmaxOut Transformation-Permutation Network (SOTPN). The SOTPN is a neural version of Random Permutation Maxout (RPM) transform, which was introduced for facial template protection. We present a specialized SoftmaxOut layer integrated with the permutable MaxOut units and the parameterized softmax function to approximate the non-differentiable permutation and the winner-takes-all operations in the RPM transform. On top of that, a novel pairwise ArcFace loss and a code balancing loss are also formulated to ensure that the SOTPN-transformed facial template is cancellable, discriminative, high entropy and free from quantization errors when coupled with the SoftmaxOut layer. The proposed SOTPN is evaluated on three face datasets, namely LFW, YouTube Face and Facescrub, and our experimental results disclosed that the SOTPN outperforms the RPM transform significantly.**

Keywords— Template Protection, Cancellable Biometrics, Face Recognition, Security.

I. INTRODUCTION

Recently, facilities and computer systems rely on biometrics of individuals for physical and logical access control. Although biometric systems offer a number of advantages, e.g., increased public acceptance, remarkable performance and reliability gain, etc., they are not without controversy. Unlike password, badges, or token which are typically re-issuable in an effortless manner, biometrics are uniquely associated to only an individual and thus are of irreplaceable once compromised. In addition, an identical biometric template could alternatively be stored or shared across multiple applications, and this permits an adversary to launch the cross-matching attack. Therefore, a stringent biometric template protection countermeasure against security threats is of essential.

Cancellable biometrics is one of the widely adopted template protection approaches to address the above-mentioned concerns [1]. In principle, it refers to a non-invertible transformation that converts the original template to the protected instance by means of a user-specific parameter. If the protected template is stolen, it can simply be revoked and replaced by changing the parameter. In practice, a cancellable biometrics scheme is outlined based on the following four criteria [2]:

- *Non-invertibility or Irreversibility*: The recovery attempt for the original template from a protected instance should be impossible or computationally hard.
- *Revocability or Renewability*: A protected template should easily be re-issued whenever needed.
- *Non-linkability or Unlinkability*: Two or more protected templates generated for the same person must be untraceable to any linkage to prevent cross-matching of those templates.
- *Performance Preservation*: The accuracy performance of the protected systems must be preserved with respect to its original counterpart.

Random Permutation MaxOut (RPM) was introduced in [3] as a means of cancellable biometrics for face template protection. The RPM transforms a pre-extracted facial feature vector into a discrete code by localizing the maximal entries of the truncated and permutated original template (refer to section III). The RPM was demonstrated satisfying the criteria of template protection. However, as a data-agnostic scheme, the accuracy performance for the RPM-transformed code is substantiated an unfavorable depreciation compared to its original (unprotected) counterpart. This problem is exaggerated for unconstrained face biometrics due to severe intra-class variability. Therefore, the vanilla RPM transform is unlikely to attain decent accuracy performance.

In this paper, we re-formulate RPM as a data-driven cancellable biometric scheme, known as SoftmaxOut Transformation-Permutation Network (SOTPN). The SOTPN is a fully-connected neural network, where it receives two inputs – a facial template (unprotected) in the continuous-valued vector form and a user-specific parameter, to generate a discrete vector (protected) in return. We propose a specialized layer, termed as *permutable SoftmaxOut* layer hereafter, to assemble the SOTPN. To be more specific, we equip the permutable SoftmaxOut layer with permutable MaxOut units [4] and a parameterized softmax function to approximate the non-differentiable permutation and the winner-takes-all operations in the RPM transform. Aside from that, the parameterized softmax function is also beneficial to minimize the quantization errors incurred by the SoftmaxOut approximation.

Since SOTPN serves to render the protected templates from the network output layer directly via learning, the conventional classification-based loss such as softmax loss is inappropriate. We hence re-interpret the cancellable transformation problem as an embedding learning (also known as metric learning) problem [5]. Accordingly, a pairwise distance based loss is to be applied to optimize the margin between intra- and inter-class distances. In this paper, we propose a novel pairwise-based loss function,

dubbed *pair-wise ArcFace loss*. This is inspired by the ArcFace loss [6], which is a variant of the softmax loss for classification-based networks essentially. In addition, to maximize the entropy of the SOTPN-transformed codes, we also devise in this paper a code balancing loss.

In general, the SOTPN is applicable to any continuous-valued biometric template as a means of template protection. We comprehensively evaluate the aptitude of SOTPN based on the facial templates extracted from the pre-learned ArcFace network [6] on three unconstrained face datasets, namely LFW, YouTube Face and FaceScrub. Our experimental results demonstrate that the SOTPN outperforms the RPM significantly, and moreover satisfying the cancellable biometrics criteria.

II. Related Work

In this section, we provide a brief review on the relevant and recent facial template protection literature.

A. Salting-Based Cancellable Face Template Protection

We highlight specifically the salting-based cancellable biometric schemes of which the RPM transform and SOTPN belong to. On top of biometrics, the salting-based approaches are of two-factor authentication systems demanding a user-specific parameter, either a password or a token, as an additional input. The key advantage is that the protected templates have very low or even zero correlation amongst themselves. This is attributed to the external parameter, which is particularly crucial to achieve the unlinkability and revocability requirements. If the parameter is leaked, the protected templates can be vulnerable to inversion attempts. However, the parameter can be secured with additional computation cost [7][8][9].

The Biohashing [10] is the first instance of the salting-based construct utilizing a user-specific password or token generated random matrix **R** as the second input. Despite of satisfying both unlinkability and revocability, it suffers from huge performance deterioration when **R** is stolen. To confront with this problem, a wide range of countermeasure are proposed, e.g., orthonormal random projection [11], augmented random projections [12] and sparse random projections [13] are proposed. The authors reveal both in theory and empirically that the accuracy performance of the stolen **R** scenario appears similar to the before-transformed counterparts.

Recently, Jin et. al [14] put forward the Index-of-Max (IoM) hashing for biometric template protection. With IoM hashing, the biometric template is transformed into the indices of the first ranked features chosen from a number of random projections of the original features. The IoM hashing is indeed a generalization of Biohashing. The RPM transform shares the similarity with the IoM hashing where both are motivated by the winner-takes-all hashing, which is devised for data retrieval [15]. However, the RPM transform relies on user-specific permutation, whereas the IoM hashing applies random projection to re-issue the revoked templates.

B. Deep Learning-Based Face Template Protection

Deep learning-based facial template protection is gaining enormous attention very recently [16][17][18][19]. As a whole, these works pursue the same pipeline, where the facial template is bound to a user-specific binary bit string (an authenticator) via a deep convolutional neural networks (CNN). The authenticator is typically protected using a cryptographic hash function, e.g., SHA-3, and is compiled into a database. During authentication, the authenticator is generated by the CNN based upon the query facial image, and the output is subsequently hashed by SHA-3 and matched with the templates stored in the database. An exact match is demanded to render a final decision. The authenticator can be revoked and replaced whenever needed.

This line of work is insecure in practice. To be more specific, an authenticator is needed to be assigned to each enrolled subject, and the samples of all subjects with the corresponding assigned authenticator are utilized to train the CNN. It is noteworthy that these CNNs work well if no new subjects are enrolled after being deployed. However, when a new subject has to be enrolled, or a particular authenticator has to be renewed, the entire CNN has to be retrained from scratch with respect to the facial images and the authenticators of all the enrolled and the new subjects. This poses a risk where both facial images and the authenticators may be unveiled unintentionally. Moreover, the adversary may learn some useful information about the enrolled subjects based on the network's parameters.

The closest scheme to SOTPN is Deep Table-based Hashing (DTH) reported in [20]. The DTH employs an end-to-end trained CNN to generate a cancellable template in the binary string form at the network output layer from the raw facial images. A new template is re-issued by re-shuffling the hash table associated to the CNN. While the DTH is shown offering high accuracy, it is unrealistic and insecure as the entire network is trained directly from the enrollees' face images [21].

III. Preliminary

A. Random Permutation Maxout Transform

In a nutshell, the Random Permutation Maxout (RPM) [3] transform maps a continuous-valued facial feature vector onto a discrete code representation. The algorithm of RPM transform is summarized as follows:

1) Generate a user-specific permutation matrix $\boldsymbol{\mathcal{P}}$ that stacked with m layers, i.e., $\boldsymbol{\mathcal{P}} \in \{0,1\}^{d\times d\times m}$, where m indicates the length of the RPM-transformed code, and d is the dimension of the input facial vector $\mathbf{x}$.
2) Given $\boldsymbol{\mathcal{P}}$, permute the facial feature vector $\mathbf{x} \in \mathbb{R}^d$ via $\mathbf{W} = \mathbf{x}\boldsymbol{\mathcal{P}} \in \mathbb{R}^{d\times m}$.
3) Truncate the last $d - q$ columns from $\mathbf{W}$ to yield $\mathbf{Y} \in \mathbb{R}^{m\times q}$.
4) Identify the maximal entry for each row of $\mathbf{Y}$, and the indices of all maximal entries are summarized onto a *integer-valued* RPM-transformed vector $\mathbf{u} \in [1\ q]^m$, where $\mathbf{u}$ is in fact a cancellable template that is renewable by twisting $\boldsymbol{\mathcal{P}}$.

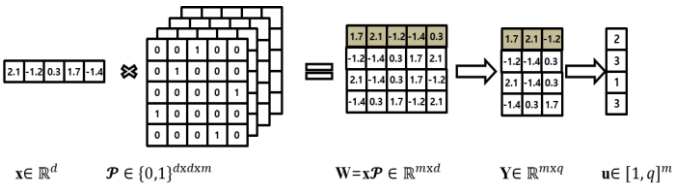

Fig. 1. An illustration of the RPM transform, provided d=5, q=3 and m=4.

Figure 1 depicts the progression of the RPM transform. Note $\{m, q\}$ are the parameter set to be determined empirically based upon different datasets and applications.

B. *ArcFace Loss*

Since the pairwise loss function for SOTPN is inspired from the ArcFace loss [6], we brief the ArcFace loss in this section.

Classification-based CNNs are affixed with the softmax loss for optimizing the inter-class separation explicitly, but not the intra-class compactness. To address this restriction, the softmax loss is modified to that of angular softmax loss, where the weight vectors of each class are normalized and a multiplicative margin is introduced to force the classification boundary closer to that specific weight vector [22]. Optimization of the multiplicative angular margin loss, however, is a non-trivial problem.

Therefore, an additive angular margin loss emerge to resolve this problem [23]. Apart from that, the additive angular margin loss also leads the learned features to be potentially separable for a larger angular distance. The ArcFace loss presented in [6] is an instance of additive margin loss offering a clearer geometrical interpretation and promising performance evaluated on a series of face recognition benchmarks.

IV. Proposed Method

Our proposed scheme for facial template protection consists of two parts as shown in Figure 2. The first part is a facial feature extractor, which serves to yield the face template. In this paper, the pre-trained ArcFace network is employed for this purpose. In general, the feature extractor can be any more powerful CNNs as long as the face template is in the form of continuous-values vector.

On the contrary, the second part is the proposed SoftmaxOut Transformation-Permutation Network (SOTPN) obligated to map a face template onto its protected form. Architecturally, the SOTPM is a two-hidden-layer network followed by a dedicated permutable SoftmaxOut layer and an output layer to realize the progression of the RPM transform delineated in Section III(A). Each network component constructing SOTPN will be detailed in the following sections.

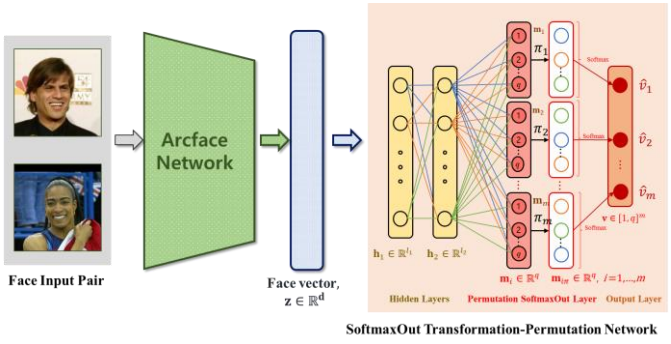

Fig. 2. Overview of SoftmaxOut Transformation-Permutation Network.

A. *Arcface Network as Feature Extractor*

For feature extraction, we utilize the ArcFace network pre-learned from 3.1M images with 8.1K identities approximately from MS-Celeb-1M. The backbone of the ArcFace network is based on the ResNet-50 containing 49 convolutional layers followed by a single fully connected layer. For every face image of size 112×112 pixels, it yields a 512-dimension feature vector, denoted by $\mathbf{z} \in \mathbb{R}^{512}$.

B. *SoftmaxOut Transformation-Permutation Network*

The SOTPN is architecturally a shallow network assembled with two fully-connected hidden layers – the first hidden layer $\mathbf{h}_1 \in \mathbb{R}^{l_1}$ is configured with l_1 neurons, and each appended with a ReLu activation function; on the other hand, the second hidden layer $\mathbf{h}_2 \in \mathbb{R}^{l_2}$ consists of l_2 neurons connected to the proposed permutable SoftmaxOut layer. The SoftmaxOut layer is set with m maxout units $\mathbf{m}_i \in \mathbb{R}^q$ composing of q linear neurons, along with the modified softmax function. Each maxout unit is indeed a function returning the index of the maximal entry (out of the q neurons). The ultimate output layer produces m transform codes v_i forming a protected template $\mathbf{v} \in [1\ q]^m$.

Recall the key ingredient of the RPM transform (Step 4, Section III(A)) is to elicit the index value of the maximum entry (winner-takes-all) of a q-dimensional permuted vector. In this context, this process is equivalent to taking the index value, v_i from a maxout unit as follows:

$$v_i = \arg\max_q \mathbf{m}_i \in \{1, \dots, q\},\ i = 1, \dots, m \tag{1}$$

Nonetheless, (1) is non-differentiable and thus non-trainable with backpropagation. To this end, (1) can be approximated with the following function:

$$v_i \approx \tilde{v}_i = \sum_{j=1}^{q} j\ \sigma_\beta(\mathbf{m}_i)\ \in \{1, \dots, q\} \tag{2}$$

where $\sigma_\beta()$ is the softmax function parameterized with $\beta > 1$:

$$\sigma_\beta(v) = \frac{\mathrm{e}^{\beta v}}{\sum_{i=1}^{q} \mathrm{e}^{\beta v_\mathrm{i}}} \tag{3}$$

Unlike the conventional softmax function, the modified softmax function is expressed with a scalar factor $\beta > 1$ forcing the network output towards 0 or 1, and thereby facilitating the SOTPN learns an integer code.

As a numerical example, suppose q = 4 and $\mathbf{m}$ = [0.19, 0.74, 0.24, 0.92], by applying (3) with $\beta = 1$, $\sigma_\beta(\mathbf{m})$ = [0.17, 0.30, 0.18, 0.35] and hence $\tilde{v} = 1(0.17) + 2(0.30) + 3(0.18) + 4(0.35) = 2.71$. This is incorrect as actual v should be equal to 4. This issue becomes more severe for large q. On the other hand, if $\beta \gg 1$, $\sigma_\beta(\mathbf{m})$ = [0, 0, 0, 1], hence $\tilde{v} = v = 4$. Therefore, the discretization effect induced by β in (3) is essential to minimize the peril of similar (dissimilar) inputs earning different (same) integer values by a narrow margin. This can effectively minimize the quantization errors that caused by the SoftmaxOut approximation.

To enable the permutation operation in the RPM transform, we apply the permutation function π_i that seeded by user-specific s_π (the second input factor) to $\mathbf{m}_i$, (2) is then modified to as follows:

$$v_i = \sum_{j=1}^{q} j\ \sigma_\beta(\pi_i(\mathbf{m}_i; s_\pi))\ \in \{1, \dots, q\} \tag{4}$$

By doing so, the SOTPN is capable of generating different protected templates $\mathbf{v} \in [1, q]^m$ at the output layer, given with different s_π. Meanwhile, the total number of cancellable templates that can be generated from a facial template would be $q!m$. The analysis for unlinkability and revocability of the SOTPN attributed to permutation is presented in section VI.

C. Loss Functions

1. Pairwise Arcface Loss

Let $\{\mathbf{z}_j \in \mathbb{R}^d \mid j = 1, \ldots, N\}$ be the N face feature vectors extracted from the pre-trained ArcFace network. The feature vector pairs $\mathbf{z}_i$ and $\mathbf{z}_j$ are associated with similarity labels s_{ij} where s_{ij} = 1 implies $\mathbf{z}_i$ and $\mathbf{z}_j$ are from the same subject (positive pair) and s_{ij} = 0 indicates $\mathbf{z}_i$ and $\mathbf{z}_j$ are from the different subjects (negative pair). The goal is to learn a non-linear transformation function f: $\mathbf{z} \rightarrow \mathbf{v} \in [1, q]^m$ based upon a loss function to transform each $\mathbf{z}$ to the protected template $\mathbf{v}$. The loss function is to make the similarity of $\mathbf{v}_i$ and $\mathbf{v}_j$ pair be high if they are positive pairs and make the dissimilarity greater than a margin for negative pairs.

Despite the ArcFace loss discussed in Section III(B) aims to achieve this goal, it is not applicable to SOTPN. This is due to the reason that the ArcFace loss is a classification-based loss that requires an explicit classification layer, of which its capacity and computational cost increases linearly with respect to the number of subjects [6]. In contrast, the output layer of SOTPN produces cancellable templates and its capacity is instead associated to the template size m. To this end, we modify the ArcFace loss to fit our goal as follows:

$$PA_{ij} = -log\left[s_{ij}\frac{e^{\gamma\cos(\theta+\alpha)}}{e^{\gamma\cos(\theta+\alpha)} + e^{\gamma\sin\theta}}\right] - log\left[(1-s_{ij})\frac{e^{\gamma\sin(\theta+\alpha)}}{e^{\gamma\sin(\theta+\alpha)} + e^{\gamma\cos\theta}}\right] \quad (5)$$

Note that γ is a scaling factor, α is the angular margin, and θ is the angle between $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{v}}_j$ or $\theta = \cos^{-1}(\hat{\mathbf{v}}_i^T \hat{\mathbf{v}}_j)$, given that $\hat{\mathbf{v}}$ is the L_2 normalized vector to be re-scaled with respect to γ.

The normalization on $\mathbf{v}_i$ and $\mathbf{v}_j$ makes the similarity measure only relies on the angle between the two vectors. The learned cancellable vectors are thus distributed on a hypersphere with a radius of γ. As the learned vectors are distributed on the hypersphere, an additive angular margin penalty α between $\hat{\mathbf{v}}_i$ and $\hat{\mathbf{v}}_j$ is introduced to simultaneously enhance the intra-class compactness and the inter-class separation.

As depicted in Fig. 3, for positive pairs (s_{ij} = 1), the loss will be 0 only when $\theta < \frac{1}{2}(\frac{\pi}{2} - \alpha)$, which indicates two vectors are similar. When θ is large, the loss value increases, and the network parameters will be updated by minimizing the loss. On the other hand, for negative pairs (s_{ij}=0), PA loss operates in the reverse way. The loss value approaches 0 when $\theta \rightarrow \frac{1}{2}(\frac{\pi}{2} - \alpha)$. The network will be updated accordingly by minimizing the loss. Note excessive large margin α may harm the inter-class separation.

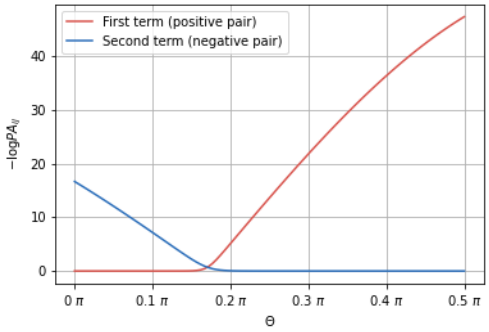

Fig. 3. Pairwise ArcFace loss versus cosine distance (θ) with α=0.5 and γ=32.

2. Code Balancing Loss

This section considers the balance property of the transform code v_i, which is accomplishable by imposing the occurrence probability be $1/q$ for each code. To do so, we let $\{v_*^b\}_{b=1}^B$, *={i, j} be the B batch samples of the transform code from the training set and suppose they form a discrete probability distribution over {1,…,q}. Ideally, we suppose that the occurrence probability for each code is uniform and hence achieving the maximal entropy. This is crucial to prevent underutilization of the information capacity of the SOTPN code. Therefore, each v_i is to be fired 100%/q of the time by minimizing the following code balancing (CB) loss:

$$CB_{ij} = \sum_{b=1}^{B}\sum_{k=1}^{m}\{\mid mean(v_i^{bk}) - \frac{q+1}{2}\mid + \mid mean(v_j^{bk}) - \frac{q+1}{2}\mid\} \quad (6)$$

where *mean*() is the average operator.

Figure 4 illustrates the SOTPN code distributions before and after applying the CB loss with q = 32 generated using the LFW dataset (See Section V(A)). We notice from Figure 3(a) that the SOTPN codes are left-skewed, falling within the range of 1 to 10 only, without CB loss. However, with the proposed CB loss applied, the code values are stretched the complete range of 1 to 32 in a relatively uniform manner.

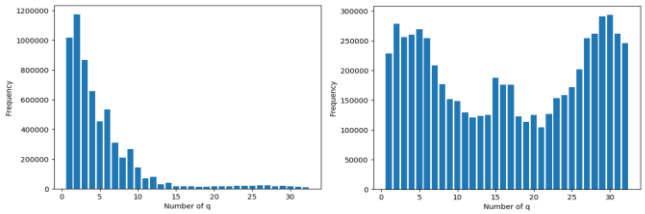

Fig. 4. (a) SOTPN code distribution without applying code balancing loss; (b) SOTPN code distribution after applying code balancing loss where q=32.

3. Overall Loss

In a nutshell, the overall loss function $\mathcal{L}_{ij}$ for optimizing the SOTPN is expressed as:

$$\min \mathcal{L}_{ij} = PA_{ij} + \lambda CB_{ij} \quad (7)$$

where λ is a hyperparameter that adjusts the balance between the PA loss and the CB loss.

V. Experiments And Analysis

A. Datasets

Our experiments are conducted on three unconstrained face datasets, namely LFW [24], YouTube [25] and FaceScrub [26].

1) LFW (labeled Faces in the wild) is a face dataset collected from the Internet. It is a composition of 13,233 images with 5,749 identities.

2) YTF (YouTube Faces) is a repository with 3,245 videos of 1,595 subjects downloaded from YouTube.

3) FS (Facescrub) dataset contains 106,863 face images of 530 celebrities with approximately 200 images per person.

Following the LFW standard evaluation protocol [24], we apply the pre-determined 3,000 matched pairs and 3,000 non-matched pairs for verification tasks. Likewise, a summation of 6,000 pairs (3,000 for each matched and unmatched pairs) are generated for evaluation using YTF and FS.

For SOTPN training, the first and the second hidden layers of SOTPN are initialized with 1,024 neurons, and the network is trained using the Adam optimizer. The mini batch size is pre-fixed at 256, and other parameters, including the coefficient of CB loss, i.e., λ in (7), is set to 0.01, and β in (3) is set to 9. For PA loss (5), we set α to 0.5 and γ to 32 for all experiments. Our experiments are carried out by using Tensorflow Python library.

Note that the datasets designated for both training and testing are of independent. For example, we employ FS for training, and another dataset, e.g., LFW, is applied for testing. This is crucial particularly for biometric template protection for two important reasons. First, the network should not be re-trained whenever a new subject is enrolled. Second, the network parameters should not be exposed with the enrolled subjects' identity information [21].

We apply Cosine distance as a similarity metric. We report the overall verification performance in terms of Equal Error Rate (EER) and Receiving Operating Characteristic (ROC). For a fair comparison on accuracy performance, we do not enable permutation in the SOTPN.

B. Parameter Analysis

This section analyzes the SOTPN performance with respect to different settings of m and q. By fixing q=32, we discern from Table I that EER diminishes when m is increased. Meanwhile, by fixing m=512, Table II discloses that q is of dataset-specific.

Table I. EER (%) of SOTPN with various m by fixing q =32

Training/Testing	m				
	32	64	128	256	512
FS/LFW	9.47	6.20	5.00	4.23	**3.97**
FS/YTF	17.44	12.80	11.40	9.72	**9.12**
YTF/FS	10.06	5.60	4.04	3.30	**3.22**

We therefore fix q = 64, 32, 16 for FS/LFW and FS/YTF, respectively, in the experiments discussed in the following section.

Table II. EER (%) of SOTPN with various q by fixing m =512.

Training/Testing	q				
	8	16	32	64	128
FS/LFW	3.90	3.83	3.97	**3.73**	4.77
FS/YTF	11.60	11.76	**9.12**	9.86	9.50
YTF/FS	3.12	**3.00**	3.22	3.24	3.48

C. Performance Comparison

An extensive performance comparison between the SOTPN-transformed template, the unprotected ArcFace template and the RPM-transformed template is presented in this section. On top of that, we compare also the SOTPN performance to the SOTPN counterpart trained from the conventional pairwise cross entropy (CP) loss introduced by [27] as follows:

$$CP_{ij} = \sum_{s_{ij} \in S} \left(\log\left(1 + \exp\left(\hat{\mathbf{v}}_i^T \hat{\mathbf{v}}_j\right)\right) - s_{ij} \hat{\mathbf{v}}_i^T \hat{\mathbf{v}}_j\right) \quad (8)$$

Our first experiment adopts FS for training and LFW for testing, and the another utilizes FS for training and YTF for testing. The m is varied from 32 to 512, whereas the best q of each method as shown in Table II is adopted. We summarize our empirical findings comparing all four templates in the terms of EER and ROC curves in Figure 5 and 6.

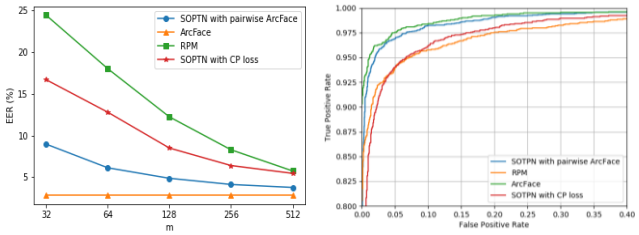

Fig. 5. (a) EER vs m; (b) ROC curves (trained with FS and tested with LFW)

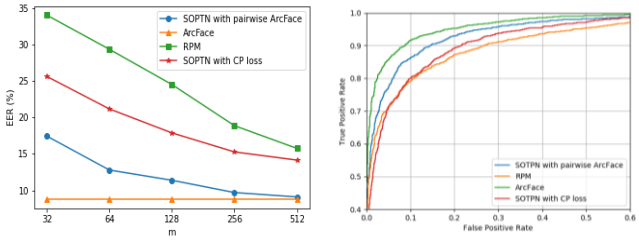

Fig. 6. (a) EER vs m; (b) ROC curves (trained with FS and tested with YTF)

We observe that the unprotected ArcFace template performs the best, whilst the SOTPN, as a data-driven template protection technique, notably outperforms the data-agnostic counterpart, i.e. RPM transform, for all m. In the meantime, the performance of SOTPN is discerned comparable to that of unprotected ArcFace template for a larger m, specifically m = 512. As for the SOTPN learned with CP loss (8), it is outperformed by the unprotected ArcFace template, i.e., its direct counterpart trained with PA loss

(5)), although it is superior to RPM. This attests the efficacy of our proposed loss function.

It is remarked that the comparison with other state of the arts is inappropriate. As presented in Section II(B), the deep learning-based facial template protection solutions, e.g., [16], [17], [18], and [19], pursue the authenticator-biometric binding approach, which require exact matching and it is different from us in terms of evaluation. Furthermore, despite [20] is similar to us, it is an end-to-end network, and the datasets for training and verification are not independent, which makes it unfair for comparison.

D. Ablation study

This section summarizes our ablation study for the proposed SOTPN trained with respect to the FS dataset for $m = 512$ and $q = 32$, and our evaluation is performed on the LFW dataset.

From Table III, we observe that SOTPN performs the worst at EER=5.7% without applying CB loss and without considering the quantization error issue (setting $\beta = 1$ in (3)). In contrast to that, a significant improvement from 5.7% to 4.07% is observed when β is set to 9, despite CB loss is absent. On the other hand, we notice the CB loss improves the SOTPN performance from 5.70% to 4.50%, without considering the quantization issue. To sum up, the quantization error affects the accuracy performance more remarkably than the CB loss. However, the code balancing is of essential from the security perspective as it is beneficial to improve the SOTPN code's entropy.

TABLE III. ABLATION STUDY FOR COTPN IN TERMS OF EER%.

SOTPN Configuration	EER (%)
PA loss with $\beta = 9$ + CB Loss	**3.53**
PA loss with $\beta = 9$, without CB loss	4.07
PA loss with $\beta = 1$ + CB loss	4.50
PA loss with $\beta = 1$, without CB loss	5.70

VI. UNLINKABILITY AND REVOCABILITY ANALYSIS

Although SOTPN resembles RPM transform in several aspects but feature permutation of them is slightly different where the former takes place in the maxout units while the latter occurs at original facial template. Despite unlinkability and revocability of the RPM transform has been analyzed in [3], we re-evaluate the SOTPN due to different permutation mechanism.

A. Unlinkability Evaluation

The unlinkability demands that the transform templates are not differentiable whether they are from same subject. This is to prevent against matching across different applications (cross matching attack). Under this attack, an adversary is assumed familiar with SOTPN and holds protected templates of different applications. The adversary can exploit the matching score distributions of the protected templates to learn the template is from the same person.

We evaluate the unlinkability of SOTPN based upon the method and protocol outlined in [28]. The experiments are conducted under the setting where the network trained by FS and tested by LFW, with $m = 512$ and $q = 32$. Suppose we call the matching score of a pair of protected templates from the same subject as *mated score*, while the matching score of a pair from different subjects as *non-matched score*. In accordance with [28], two measures for linkability are defined, namely local measure $D_{\leftrightarrow}(s) \in [0,1]$, and global measure $D_{\leftrightarrow}^{sys}$. The $D_{\leftrightarrow}(s)$ evaluates the linkability for each specific linkage score s at score-wise level. Given a score s, $D_{\leftrightarrow}(s) = 1$ indicates the adversary can decide which protected template is from the same person with almost all certainty and otherwise for $D_{\leftrightarrow}(s_k) = 0$. $D_{\leftrightarrow}^{sys} \in [0,1]$ evaluates the unlinkability of the whole system and can be used as a benchmark for different systems independently of the score. $D_{\leftrightarrow}^{sys} = 1$ indicates the system is fully linkable for all scores of the mated subjects and otherwise for $D_{\leftrightarrow}^{sys} = 0$.

We generate 10 transform templates of the same identity with different permutation seeds. In subsequent to that, the mated scores and non-mated scores are computed among those templates and their distributions are illustrated in Figure 7. From figure 7, it is disclosed that SOTPN achieves good unlinkability for $D_{\leftrightarrow}^{sys} = 0.06$.

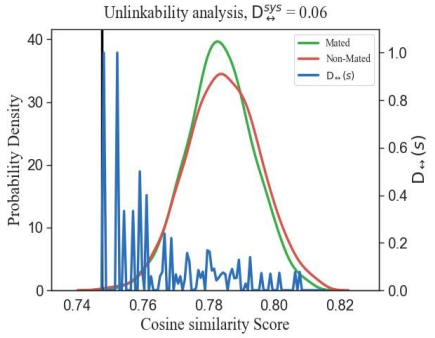

Fig. 7. Unlinkability analysis of SOTPN templates

B. Revocability Evaluation

Revocability refers to the capability of re-issuing a compromised template. It should be computationally simple to generate numerous protected templates whenever needed.

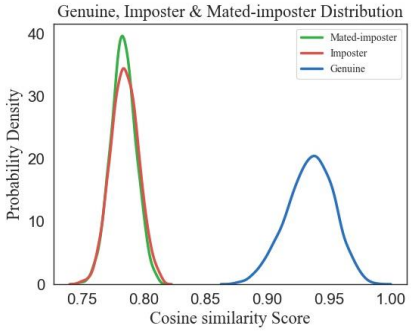

Fig. 8. Revocability analysis of SOTPN templates

In order to evaluate the revocability of SOTPN, a user-specific key scenario is considered. This implies that each subject should privately possess his/her own permutation seed. The revocability can be examined by observing the distributions of mated-imposter scores, genuine scores, and imposter scores. In practice, the genuine/imposters scores can be calculated by

matching the transform templates generated from same/different subjects, respectively. The mated-imposter scores, on the other hand, can be generated by matching the transform templates of the same subject but different permutation seeds. Note that our setting of revocability evaluation is similar with the unlinkability evaluation described in the preceding section.

As shown in Fig. 8, the distribution of imposter and mated-imposter scores are overlapped largely. This indicates that there is no difference between the transform templates of the same subject and different subjects with varying permutation seeds.

VII. Non-invertibility Analysis

In our context, irreversibility measures the computational hardness in restoring the ArcFace features $\mathbf{z} \in \mathbb{R}^{512}$ from the SOTPN vector $\mathbf{v} \in [1, q]^m$, with and without accessing to the knowledge of user-specific permutation seed s_π.

For this analysis, the RPM transform and the SOTPN share the identical traits, where both are of two-factor authentication schemes leveraging permutation and winner-takes-all strategy to define **v**. Therefore, revelation of **v** and/or s_π are highly unlikely to recover **z** as no direct link exists between **z** and **v** due to the winner-takes-all characteristic. Furthermore, each SOTPN code v_i is independent and of high entropy, thereby heightening the difficulty of inversion. For detailed analysis, readers are referred to [3].

False Accept Attack (Dictionary Attack) can also be launched by trying to find an approximate instance of **z**, denoted as **z**' ($\approx$ **z**) satisfying $f(\mathbf{z}; s_\pi) = f(\mathbf{z}'; s_\pi)$ where $f()$ is the SOTPN transformation. Generally, they use a huge set of **z** to exploit the false accept rate of the system or even adopt a more advanced solver such as Genetic Algorithm based Similarity-based Attack Framework (GASAF) [29] to estimate **z**'. However, these attacks would not be succeeded if s_π is secured. This can actually be done in practice with the techniques suggested by [7][8][9].

VIII. Conclusion

In this paper, a neural version of Random Permutation Maxout (RPM) transform, namely SoftmaxOut Transformation-Permutation Network (SOTPN), is outlined for facial template protection. In order to realize the non-differentiable winner-takes-all strategy and the permutation operations in the RPM transform, we introduce the permutable SoftmaxOut layer that integrates maxout units and a parameterized softmax function. We put forward pairwise ArcFace loss and code balancing loss to ensure the transform codes are of discriminative and high entropy. Our empirical results showed that the proposed SOTPN prevails over the RPM transform significantly and furthermore satisfying the pre-requisite cancellable biometrics design criteria. As for future work, we will improve the scheme by hiding the second input factor with biometric cryptosystem techniques [9] so that security of the SOTPN can be further enhanced.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NO. NRF-2019R1A2C1003306)

References

[1] Patel, Vishal M., Nalini K. Ratha, and Rama Chellappa. "Cancelable biometrics: A review." IEEE Signal Processing Magazine 32.5 (2015): 54-65.

[2] Jain, Anil K., Karthik Nandakumar, and Abhishek Nagar. "Biometric template security." EURASIP Journal on advances in signal processing 2008 (2008): 1-17.

[3] Teoh, Andrew Beng Jin, Sejung Cho, and Jihyeon Kim. "Random permutation Maxout transform for cancellable facial template protection." Multimedia Tools and Applications 77.21 (2018): 27733-27759.

[4] A. A. Kobaisi and P. Wocjan, "Supervised Max Hashing for similarity Image Retrieval," 17th IEEE International Conference on Machine Learning and Applications (2018) pp. 359-365

[5] Kulis, Brian. "Metric learning: A survey." Foundations and trends in machine learning 5.4 (2012): 287-364.

[6] Deng, Jiankang, et al. "Arcface: Additive angular margin loss for deep face recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[7] Takahashi, Kenta, and Shinji Hirata. "Parameter management schemes for cancelable biometrics." 2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM). IEEE, 2011.

[8] Kim, Jihyeon, and Andrew Beng Jin Teoh. "One-factor cancellable biometrics based on indexing-first-order hashing for fingerprint authentication." 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.

[9] Lai, Yenlung, et al. "Secure Secret Sharing Enabled b-band Mini Vaults Bio-Cryptosystem for Vectorial Biometrics." IEEE Transactions on Dependable and Secure Computing (2018).

[10] Teoh, Andrew BJ, Alwyn Goh, and David CL Ngo. "Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs." IEEE transactions on pattern analysis and machine intelligence 28.12 (2006): 1892-1901.

[11] Teoh, Andrew Beng Jin, and Chong Tze Yuang. "Cancelable biometrics realization with multispace random projections." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37.5 (2007): 1096-1106.

[12] Wang, Yongjin, and Dimitrios Hatzinakos. "On random transformations for changeable face verification." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 41.3 (2010): 840-854.

[13] Oh, Beom-Seok, et al. "Extraction and fusion of partial face features for cancelable identity verification." Pattern recognition 45.9 (2012): 3288-3303.

[14] Jin, Zhe, et al. "Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing." IEEE Transactions on Information Forensics and Security 13.2 (2017): 393-407.

[15] Yagnik, Jay, et al. "The power of comparative reasoning." 2011 International Conference on Computer Vision. IEEE, 2011.

[16] Kumar Jindal, Arun, Srinivas Chalamala, and Santosh Kumar Jami. "Face template protection using deep convolutional neural network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.

[17] Talreja, Veeru, Matthew C. Valenti, and Nasser M. Nasrabadi. "Zero-Shot Deep Hashing and Neural Network Based Error Correction for Face Template Protection." arXiv preprint arXiv:1908.02706 (2019).

[18] Chen, Lingying, et al. "Face template protection using deep LDPC codes learning." IET Biometrics 8.3 (2018): 190-197.

[19] Dang, Thao M., et al. "FEHash: Full Entropy Hash for Face Template Protection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.

[20] Jang, Young Kyun, and Nam Ik Cho. "Deep Face Image Retrieval for Cancelable Biometric Authentication." 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2019.

[21] Osadchy, Margarita, and Orr Dunkelman. "It is All in the System's Parameters: Privacy and Security Issues in Transforming Biometric Raw Data into Binary Strings." IEEE Transactions on Dependable and Secure Computing 16.5 (2018): 796-804.

[22] Liu, Weiyang, et al. "Sphereface: Deep hypersphere embedding for face recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[23] Wang, Feng, et al. "Additive margin softmax for face verification." IEEE Signal Processing Letters 25.7 (2018): 926-930.

[24] Huang, Gary B., et al. "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments." 2008.

[25] Wolf, Lior, Tal Hassner, and Itay Maoz. "Face recognition in unconstrained videos with matched background similarity." CVPR 2011. IEEE, 2011.

[26] Ng, Hong-Wei, and Stefan Winkler. "A data-driven approach to cleaning large face datasets." 2014 IEEE international conference on image processing (ICIP). IEEE, 2014.

[27] Zhu, Han, et al. "Deep hashing network for efficient similarity retrieval." Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[28] Gomez-Barrero, Marta, et al. "General framework to evaluate unlinkability in biometric template protection systems." IEEE Transactions on Information Forensics and Security 13.6 (2017): 1406-1420.

[29] X. Dong, Z. Jin, and A. B. J Teoh. A genetic algorithm enabled similarity-based attack on cancellable biometrics. IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS), 2019.