This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Face Template Protection using Deep Convolutional Neural Network

Arun Kumar Jindal, Srinivas Chalamala, Santosh Kumar Jami
TCS Research, Tata Consultancy Services, India
{jindal.arun, chalamala.srao, santoshkumar.jami}@tcs.com

Abstract

The growing use of biometrics has led to rising concerns about the security and privacy of the biometric data (template) since it is unique to each individual and cannot be replaced. To address this problem, many biometric template protection algorithms have been reported but most have a trade-off between matching performance and template security. In this work, we propose a method for face template protection, which improves upon existing face template protection algorithm, to provide better matching performance. The proposed method uses deep Convolutional Neural Network (CNN), with one-shot and multi-shot enrollment, to learn a robust mapping from face images of the users to the unique binary codes (assigned to the users during enrollment phase). The cryptographic hash (like SHA-3 512) of the user's binary code represents the protected face template. The deep CNN is trained to minimize the intra-class variations and maximize the inter-class variations. During verification, given an input face image of a user, deep CNN predicts the binary code assigned to the user. The hash of the predicted binary code is matched with the hash of the actual binary code assigned to the user during enrollment. Three face datasets, namely CMU-PIE, FEI and Color FERET are used for evaluation. The proposed method improves the matching performance by ∼6% and reduces Equal Error Rate by about 4 times when compared to related work, while providing high template security.

1. Introduction

The term biometrics is defined as *automated recognition of individuals based on their unique behavioral and biological characteristics* (ISO/IEC JTC1 SC37). A typical biometric system obtains these unique behavioral and physical characterstics by acquiring the user's biometric trait (such as fingerprints, iris, face, voice, gait etc.) via a sensor. Acquired data is processed to extract the salient information (feature set). During enrollment phase, the extracted feature set is stored in the database as a template T_x. During verification, matcher module accepts two biometric templates T_x (stored template)and T_y (query template) as inputs and outputs a matching score S indicating the similarity between the two templates. If the matching score exceeds a certain threshold the user is verified successfully.

But a secure biometric system should not only accurately authenticate the user (less false rejects) and deny access to imposters (less false accepts), it should also store the templates in a secure manner. This is important because unlike credit cards and passwords which when compromised can be revoked and reissued, biometric data (template) is permanently associated with the user and cannot be replaced[18]. If a biometric template is exposed once, it is lost forever. Further, a compromised biometric template can be misused for cross-matching across databases. Therefore biometric template protection is the most important issue in designing a secure biometric system[7].

An ideal biometric template protection scheme should meet the following requirements[5][7][11][17][20]:

i. **Diversity** - Different protected biometric templates can be generated based on the same biometric data of an individual for use in different applications. These different protected templates should not allow cross matching [11] across applications.
ii. **Revocability** - It should be easy to revoke the compromised biometric template and generate a new protected biometric template based on the user's biometric data.
iii. **Security** - It should be computationally hard to reconstruct the original biometric template from the protected template.
iv. **Performance** - The recognition performance (Genuine Accept Rate (GAR), False Accept Rate (FAR) and False Reject Rate (FRR)) of the biometric system should not be degraded by the use of the biometric template protection scheme.

The major challenge in a biometric template protection method satisyfing the above requirements is the high intra-user variability in the biometric templates[5][7] and low inter user variability. High intra-user variability (caused by changes in pose, illumination, expression etc.) leads to high FRR whereas low inter-user variability leads to high FAR. To achieve high template security and matching perfor-

mance together, prior art tries to minimize intra-user variability and maximize inter-user variability, by multiple acquisitions of the user's biometric trait (here face) with variations in pose, illumination and expression. But there is a trade-off between template security and matching performance.

1.1. Contributions

To address the above problem, we propose a hybrid method for biometric (face) template protection, which improves upon existing face template protection algorithm, to provide better matching performance. The proposed hybrid method combines both the transform based approach and the biometric cryptosystems approach for template protection. Key contributions of our work are summarized below:

1. We investigate the use of one shot enrollment in template protection enabled biometric systems where strictly one image of the user's biometric trait (here face) is used for enrollment. For performance comparison, we also use multi-shot enrollment where more than one image of the user's biometric trait (here face) is used for enrollment.

2. We use deep CNN for face template protection. During enrollment phase, the deep CNN learns the robust mapping from the face images of the users to the unique binary codes (bit wise randomly generated) assigned to the users. The deep CNN makes use of the pre-trained VGG-Face model for feature vector extraction and maps it to the bit wise randomly generated unique binary codes assigned to each user. Use of pre-trained VGG-Face architecture enables the proposed deep CNN to capture uniqueness in the extracted feature set of each user thus maximizing inter-user variations. The robust mapping network, minimizes the intra-user variations while mapping the extracted feature vector to the bit wise randomly generated unique binary codes assigned to each user. During verification, given an input face image of a user, it predicts the binary code assigned to the user during enrollment. The predicted binary codes are hashed using SHA-3 512 and matched with the hash of the actual binary code assigned to the user during enrollment.

3. We improve upon the architecture for face template protection provided by Pandey et al.[13] to provide a more robust face template protection method with better matching performance.

4. We use three face datasets namely CMU-PIE, FEI and Color FERET for evaluation of our method. We compare the performance (GAR, FAR, FRR) of our face template protection method with the other algorithms[4][5][13] on the CMU-PIE dataset. The proposed method improves the matching performance by $\sim$6% and reduces Equal Error Rate (EER) by about 4 times when compared to related work, while providing high template security.

1.2. Related Work

Different approaches for face template protection can be broadly categorized as (i) biometric cryptosystems (ii) transform based approaches and (iii) hybrid approaches which combine (i) and (ii).

Biometric cryptosystems use cryptosystem based security for template protection, thus offering high security. Popular biometric cryptosystem based approaches like fuzzy commitment scheme [1][9][10][25] and fuzzy vault scheme[8][26] output an encrypted template thus offering high security. They use error correcting coding techniques to deal with intra-class variations but fail to handle large intra-class variations thus leading to low performance. To overcome this limitation we use deep CNN to minimize intra-class varaitions and maximize inter-class variations.

Transform based approaches transform the original template into a new domain. This transformation can be achieved using non-invertible transform and salting. But these approaches have a tradeoff between performance and security. Ratha et al.[19] provided three non invertible transforms, namely cartesian, polar and functional, for generating cancelable face and fingerprint templates. They achieve high template security but the matching performance is low. Teoh at al.[23] propose Random Multispace Quantization (RMQ) algorithm which uses salting based transform approach for face template protection. But the RMQ algorithm may have security issues if the user data is stolen and without user data performance is impacted. Transformation functions used in salting approaches are largely invertible and their security depends upon the key.

Hybrid approaches combine the biometric cryptosystem and transform based approaches. Feng et al. [5] proposed a hybrid approach for generating secure face templates. The proposed approach extracts the face template through a feature vector extractor. Random projection is used on the extracted face template, to project the original template into a subspace, generating a cancelable template. Discriminability preserving transform is applied to the cancelable template to enhance the discriminability and convert the real valued template into a binary template. Finally the fuzzy commitment scheme is used to protect the binary face template.

Pandey and Govindraju[12] extract features(using Histogram of Gradients (HoG) and Local Binary Pattern (LBP) histograms) from the set of selected local regions of the face. Features extracted from each local region are quantized followed by cryptographic hashing(SHA-256). Thus transformed face template is the set of hashed local features extracted from the face. The proposed algorithm had low matching accuracy and the feature space being hashed was not uniformly distributed. To overcome the shortcomings of the algorithm, Pandey et al.[13] provide another secure face template protection algorithm. The algorithm assigns

unique maximum entropy binary code(bit wise randomly generated) to each user and uses CNN to learn a mapping from face images (illumination normalized using [22]) to the binary codes. Binary code assigned to each user is hashed using cryptographic hash function(SHA-512). Thus the transformed face template is the cryptographic hash of the binary code assigned to the user. The proposed algorithm has high FRR ($\sim$5%) even at very low matching scores. FRR increases to as high as $\sim$80% for matching scores as high as 1 due to high intra-user variability. We improve upon this work using a better and deeper CNN to learn a robust mapping from face images to the binary codes. The deep CNN minimizes intra-user variability and maximizes inter-user variability, to provide a more robust face template protection method with significantly better matching performance (with both one shot and multi-shot enrollment) even for matching scores as high as 1.

2. Methodology

In this section, we describe each component of the proposed method shown in Figure 1.

2.1. Sensor

User images are captured via a sensor. In one shot enrollment, only one image per user is used for enrollment and the rest images are used for verification. In multi-shot enrollment, multiple images per user are used for enrollment and the remaining images are used for verification.

2.2. Pre-Processing

i. **Face Detection**: Each of the acquired user image is subject to face detector algorithm[3] to identify the human face in each image. Detected human faces are resized to a 224*224 image and saved.
ii. **Data Augmentation**: Data augmentation is particularly required in case of biometric data since the training samples are limited and deep networks require large number of training data to achieve good performance. Since we also use one-shot enrollment, where only one face image of the user is used for enrollment, we perform data augmentation on the image to increase the number of samples per user. We perform data augmentation on each face image, using Keras[2] image data generator API with operations like horizontal flip, zoom, re-scaling, changing the shear angle, rotation to generate five augmented images. For each augmented image of size $m*m$ we extract all possible crops of size $n*n$. This yields a total of $(m-n+1)*(m-n+1)$ crops. The crops are then resized back to $m*m$. Data augmentation yields a total of $5*(m-n+1)*(m-n+1)$ images for each face image. Value of m is 224 and n is chosen as 221.

2.3. Binary Code Generation and Assignment

The first step of enrollment process is the generation of unique binary codes with maximum entropy. To maximize the entropy, each bit of the binary code is randomly generated and have no correlation with the original biometric sample (face images captured by the sensor), as in [13]. Each user being enrolled is assigned a unique binary code. These binary codes are used internally for training the deep CNN during enrollment phase only and are neither exposed to the user nor retained in an unprotected form post training. Cryptographic hash (SHA3-512) of the unique binary code assigned to a user, representing the secure face template of the user, is computed and stored in the database. To make the brute force attacks in the binary code domain infeasible, we use the binary codes with K=256 and K=1024 bits (as in [13]).

This method of binary code generation and assignment satisfies the diversity and revocability requirements of a biometric template protection scheme. It is to be noted that different applications can assign different binary codes to each enrolled user. An application can also change the binary codes assigned to its enrolled users (or enroll new users by assigning them new binary codes) and re-learn the mapping of the users face images to the assigned binary codes.

2.4. Deep Convolutional Neural Network(CNN)

To learn a robust mapping of the users face images to the assigned binary codes, we use deep CNN. The deep CNN maximizes the inter-user variations and minimizes the intra-user variations to give high matching performance. Figure 2 and Figure 3 represent the deep CNNs to map the face images to 256 bit and 1024 bit binary codes respectively.

The deep CNN has two components, as shown in the block diagram in Figure 1. First component uses a pre-trained VGG-Face CNN to extract a 4096 dimensional feature vector corresponding to each input face image. The extracted feature vector serves as a highly discriminative and compact encoding of the input face image. Second component maps the extracted feature vector to the 256 bit or 1024 bit binary code.

2.4.1 Feature Vector Extraction Network

We use the pre-trained VGG-Face CNN (trained over 2.6M images over 2.6K people) for feature vector extraction. VGG-Face CNN descriptors are computed using the CNN implementation of VGG-Very-Deep-16 CNN architecture (comprised of 16 weight layers) provided by Parkhi et al.[14].

The input to the pre-trained VGG-Face CNN is a 224*224 face image with the average face image (computed from the training set) subtracted. Therefore in pre-processing(as discussed in section 2.2), we resize all face

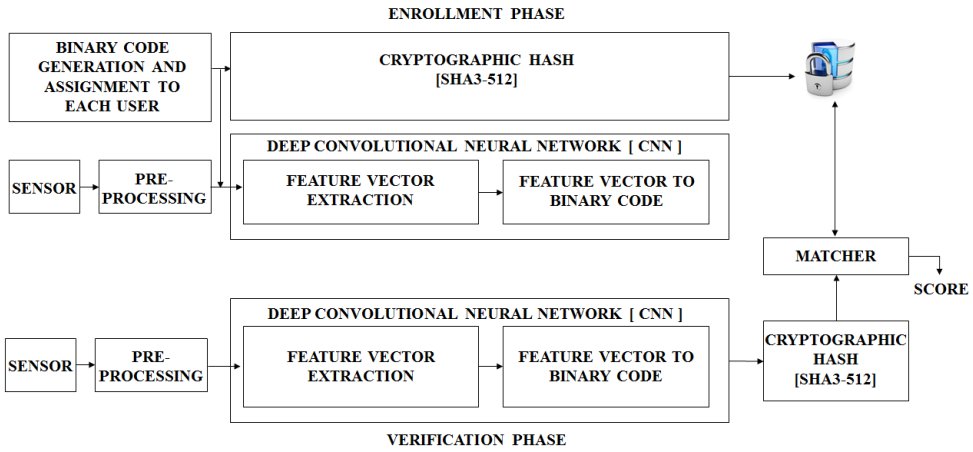


Figure 1: Block Diagram of the proposed method

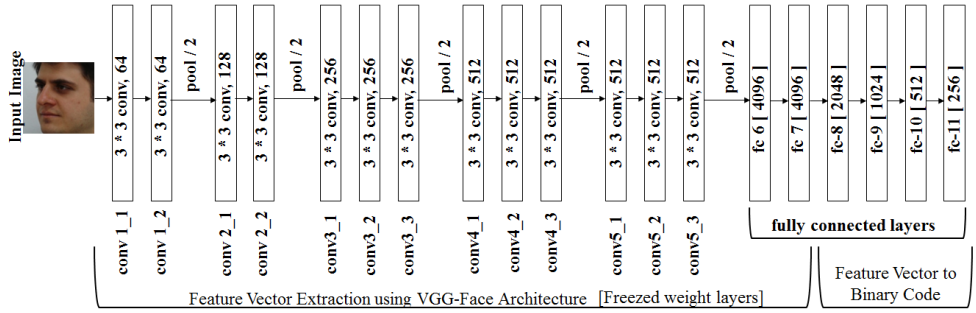


Figure 2: Deep Convolutional Neural Network to map face images to 256 bit binary code.

images to 224*224 both during face detection and data augmentation.

In the proposed method, we use first 15 weight layers (comprised of 13 convolutional layers and 2 fully connected layers) adapted from the VGG-Face CNN. These layers are common to both the deep CNN in Figure 2 and Figure 3. The 4096 dimensional output of the fully connected layer (represented by fc7 layer) is regarded as the feature vector of the input face image.

The pre-trained VGG-Face architecture captures the uniqueness in the extracted feature set of each user thus maximizing inter-user variations.

2.4.2 Feature Vector to Binary Code Mapping Network

In the enrollment phase, the deep CNN learns a robust mapping of the extracted feature vector (representing users face image), to the assigned binary code.

To learn this mapping, some modifications to the CNN training procedure are made. The one hot encoding of class labels is replaced by the binary codes assigned to each user. Consequently, the last layer of the neural network uses the sigmoid activation function, instead of softmax, since multiple bits of the network output will be one instead of a single bit. The network uses binary crossentropy as the loss function.

Depending upon the binary code (K=256, K=1024) dimension, we use different number of fully connected layers in the neural network. To map the 4096 dimensional feature vector to a 256 bit binary code we use 4 fully connected layers (namely fc-8[dim:2048], fc-9[dim:1024], fc-10[dim:512], fc-11[dim:256]) as shown in Figure 2. To map the 4096 dimensional feature vector to a 1024 bit binary code we use 6 fully connected layers (namely fc-8[dim:3584], fc-9[dim:3072], fc-10[dim:2560], fc-11[dim:2048]), fc-12[dim:1536], fc-13[dim:1024]) as shown in Figure 3. The number of neurons in each fully connected layer is reduced gradually enabling the network to learn a robust mapping of the extracted feature vector to the assigned binary code.

In the verification phase, given an input face image, the deep CNN (trained in enrollment phase) predicts the binary code (i.e. network output) assigned to the user at the time of enrollment. The output is binarized using a simple thresholding operation, where the neuron value is set to 1 if greater

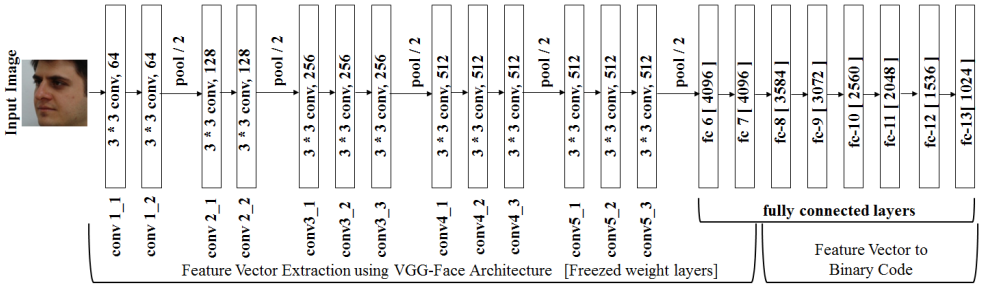


Figure 3: Deep Convolutional Neural Network to map face images to 1024 bit binary code.

than 0.5 else it is set to 0, thus predicting the binary code corresponding to the input face image.

The mapping network described above, minimizes the intra-user variations while mapping the extracted feature vector to the bit wise randomly generated unique binary code assigned to each user.

2.5. Cryptograhic Hash (SHA-3 512)

To protect the template, represented by the unique binary code assigned to each user, we use the Secure Hash Algorithm 3 (SHA-3) which is the latest addition to the Secure Hash Algorithm family of standards. We use the SHA3-512 variant of the SHA-3 family for template protection. The input to the SHA3-512 algorithm is the binary code assigned to the user and the output is the 512 bit cryptographic hash which represents the protected face template.

During enrollment phase, the SHA-3 512 hash of each binary code, representing the protected face template, is stored in the database. During verification phase, the SHA-3 512 hash of the binary code predicted by the trained deep CNN is computed.

2.6. Matcher

During verification phase, matcher module accepts two biometric templates T_x(protected face template) and T_y(query template) and outputs a matching score S of true/false nature. To obtain a tunable score to adjust the FAR and FRR of the biometric system, several data augmented images are taken for each image presented for verification. Deep CNN (trained in enrollment phase) is used to predict the binary code corresponding to each augmented image. SHA3-512 hash of each of the predicted binary code is computed, thus yielding a set of hashes H. As in [13], the final matching score is defined as the number of hashes in H that match the stored template, scaled by the cardinality of H.

3. Experiments

In this section we describe the databases, evaluation metrics and the experimental parameters used in the evaluation.

3.1. Databases

We use the following public domain databases for our experiments:

i. The CMU PIE [21] database consists of 41368 images of 68 subjects. Each subject has images under 43 different illumination conditions, 13 different poses and 4 different expressions. We use 5 poses (p05, p07, p09, p27, p29) and all illuminations variations for our experiments. In one shot enrollment we randomly select one image per user for training and the rest are used for testing. In multi-shot enrollment, 10 images are randomly chosen for training and the rest are used for testing, as done in [5][13].

ii. The FEI [24] database contains 2800 color images of 200 subjects. Each subject has 14 images with pose rotation of upto about 180 degrees. Out of 14 images, each subject has two full frontal poses with expression variation (non-smiling expression and smiling facial expression). We use 9 poses (p03, p04, p05, p06, p07, p08, p11, p12, p13) for our experiments. In one shot enrollment we randomly select one image per user for training and the rest are used for testing. In multi-shot enrollment, 4 images are randomly chosen for training and the rest 5 are used for testing.

iii. In the color FERET [15][16] database, 237 individuals are selected and each individual has 4 different face images. There are pose, illumination, age and occlusion(glasses) variation in the FERET database. In one shot enrollment we randomly select one image per user for training and the rest are used for testing. In multi-shot enrollment, 2 images are randomly chosen for training and the rest 2 are used for testing.

3.2. Evaluation Metrics

We report Equal Error Rate (EER) as the evaluation metric. Since the train-test splits used are randomly generated, we report the mean and standard deviation of EER for 5 different random train-test splits. We also report the mean of Genuine Accept Rate (GAR) at different False Accept Rate (FAR).

3.3. Experimental Parameters

The architecture of the deep CNN used to map the users face images to the assigned binary codes is described below:

To extract the feature vector, we adopt the first 15 weight layers (comprised of 13 convolutional layers in 5 convolution blocks and 2 fully connected layers) of the pre-trained VGG-Face CNN. The convolutional blocks are: Convolution block 1 (having conv1_1 and conv1_2 layers) each having 64 filters of size 3 x 3, convolution block 2 (having conv2_1 and conv2_2 layers) each having 128 filters of size 3 x 3, convolution block 3 (having conv3_1 conv3_2 and conv3_3 layers) each having 256 filters of size 3 x 3, convolution block 4 (having conv4_1 conv4_2 and conv4_3 layers) each having 512 filters of size 3 x 3, convolution block 5 (having conv5_1 conv5_2 and conv5_3 layers) each having 512 filters of size 3 x 3. Each convolution block is followed by max pooling layer of size 2 x 2. This is followed by 2 fully connected layers (fc6 and fc7) each of size 4096. Rectifier activation function is used in all layers. The 4096 dimensional output of the fc7 layer represents the feature vector of the input face image. During training the weights of the first 15 weight layers, described above, is freezed.

To learn a robust mapping of the extracted feature vector to the assigned binary code we use a sequence of fully connected layers. To map the extracted feature vector to a 256 bit binary code (Figure 2) we use fully connected layers (fc-8, fc-9, fc-10 and fc-11) with dimensions of 2048, 1024, 512 and 256 respectively. To reduce overfitting in the deep CNN, we apply dropout[6] in all these fully connected layers with 0.25 probability of discarding one hidden activation. To map the extracted feature vector to a 1024 bit binary code (Figure 3) we use fully connected layers (fc-8, fc-9, fc-10, fc-11, fc-12 and fc-13) with dimensions of 3584, 3072, 2560, 2048, 1536 and 1024 respectively. To reduce overfitting, we apply dropout[6] in all these fully connected layers with 0.10 probability of discarding one hidden activation. In both the deep CNN, each of the fully connected layers, except last fully connected layer uses rectifier activation function. The last fully connected layer uses sigmoid activation function, as discussed in 2.4.2.

The proposed deep CNN (in Figure 2 and Figure 3) is trained by minimizing the binary cross entropy loss for 50 epochs using adam for stochastic optimization with batch size of 16.

During verification, the trained deep CNN is used to predict the binary code assigned to a user, given the input face image. The output is binarized using a simple thresholding operation, where the neuron value is set to 1 if greater than 0.5 else it is set to 0, thus predicting the binary code corresponding to the input face image.

4. Results

The experimental results are shown in Table 1. We report the mean and standard deviation of Equal Error Rate (EER)and the mean Genuine Accept Rate (GAR) at different False Accept Rate (FAR) for 5 different train-test splits, with one-shot enrollment and multi-shot enrollment, using binary code with dimensions K=256 and K=1024.

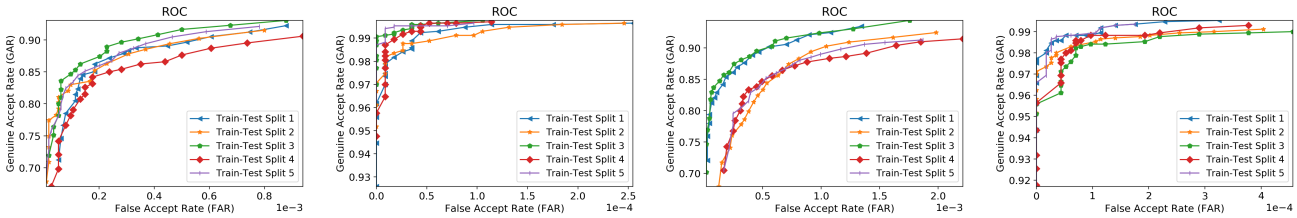
For PIE dataset, we achieve GAR of ∼91%@0.1%FAR with one shot enrollment and GAR of ∼97%@0%FAR with multi shot enrollment in PIE dataset with K=1024. We achieve GAR of ∼95%@0.01%FAR with one shot enrollment and GAR of ∼99%@0%FAR with multi-shot enrollment in FEI dataset with K=1024. We achieve a lower GAR of ∼81%@0.01%FAR with one shot enrollment and GAR of ∼92%@0.01%FAR with multi-shot enrollment with K=1024. The relatively lower performance in color FERET dataset can be attributed to the fact that it has pose, illumination, age and occlusion(glasses) variations whereas PIE and FEI datasets have only pose and illumination variations. To get an idea of the verification performance we also show the ROC curves for PIE, FEI and color FERET datasets in Figure 4, Figure 5 and Figure 6.

It is to be noted that the proposed method not only has very low FAR values even at low matching scores but it also has very low FRR (or high GAR) values even at matching scores as high as 1. For example with PIE dataset, with one-shot enrollment, we get more than 67% GAR even for matching scores as high as 1 for both K=256 and K=1024. For PIE dataset, with multishot enrollment, we get more than 90%GAR@0%FAR values even for matching scores as high as 1 for both K=256 and K=1024. This is in contrast to the results reported in [13] where GAR dips to ∼20% for matching score of 1. This validates that the proposed method successfully minimizes the intra-class variations and maximizes the inter-class variations.

Dataset	Enrollment Type	K	GAR@FAR	EER
PIE	One-Shot	256	91.91%@0.1%FAR	4.00% ± 0.41%
		1024	91.34%@0.1%FAR	3.60% ± 0.58%
	Multi-Shot	256	97.35%@0%FAR	0.15% ± 0.02%
		1024	96.53%@0%FAR	0.35% ± 0.09%
FEI	One-Shot	256	96.05%@0.01%FAR	1.97% ± 0.03%
		1024	95.19%@0.01%FAR	1.81% ± 0.19%
	Multi-Shot	256	98.54%@0%FAR	0.16% ± 0.14%
		1024	99.10%@0%FAR	0.20% ± 0.14%
Color FERET	One-Shot	256	86.92%@0.01%FAR	5.50% ± 0.40%
		1024	80.90%@0.01%FAR	6.67% ± 0.39%
	Multi-Shot	256	94.85%@0.01%FAR	2.16% ± 0.45%
		1024	92.70%@0.01%FAR	3.03% ± 0.35%

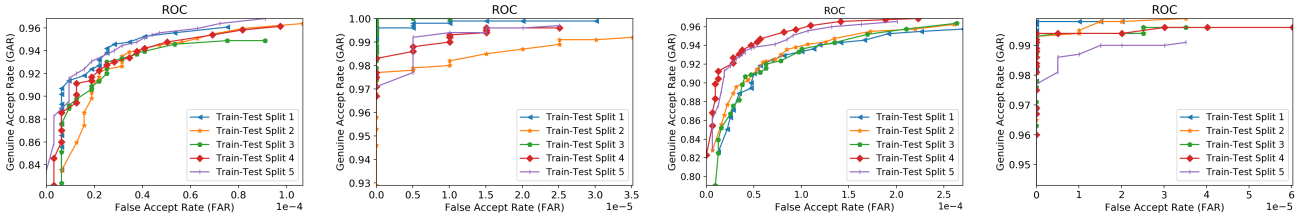
Table 1: Verification results from different datasets

A comparison of our results with other face template protection algorithms on PIE dataset is shown in Table 2. We compare the binary code dimensionality parameter K to the equivalent parameter in the shown approaches. It is noteworthy that even with one-shot enrollment in PIE dataset, we report a lower EER of 4%



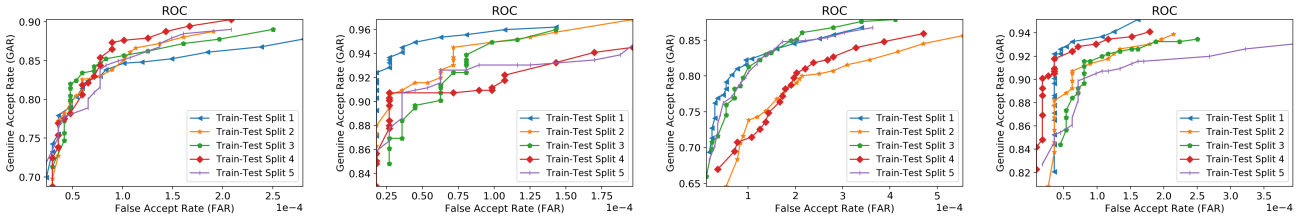
(a) Enrollment:One-Shot,K=256 (b) Enrollment:Multi-Shot,K=256 (c) Enrollment:One-Shot,K=1024 (d) Enrollment:Multi-Shot,K=1024

Figure 4: ROC curves for PIE dataset [Enrollment Type: One-Shot, Multi-Shot][K=256, 1024]



(a) Enrollment:One-Shot,K=256 (b) Enrollment:Multi-Shot,K=256 (c) Enrollment:One-Shot,K=1024 (d) Enrollment:Multi-Shot,K=1024

Figure 5: ROC curves for FEI dataset [Enrollment Type: One-Shot, Multi-Shot][K=256, 1024]



(a) Enrollment:One-Shot,K=256 (b) Enrollment:Multi-Shot,K=256 (c) Enrollment:One-Shot,K=1024 (d) Enrollment:Multi-Shot,K=1024

Figure 6: ROC curves for Color FERET dataset [Enrollment Type: One-Shot, Multi-Shot][K=256, 1024]

(for K=256) and 3.6% (for K=1024) when compared to [5] which reports an EER of 6.81% for K=210. We also report a better GAR of ∼91%@0.1% FAR whereas [5] reports 90.6%GAR@1%FAR. With multi-shot enrollment in PIE dataset, we outperform the face template protection algorithms [4][5][13] in terms of both matching performance and Equal Error Rate (EER). We achieve 96.53%GAR@0%FAR (for K=1024) which is ∼6% improvement in matching performance compared to 90.13%GAR@0%FAR reported in [13]. We report an EER of 0.35% (for K=1024) which is about 4 times less when compared to EER of 1.14% reported in [13].

Method	Enrollment Type	K	GAR@FAR	EER
Hybrid Approach[5]	Multi-Shot	210	90.61%@1%FAR	6.81%
BDA[4]	Multi-Shot	76	96.38%@1%FAR	-
MEB Encoding[13]	Multi-Shot	256	93.22%@0%FAR	1.39%
		1024	90.13%@0%FAR	1.14%
Our Method	**One-Shot**	256	**91.91%@0.1%FAR**	**4.00%**
		1024	**91.34%@0.1%FAR**	**3.60%**
	Multi-Shot	256	**97.35%@0%FAR**	**0.15%**
		1024	**96.53%@0%FAR**	**0.35%**

Table 2: Performance comparison with other algorithms on PIE Dataset

4.1. Security Analysis

As discussed in [13], we analyze the security of the proposed method in the scenario where the protected face template is stolen. As discussed in section 2.5, the protected face template is the SHA-3 512 hash of the unique binary code (bit wise randomly generated) assigned to the user during the enrollment phase. It is to be noted that the original binary codes are used internally for training the deep CNN during enrollment phase only and are neither exposed to the user nor retained in an unprotected form post training.

In the scenario where the attacker has access to only the stolen protected face template with no knowledge of the deep CNN model (being used to map face images to unique binary codes), no information about the original binary codes can be extracted from the stolen protected face template. This is due to the pre-image resistance property of the cryptographic hash functions. Therefore in such a scenario, only brute force attacks can reveal the binary codes. But the brute force attacks in this scenario are computationally infeasible since for a 256 bit and 1024 bit binary code,

the search space would 2^{256} and 2^{1024}.

In the scenario, where the attacker has access to both the stolen protected face templates and the deep CNN model (being used to map face images to unique binary codes), the attacker would attempt to generate attacks to exploit the FAR of the system. To exploit the system FAR, the attacker may perform a dictionary attack using a large set of faces. In this attack scenario, the minimal FAR of the proposed method is a reasonably good measure of the face template security. To evaluate the template security, we study the genuine and imposter score distributions (as done in [13]) for the following dictionary attacks on the proposed method with the deep CNN model trained with multi-shot enrollment and K=1024:(i)In the first attack we use FEI database as the genuine face database and PIE database as an attacker database.(ii)In the second attack we use PIE database as the genuine face database and FEI database as an attacker database. The genuine and imposter score distributions for the above dictionary attacks reveal that the imposter scores tend to zero and genuine scores tend to 1 thus showing that it is unlikely for the proposed method to falsely accept the external faces for the enrolled ones.

5. Conclusion and Future Work

In this work we present a method for face template protection using deep CNN. We show that a deep CNN can be used to provide better matching performance even with one-shot enrollment in a template protection enabled biometric system. The matching performance is even better with multi-shot enrollment. The proposed method improves the matching performance by $\sim$6% and reduces EER by about 4 times when compared to related work, while providing high template security. The current work deals with the problem of template protection for faces which is a physiological biometric trait. We plan to extend this work to come up with template protection algorithm for behavioral biometrics like voice, key stroke dynamics and gait patterns.

References

[1] M. Ao and S. Z. Li. Near infrared face based biometric key binding. In *International Conference on Biometrics*, pages 376–385. Springer, 2009. 2

[2] F. Chollet et al. Keras, 2015. 3

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 3

[4] Y. C. Feng and P. C. Yuen. Binary discriminant analysis for generating binary face template. *IEEE Transactions on Information Forensics and Security*, 7(2):613–624, 2012. 2, 7

[5] Y. C. Feng, P. C. Yuen, and A. K. Jain. A hybrid approach for generating secure and discriminating face template. *IEEE transactions on information forensics and security*, 5(1):103–117, 2010. 1, 2, 5, 7

[6] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 6

[7] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on advances in signal processing*, 2008:113, 2008. 1

[8] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006. 2

[9] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36. ACM, 1999. 2

[10] H. Lu, K. Martin, F. Bui, K. Plataniotis, and D. Hatzinakos. Face recognition with biometric encryption for privacy-enhancing self-exclusion. In *Digital Signal Processing, 2009 16th International Conference on*, pages 1–8. IEEE, 2009. 2

[11] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009. 1

[12] R. K. Pandey and V. Govindaraju. Secure face template generation via local region hashing. In *Biometrics (ICB), 2015 International Conference on*, pages 299–304. IEEE, 2015. 2

[13] R. K. Pandey, Y. Zhou, B. U. Kota, and V. Govindaraju. Deep secure encoding for face template protection. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 77–83. IEEE, 2016. 2, 3, 5, 6, 7, 8

[14] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 3

[15] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10):1090–1104, 2000. 5

[16] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998. 5

[17] S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: Security and privacy concerns. *IEEE security & privacy*, 99(2):33–42, 2003. 1

[18] N. K. Ratha. Privacy protection in high security biometrics applications. In *Ethics and Policy of Biometrics*, pages 62–69. Springer, 2010. 1

[19] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on pattern analysis and machine intelligence*, 29(4):561–572, 2007. 2

[20] C. Rathgeb and A. Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):3, 2011. 1

[21] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 53–58. IEEE, 2002. 5

[22] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6):1635–1650, 2010. 3

[23] A. B. Teoh, A. Goh, and D. C. 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):1892–1901, 2006. 2

[24] C. E. Thomaz and G. A. Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 5

[25] M. Van Der Veen, T. Kevenaar, G.-J. Schrijen, T. H. Akkermans, and F. Zuo. Face biometrics with renewable templates. In *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, page 60720J. International Society for Optics and Photonics, 2006. 2

[26] Y. Wu and B. Qiu. Transforming a pattern identifier into biometric key generators. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 78–82. IEEE, 2010. 2