2021

Integration of Deep Hashing and Channel Coding for Biometric Security and Biometric Retrieval

Veeru Talreja
vtalreja@mix.wvu.edu

Recommended Citation

Talreja, Veeru, "Integration of Deep Hashing and Channel Coding for Biometric Security and Biometric Retrieval" (2021). *Graduate Theses, Dissertations, and Problem Reports*. 8240.
https://researchrepository.wvu.edu/etd/8240

2021

Integration of Deep Hashing and Channel Coding for Biometric Security and Biometric Retrieval

Veeru Talreja

Integration of Deep Hashing and Channel Coding for Biometric Security and Biometric Retrieval

Veeru Talreja

Dissertation submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Electrical Engineering

Matthew C. Valenti, Ph.D., Chair
Jeremy Dawson, Ph.D.
Nasser M. Nasrabadi, Ph.D.
Natalia A. Schmid, Ph.D.
Omid Dehzangi, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2021

Keywords: Channel Coding, Deep Learning, Hashing, Multibiometrics, Secure-sketch, Template Security, GAN

Abstract

Integration of Deep Hashing and Channel
Coding for Biometric Security and Biometric Retrieval

Veeru Talreja

In the last few years, the research growth in many research and commercial fields are due to the adoption of state of the art deep learning techniques. The same applies to even biometrics and biometric security. Additionally, there has been a rise in the development of deep learning techniques used for approximate nearest neighbor (ANN) search for retrieval on multi-modal datasets. These deep learning techniques knows as *deep hashing (DH)* inte-grate feature learning and hash coding into an end-to-end trainable framework. Motivated by these factors, this dissertation considers the integration of deep hashing and channel coding for biometric security and different biometric retrieval applications. The major focus of this dissertation is biometric security, wherein deep hashing is integrated with channel coding to develop a secure biometric authentication system. In this system, multiple biometric modal-ities of a single user are combined at the feature level using deep hashing (binarization). A hybrid secure architecture that combines cancelable biometrics with secure sketch techniques is integrated with the deep hashing framework, which makes it computationally prohibitive to forge a combination of multiple biometrics that passes the authentication. The integration of deep hashing and channel coding not only finds application in biometric security but it can also be extended to different biometric applications. To this end, the integration of deep cross-modal hashing and error correcting codes has been extended to improve the efficiency of attribute-guided face image retrieval.

Additionally, the dissertation also presents a framework for cross-resolution (low-resolution to high-resolution) face recognition, and profile-to-frontal face recognition. A novel attribute-guided cross-resolution (low-resolution to high-resolution) face recognition system that leverages a coupled generative adversarial network (cpGAN) structure with adversarial training to find the hidden relationship between low-resolution and high-resolution images in a latent common embedding subspace is developed and presented. A similar framework that leverages cpGAN structure has been developed for a profile-to-frontal face recognition system. Finally, the performance of this cpGAN architecture for profile-to-frontal face recognition system has been evaluated and compared with a coupled convolutional neural network (cpCNN) and an adversarial discriminative domain adaptation (ADDA) network.

Acknowledgements

This PhD “journey” has been a full circle of acquiring knowledge, facing challenges, and having a personal and professional development. It would not have been possible without support and motivation from some of the amazing people in my life. I would like to take this opportunity to thank all these people who have been by my side and contributed in some or the other way to this journey. First and foremost, I would sincerely like to thank my advisor and committee chair Dr. Matthew C. Valenti for accepting me as his PhD student, for his thoughtful guidance, understanding, and his mentorship throughout this journey. He has not only helped me to grow professionally but also personally. I have learnt a lot of valuable lessons from him such as professionalism, optimism, work-ethics, attention to details, commitment to student development, and perseverance. In real terms, he has been a perfect role model for me and I am truly blessed and honored to have such an amazing advisor and will always be thankful to him for his contributions to my professional and personal development.

I have had this rare opportunity of working with two amazing professors at the same time. One is my advisor Dr. Matthew C. Valenti and the other is Dr. Nasser M. Nasrabadi who was no less than an advisor during this journey. I would like to sincerely thank Dr. Nasser M. Nasrabadi for welcoming me in his group and letting me collaborate with his students. I am grateful to him for helping me with new ideas and always pushing me to do better. I have also learnt a lot from him at both professional and personal level. His enthusiastic attitude, and his hard work have inspired me from the very first moment I took his class and started collaborating with him.

I would like to thank Dr. Natalia A. Schmid, Dr. Jeremy Dawson, and Dr. Omid Dehzangi for being on my committee, and for all their valuable insight into my research. I have been very blessed to take classes and work with some of them, and I truly cherish the knowledge and experience.

I would like to thank all the students of the wireless communications research lab with whom I have had the pleasure of working with. In particular, I would like to thank Dr. Terry Ferrett, who was a great help when I started my PhD and I learnt a lot from him. The credit for my knowledge of linux and shell scripting goes to Dr. Terry Ferrett. He is a very valuable friend and I specially liked our technical discussions and meetings in Panera. I would also like to thank Aruna Sri Bommagani, Dr. Salvatore Talarico, and Dr. Mohammad Fanaei for a lot of technical discussions and helping me set my feet in at the beginning of my PhD.

I would also like to take this opportunity to thank all the past and present students of the Deep Learning Lab. Specially, I would like to thank my good friend and collaborator Dr. Fariborz Taherkhani for all his support, technical and cultural discussions, and all the work that we did together. I am grateful to him for collaborating with me and sharing his technical knowledge and helping me grow. I would also like to thank Dr. Sobhan Soleymani for always being present for any kind of discussion, be it technical, social, political, or ethical. He was always very understanding and supportive. Furthermore, I would like to thank Domenick Poster, Dr. Hadi Kazemi, Dr. Mehdi Iranmanesh, Ali Dabouei, Uche Osahor, Moktari Mostafa, Nyma Ferdous, and Salman Mohamadi for great discussions on varying topics from machine learning to political outcomes.

I would also like to thank all the member universities and affiliates of Center for Identification Technology Research (CITeR). My major funding during PhD came from CITeR. I think the immense experience that I have gained while doing the CITeR projects has been very valuable in helping me grow as a researcher. Additionally, the exposure and knowledge sharing among the great group of researchers during the CITeR meetings and workshops has helped me stay updated and learn new advancements in the field of biometrics and computer vision.

West Virginia University (WVU) has always been a home away from home. This has been made possible by some of the special friends who have always been on my side and supported me during this journey. Special thanks to Dr. Deepak Tiwari who has been more like a brother at WVU and helped me during the ups and down in this journey. Also, thanks to Moiz Usmani and Vishal Verma for helping me through this journey with all the recreational fun time and night outs playing board games.

I would like to thank my parents-in-law, and brothers-in-law for their unfailing emotional support and heart-warming kindness throughout this journey. They always believed in me and supported me, and for that I will be forever thankful to them.

I am forever indebted to my parents, and siblings for giving me the opportunities and experiences that have made me who I am. They selflessly encouraged me to explore new directions in life and seek my own destiny. I am forever grateful to my parents for their unconditional trust, timely encouragement, and endless patience.

Finally, and most importantly, I would like to thank with love my wife Kanchan Talreja, my first daughter Vihana Talreja, and my second daughter (expected in Nov., 2021). They have stood by me through all my travails, my absences, my fits of pique and impatience. My PhD would not have been possible without Kanchan's belief in me and her motivation to push me to do better and achieve great heights. Her sacrifices, support, love, and encouragement has helped me get through this amazing journey in the most positive way. It was Kanchan's PhD as much as it was mine and so, I dedicate this thesis to her

Contents

List of Figures

List of Tables

List of Abbreviations

ADDA	adversarial discriminative domain adaptation
ADCMH	attribute-based deep cross-modal hashing
ANN	approximate nearest neighbor
BP	belief propagation
BLA	bilinear architecture
BCH	Bose–Chaudhuri–Hocquenghem
CTM	cancelable template module
CNN	convolutional neural network
cpGAN	coupled generative adversarial network
cpCNN	coupled convolutional neural network
CMH	cross-modal hashing
DH	deep hashing
DCMH	deep cross-modal hashing
DFB	deep feature extraction and binarization
DNDCMH	deep neural decoder cross-modal hashing
DSL	domain-specific layer
ECC	error-correcting code
EER	equal error rate
FAR	false accept rate
FRR	false reject rate
FEC	forward error correction

FCA	fully concatenated architecture
GAR	genuine accept rate
GAN	generative adversarial network
JRL	joint representation layer
LDPC	low-density parity check
MAP	mean average precision
MDH	multimodal deep hashing
MDHND	multimodal deep hashing neural decoder
NECD	neural error-correction decoder
NND	neural network decoder
NDCG	normalized discounted cumulative gain
ROC	receiver operating characteristic
RS	Reed-Solomon
SSTM	secure sketch template module

Chapter 1

Introduction

This chapter provides an introduction to the contributions presented in this dissertation. A conceptual background is provided to give context to the underlying goals and contributions. A contextual description and literature review is provided for the fundamental technical concepts.

Biometrics are difficult to forge, and unlike in traditional password-based access control systems, they do not have to be remembered. As much as these characteristics provide an advantage, they also create challenges related to protecting biometrics in the event of identity theft or a database compromise as each biometric characteristic is distinct and cannot be replaced by a newly generated arbitrary biometric. There are serious concerns about the security and privacy of an individual because of the proliferation of biometric usage. The security and privacy concerns cannot be alleviated by using conventional cryptographic hashing methods as in the case of alpha-numeric passwords because the cryptographic hashes are extremely sensitive to noise and are not suitable for the protection of biometrics due to inherent variability and noise in biometric measurements.

The leakage of biometric information to an adversary constitutes a serious threat to security and privacy because if an adversary gains access to a biometric database, he can potentially obtain the stored user information. The attacker can use this information to gain unauthorized access to the system by reverse engineering the authentication process and creating a physical spoof. Furthermore, an attacker can abuse the biometric information for unintended purposes and violate user privacy [2].

Multimodal biometric systems use a combination of different biometric traits such as face and iris, or face and fingerprint. Multimodal systems are generally more resistant to spoofing attacks [3]. Moreover, multimodal systems can be made to be more universal than unimodal systems, since the use of multiple modalities can compensate for missing modalities in a small portion of the population. Multimodal systems also have an advantage of lower error rates and higher accuracy when compared to unimodal systems [2]. Consequently, multimodal systems have been deployed in many large scale biometric applications including the FBI's Next Genration Identification (NGI), the Department of Homeland Security's US-VISIT, and the Government of India's UID. However, Multimodal systems have an increased demand for integrity and privacy because the system stores multiple biometric traits of each user. The main focus of this dissertation is the development and analysis of a secure multibiometric system.

In biometrics systems, the biometric characteristics of an individual are encoded into a data structure that is commonly referred to as a *template*. Because templates may contain personally identifiable information, it is essential that they be protected in order to preserve privacy. For a template to be secure, it must satisfy the important properties of *noninvertibility* and *revocability*. Noninvertibility implies that given a template, it must be computationally difficult to recover the original biometric data from the template. Revocability implies that if a template gets compromised, it should be possible to revoke the compromised template and generate a new template using a different transformation. Moreover, it should be difficult to identify that the new template and the old compromised template are generated from the same underlying biometric data.

The fundamental challenge in designing a biometric template protection scheme is to manage the intra-user variability that occurs due to signal variations in the multiple acquisitions of the same biometric trait. With respect to biometric template protection, four main architectures are widely used: *fuzzy commitment, secure sketch, secure multiparty computation, and cancelable biometrics* [1]. Fuzzy commitment and secure sketch are biometric cryptosystem methods and are usually implemented with error-correcting code (ECC) and provide information-theoretic guarantees of security and privacy (e.g., [4–8]). Secure multiparty computation architectures are distance based and use cryptographic tools. Cancelable biometrics use revocable and non-invertible user-specific transformations for distorting the enrollment biometric (e.g., [9–12]), with the matching typically performed in the transformed domain.

Fuzzy commitment, a classical method of biometric protection, was first proposed by Juels and Wattenberg [5] in 1999. forward error correction (FEC) based fuzzy commitment can also be viewed as a method of extracting a secret code by means of polynomial interpolation [6]. An implementation example of such a fuzzy commitment scheme appears in [8], wherein a BCH code is employed for polynomial interpolation; experiments show that when the degree of the interpolated polynomial is increased, the matching becomes more stringent, reducing the false accept rate (FAR), but increasing the false reject rate (FRR).

Cancelable biometrics was first proposed by Ratha *et al.* [9], after which, there have been various different methods of generating cancelable biometric templates. Some of the popular methods use non-invertible transforms [9], bio-hashing [10], salting [11] and random projections [12]. Literature surveys on cancelable biometrics can be found in [1], and [13].

1.0.1 Secure Biometric System Model

In this section, we describe a generalized biometric framework within which we can evaluate and analyze secure biometric system. Consider the general model of secure biometric system given in Fig. 1.1 [1]. This system consists of a suitable procedure to encode the enrollment biometric signal into data that can be stored on the access control device. The other important part of the system entails a decoding procedure to use the probe biometric signal and combine it with the stored data to generate an authentication decision. Note that this is an authentication system where the probe biometric is matched against the enrollment of a claimed user. This authentication system differs from an identification system, in which a probe biometric is matched against each enrollment in the database to find the identity associated with the probe [1].

Biometric Feature Vectors: The enrollment feature vector $\mathbf{A} = (A_1, A_2, ...A_n)$ is extracted from the biometric measurement provided by the user using a feature extraction algorithm. This feature vector is used to generate the secure template that is stored on

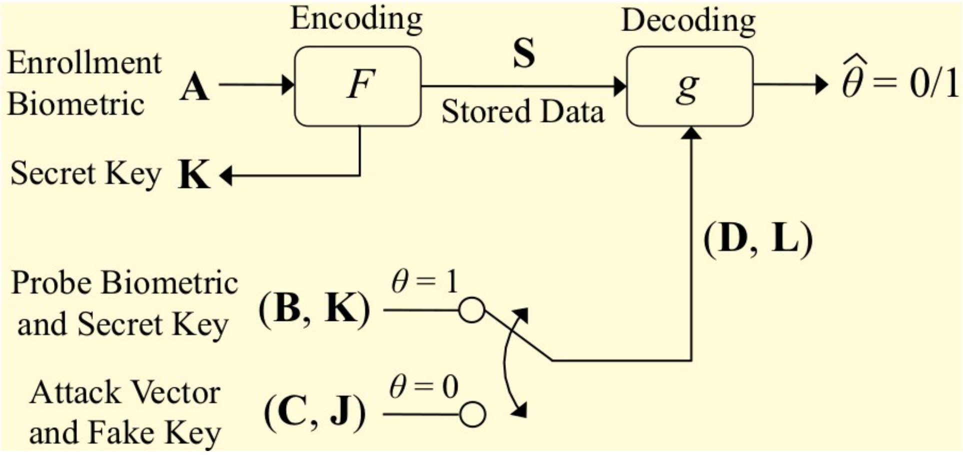


Figure 1.1: Generalized biometric secure framework (from [1])

access control device. During authentication, the user provides a biometric measurement, from which a probe feature vector $\mathbf{B} = (B_1, B_2, ...B_n)$ is extracted using the same feature extraction algorithm. This probe feature vector is decoded by the decoding procedure on access control device to authenticate the user. An important note here is that most theoretical analyses of secure biometric systems work with the feature vectors as a conceptual model for the biometric signals [1] and omit the feature extraction algorithm. Another important note is that it is easier to analyze the models where the feature vectors are binary with certain statistical properties. In general, the bits extracted from the biometric measurements are neither independent nor identically distributed. However, it is possible to convert biometric readings to bits that are independent and identically distributed (i.i.d.) as equiprobably (p=0.5) Bernoulli random variables by designing specific feature extraction algorithms for this purpose [14].

Enrollment: In Fig. 1.1, the Encoding block entails an encoding function $F(.)$, which takes in the enrollment feature vector $\mathbf{A}$ as input. The output of the function F is

- A biometric template $\mathbf{S}$, which is retained by the access control device, and
- An optional key vector $\mathbf{K}$ which can be returned to the user or can also be stored in the cloud, smart card or the access control device.

Thus, the encoding function is given by $F(\mathbf{A}) = (\mathbf{S}, \mathbf{K})$ and is governed by conditional distribution $P_{\mathbf{S},\,\mathbf{K}|\mathbf{A}}$. Such systems are called as *two-factor* systems because both the factors (biometric and the key) are required for authentication. We can even set K to be null. In such a case, where K is set to be null is called as a *one-factor* or a keyless system and does not require a separate key storage (cloud, smart card).

Authentication: During authentication, a legitimate user provides the biometric feature vector **B** and the key **K** in a two-factor system. An attacker, on the other hand, provides a stolen or artificially synthesized biometric feature vector **C** and the stolen or artificially synthesized key **J**. The binary parameter θ indicates presence of the legitimate user or the attacker. The (feature vector, key) pair that is provided during the authentication step is denoted by (**D**,**L**).

$$(\mathbf{D},\ \mathbf{L}) = \begin{cases} (\mathbf{B},\ \mathbf{K}), & \text{if } \theta = 1, \\ (\mathbf{C},\ \mathbf{J}), & \text{if } \theta = 0. \end{cases} \tag{1.1}$$

The authentication decision is computed by the binary-valued decoding function as $\hat{\theta} = g(\mathbf{D},\ \mathbf{L},\ \mathbf{S})$. In a keyless system, the procedure remains the same with **K, J,** and **L** removed from the above description.

To keep the explanation of the system simple, we have considered only a single biometric measurement but this could be similarly extended to include multiple biometric measurements and build a multibiometric secure system. The secure biometric frameworks have been extended to include multiple biometric traits of a user [2, 15–17]. In [15] face and fingerprint templates are concatenated to form a single binary string and this concatenated string is used as input to a secure sketch scheme. Kelkboom *et al.* [18] provided results for decision-level, feature-level, and score-level fusion of templates by using the number of errors corrected in a biometric cryptosystem as a measure of the matching score. Nagar *et al.* [2] developed a multimodal cryptosystem based on feature-level fusion using two different security architectures, *fuzzy commitment* and *fuzzy vault.*

1.0.2 Issues with the Secure Biometric System Model

One important issue for multimodal systems is that the multiple biometric traits generally do not have the same feature-level representation. Furthermore, it is difficult to characterize multiple biometric traits using compatible feature-level representations, as required by a template protection scheme [2]. To counter this issue there have been many fusion techniques for combining multiple biometrics [2, 15, 16]. One possible approach is to apply a separate template protection scheme for each trait followed by decision-level fusion. However, such an approach may not be highly secure, since it is limited by the security of the individual traits. This issue motivated our proposed approach of using multimodal biometric security to perform a joint feature-level fusion and classification.

Another important issue is that biometric cryptosystem schemes are usually implemented using ECC. In order to apply ECC, the biometric feature vectors must be quantized, for instance by binarizing. One method of binarizing the feature vectors is thresholding the feature vectors, for example, by thresholding against the population mean or thresholding against zero. However, thresholding causes a quantization loss and does not preserve the semantic properties of the data structure in Hamming space. In order to avoid thresholding and minimize the quantization loss, we have used the idea of *hashing* [19, 20], which is used in the image and data retrieval literature to achieve fast search by binarizing the real-valued image features. The basic idea of hashing is to map each visual object into a compact binary feature vector that approximately preserves the data structure in the original space. A key principle of hashing is to map visually similar samples to similar binary codes. Owing to its storage and retrieval efficiency, hashing has been extensively used in approximate nearest neighbor search for large scale visual search and image retrieval.

1.0.3 Deep Hashing

Deep structured learning, or more commonly called deep learning, has emerged as a new area of machine learning. Deep learning is being extensively applied to solve problems that have resisted the best attempts of the machine learning and artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business, and government.

Recent progress in image classification, object detection, face recognition, speech recognition and many other computer vision tasks demonstrates the impressive learning ability of *convolutional neural networks (CNNs)* [21–29]. The robustness of features generated by the CNNs has led to a surge in the application of deep learning for generating binary codes from raw image data [30, 31]. Deep hashing [32–35] is the technique of integrating hashing and deep learning to generate compact binary vectors from raw image data. Xia *et al.* [32] adopted a two-stage learning strategy wherein the first stage computes hash codes from the pairwise similarity matrix and the second stage trains a deep neural network to fit the hash codes generated in the first stage. The model proposed by Lai *et al.* [33] simultaneously captures the intermediate image features and trains the hashing function in a joint learning process. The hash function in [33] uses a divide-and-encode module, which splits the image features derived from the deep network into multiple blocks, each block encoded into one hash bit. Liu *et al.* [35] present a deep hashing model that learns the hash codes by simultaneously optimizing a contrastive loss function for input image pairs and imposing a regularization on the real-valued outputs to approximate the binary values. Zhu *et al.* [36] proposed a deep hashing method to learn hash codes by optimizing a pairwise cross-entropy loss and a quantization loss to preserve the pairwise similarity and minimize the quantization error simultaneously.

There also exist methods (eg., [37, 38]) which adopt deep learning for cross-modal hashing (CMH) and are known as deep cross-modal hashing (DCMH) methods. DCMH techniques give improved performance over other CMH techniques which use handcrafted features [39, 40]. Jiang *et al.* were the first to propose an end-to-end DCMH framework to learn the binary hash codes in DCMH [38]. However, they just utilize the inter-modal relationship ignoring intra-modal information. In contrast, Yang *et al.* exploit this intra-modal information by using pairwise labels to propose Pairwise Relationship Guided Deep Hashing (PRDH) method [37].

In addition to the regular DCMH techniques [38, 41, 42], which exploit entropy maximization and quantization losses in the objective function of the DCMH, an ECC decoder can also be used as an additional component to compensate for the heterogeneity gap and reduce the distance between the different modalities of the same subject in the Hamming space in order to improve the CMH and retrieval efficiency. We presume that the hash code generated by DCMH is a binary vector that is within a certain distance from a codeword of an ECC. The hash code generated by DCMH can be passed through an ECC decoder, then the closest codeword to this hash code is found. which can be used as a final hash code for the retrieval process.

1.0.4 Neural Error Correcting Decoder

Recent work has shown that the same kinds of neural network architectures used for classification can also be used to decode ECC [43–45]. Motivated by this, we have used a neural error-correction decoder (NECD) [43] as an ECC decoder to improve the cross-modal retrieval efficiency. In [43], the belief propagation (BP) algorithm, which is commonly used for decoding of low-density parity check (LDPC) codes, is formulated as a neural network and it is shown that a weighted BP decoder implemented using deep learning methods can improve the BP decoding of codes by 0.9 dB in the high signal-to-noise ratio (SNR) region. Later, Lugosch *et al.* [45] proposed a neural-network architecture with reduced complexity by leveraging offset min-sum algorithm and achieved similar results to [43]. Gruber *et al.* [46] used a fully connected architecture to propose a neural network decoder that gives performance close to a maximum likelihood (ML) decoder for very small block codes. Additionally, in [47], a communication system has been formulated as an autoencoder for small block codes.

1.1 Outline and Contributions

Based on the above few technical concepts, we have developed a novel system that integrates deep hashing with neural error-correction decoder for biometric security and biometric retrieval applications. Below is the summary of the contributions described in this dissertation. All of the contributions listed have either been peer-reviewed or are under review at the time of this writing. The chapters where each is developed are listed. The specific contributions are:

1. In Chapter 2, a novel architecture that integrates a deep hashing framework with a

neural network decoder (NND) for application to face template protection is presented [48]. It improves upon existing face template protection techniques to provide better matching performance with one-shot and multi-shot enrollment.

2. In Chapter 3, a deep learning framework is developed for feature-level fusion of each user's multiple biometrics to generate a secure multimodal template [49, 50]. A deep hashing (binarization) technique is integrated with the fusion architecture to generate a robust binary multimodal shared latent representation. Additionally, a hybrid secure architecture is developed by combining cancelable biometrics with secure sketch techniques, and it is integrated with the deep hashing framework and make it computationally prohibitive to forge a combination of multiple biometrics that passes the authentication.

3. Continuing the multibiometric model, in Chapter 4, a novel multimodal deep hashing neural decoder (MDHND) architecture is developed, which integrates a deep hashing framework with a NND to create an effective multibiometric authentication system [51]. The MDHND consists of two separate modules: a MDH module, which is used for feature-level fusion and binarization of multiple biometrics, and a NND module, which is used to refine the intermediate binary codes generated by the MDH and compensate for the difference between enrollment and probe biometrics (variations in pose, illumination, etc.).

4. Chapter 5 presents a novel CMH architecture — deep neural decoder cross-modal hashing (DNDCMH), which uses a binary vector specifying the presence of certain facial attributes as an input query to retrieve relevant face images from the database [52–54]. The DNDCMH network consists of two separate components: an attribute-based deep cross-modal hashing (ADCMH) module, which uses a margin (m)-based loss function to efficiently learn compact binary codes to preserve similarity between modalities (i.e., facial attribute modality and image modality) in the Hamming space, and a NECD, which is an error-correcting decoder implemented with a neural network.

5. Chapter 6 introduces a novel attribute-guided cross-resolution (low-resolution to high-

resolution) face recognition framework that leverages a cpGAN structure with adversarial training to find the hidden relationship between the low-resolution and high-resolution images in a latent common embedding subspace [55]. The coupled generative adversarial network (GAN) framework consists of two sub-networks, one dedicated to the low-resolution domain and the other dedicated to the high-resolution domain.

6. In Chapter 7, a framework similar to the cpGAN structure used in chapter 6 has been developed for a profile-to-frontal face recognition system [56]. The performance of this cpGAN architecture for profile-to-frontal face recognition system has been evaluated and compared with a coupled convolutional neural network (cpCNN) and an adversarial discriminative domain adaptation network (ADDA).

Chapter 2

Zero-Shot Deep Hashing and Neural Network Based Error Correction for Face Template Protection

In this chapter, we present a novel architecture that integrates a deep hashing framework with a NND for application to face template protection. It improves upon existing face template protection techniques to provide better matching performance with one-shot and multi-shot enrollment. A key novelty of our proposed architecture is that the framework can also be used with zero-shot enrollment. This implies that our architecture does not need to be re-trained even if a new subject is to be enrolled into the system.

2.1 Introduction

The leakage of biometric information, such as a stored template, to an adversary constitutes a serious threat to security and privacy because if an adversary gains access to a biometric database, he can potentially obtain the stored user information [53, 57–59]. The attacker can use this information to gain unauthorized access to a system, abuse the biometric information for unintended purposes, and violate user privacy [2,60,61]. Hence, biometric template protection is an important issue and the main focus of this chapter.

For a template to be secure, it must satisfy the important properties of *noninvertibility* and *cancelability*. Noninvertibility implies that it must be computationally difficult to recover the original biometric data when a template is given (e.g., compromised). Cancelability implies that if a template gets compromised, it should be possible to revoke the compromised template and generate a new template using a different transformation.

Prior work in face template protection has tried to decrease the intra-user variability and increase the inter-user variability by using multiple acquisitions of the user's biometric trait (multi-shot) during enrollment [62–64]. Pandey *et al.* [63] provide a face template protection algorithm, where unique maximum entropy binary (MEB) codes are assigned to each user and these MEB codes are used as labels to train a CNN and learn the mapping from face images to MEB codes. The MEB code assigned to each user is cryptographically hashed and stored as a template in the database. This algorithm [63] suffers from a high FRR for higher matching accuracy and moreover, it is only compatible with multi-shot enrollment. To improve upon this algorithm, Jindal *et al.* [64] use a deeper and better CNN for robust mapping of face images to binary codes with significantly better matching performance and compatibility with both one-shot and multi-shot enrollment.

In *one-shot* enrollment, strictly only one image is used during enrollment and training, while in *multi-shot* enrollment, multiple images are used during enrollment and training of the network. However, both the deep learning based methods [63, 64] are not compatible with *zero-shot* enrollment, wherein a subject not seen during training needs to be enrolled into the system. For both of the above cited methods, whenever a new subject needs to be enrolled, the complete network needs to be retrained with the new subject included into the training database.

To address the above problems, we propose an architecture for face template protection by integrating a DH framework with a NND. Deep hashing is the application of deep learning to generate compact binary vectors from raw image data and is generally used for fast image retrieval [34–36, 52, 65–70]. In addition to using deep hashing to generate binary codes, we use ECC as an additional component. ECC is used to compensate for the difference in enrollment and probe biometrics (arising from variation in pose, illumination, noise in biometric capture). In this work, we integrate a NND [43] into our deep hashing architecture as an ECC component to improve the matching performance.

Specifically, our proposed architecture consists of two major components: a DH component, which is used for robust mapping of face images to their intermediate binary codes, and a NND component, which corrects errors in the intermediate binary codes that are caused by differences in the enrollment and probe biometrics due to factors such as variation in pose, illumination, and other factors. The final binary code generated by the NND component is then cryptographically hashed and stored as the secure face template in the database. We have used SHA3-512 as the cryptographic hash function, since it is a current standard for string-based passwords and provides strong security. The template generated after cryptographic hashing has no correlation with the binary codes generated at the output of the NND. To improve the template security, we have also optimized our deep hashing architecture by using an additional loss function to maximize the entropy of the binary codes being generated, which also helps to minimize the intra-user variability and maximize the inter-user variability.

The advantage of using a NND instead of a conventional ECC decoder is that implementing the decoder as a neural network provides the benefit of using a similar architecture as the DH component that generates the binary code, and hence, it can be more efficiently jointly optimized and implemented within a common framework. Another advantage of using the NND is that it provides an opportunity to jointly learn and optimize with respect to biometric datasets, which are not necessarily characterized by Gaussian noise as is assumed by a conventional decoder. Motivated by this, in this chapter, we have integrated our DH network with the NND by using a joint optimization process to formulate our novel face template protection architecture. This proposed architecture can also be used with zero-shot enrollment while still offering the potential to improve the matching performance with one-shot and multi-shot enrollment.

2.2 Proposed Architecture

In this section, we present a system overview of the proposed architecture, which is shown in Fig. 2.1, and also present the enrollment and authentication procedure. The architecture consists of two important components: a deep hashing component and a neural

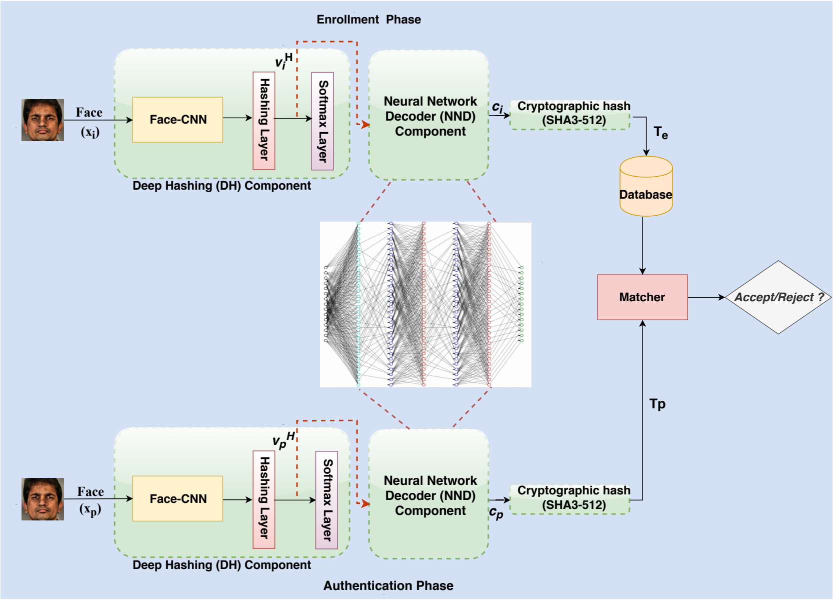

Enrollment Phase
Face
(x_i)
Face-CNN
Hashing Layer
v_i^H
Softmax Layer
Deep Hashing (DH) Component
Neural Network Decoder (NND) Component
c_i
Cryptographic hash (SHA3-512)
T_e
Database
Matcher
Accept/Reject ?
Tp
Face
(x_p)
Face-CNN
Hashing Layer
v_p^H
Softmax Layer
Neural Network Decoder (NND) Component
c_p
Cryptographic hash (SHA3-512)
Deep Hashing (DH) Component
Authentication Phase

Figure 2.1: Block diagram of the proposed system. The NND is shown in the figure. The DH is shown in Fig. 2.2

network decoder component. In this chapter, we have implemented this architecture for face biometrics. However this system could also be extended for use with other biometrics such as iris or fingerprint or a combination of multiple biometrics.

2.2.1 Deep Hashing Component

The main function of the deep hashing (DH) component (which can interchangeably be called a deep hashing network) is to map the input facial images to binary codes. These binary codes are not pre-defined as in [63, 64] but rather are generated as an output of the DH component. This is one of the reasons why this framework can be used with zero-shot enrollment as well. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ denote N facial images and $\mathbf{Y} = \{\mathbf{y}_{\mathrm{i}} \in \{0, 1\}^M\}_{i=1}^N$ be their associated label vectors, where M denotes the number of class labels. Basically, $\mathbf{y}_{\mathrm{i}}$ is an M-dimensional binary vector and $\mathbf{Y}$ is a matrix formed by such M-dimensional N vectors. An entry of the vector $\mathbf{y}_{\mathrm{i}}$ is 1 if an image $\mathbf{x}_{\mathrm{i}}$ belongs to the corresponding class and 0 otherwise. The goal of the DH component is to learn a mapping or a hash function $\mathcal{G} : \mathbf{X} \longrightarrow \{0, 1\}^{K \times N}$, which maps a set of images to their K-bit binary codes $\mathbf{C} = \{\mathbf{c}_{\mathrm{i}}\} \in \{0, 1\}^{K \times N}$, while preserving the semantic similarity among image data. Specifically, in the DH component, we

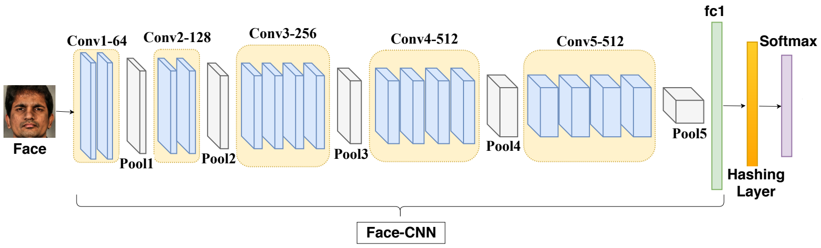


Figure 2.2: Proposed Deep Hashing network.

deploy a supervised hashing algorithm that exploits semantic labels to create binary codes with the following required properties: (1) The semantic similarity between image labels is preserved in the binary codes; images that share common class labels are mapped to same (or close) binary codes. (2) The bits in a code are evenly distributed, which means that the value of each bit is equally likely to be 0 or 1, leading to high entropy and discriminative binary codes.

With recent advances in deep learning, hash functions can be constructed using a CNN that is capable of learning semantic representations from input images. Our approach is built on the existing deep VGG-19 model [22]. The advantage of our approach is that it can be implemented using other deep models as well, such as AlexNet [21]. We introduce our approach based on VGG-19. For our architecture, we use only the first 16 convolutional layers of pre-trained VGG-19. To the 16 convolutional layers, we add our own fully connected layer *fc1* and an output softmax layer. The length of the *fc1* layer depends upon the size of the binary code K that we want to use for template generation. These 16 convolutional layers and the 1 fully connected layer *fc1* will be termed as “Face-CNN”.

The output of the Face-CNN at *fc1* is a feature vector of unquantized values. This output at *fc1* can be directly binarized by thresholding at any numerical value or thresholding at the population mean. However, this kind of thresholding leads to a quantization loss, which results in sub-optimal binary codes. To account for this quantization loss and incorporate the deep representations into the hash function learning, we add a latent layer called hashing layer H with K units to the top of layer *fc1* (i.e., the layer immediately before the output layer), as illustrated in Fig. 2.2. This hashing layer is fully connected to *fc1* and uses the sigmoid activation function so that the outputs are between 0 and 1. The main purpose of the hashing layer is to capture the quantization loss incurred while converting the extracted face features (output of *fc1* layer) into binary codes.

Let $\mathbf{W}^{(H)} \in \mathbb{R}^{d \times K}$ denote the weights between *fc1* and the latent layer. For a given image $\mathbf{x}_i$ with the feature vector $\mathbf{v}_i^{(fc1)} \in \mathbb{R}^d$ in layer *fc1*, the activations of the units in H is given as $\mathbf{v}_i^{(H)} = \sigma(\mathbf{v}_i^{(fc1)}\mathbf{W}^{(H)} + b^{(H)})$, where $\mathbf{v}_i^{(H)}$ is a K-dimensional vector, $b^{(H)}$ is the bias term and $\sigma(.)$ is the sigmoid activation defined as $\sigma(z) = 1/(1+\exp(-z))$, with z a real value.

The combination of the Face-CNN, the hashing layer and the output softmax layer forms the DH network. The details about the training of the DH network are given in Sec. 2.3.1

2.2.2 Neural Network Decoder Component

After training the DH network, we can directly binarize the output of the hashing layer using a threshold of 0.5 and use the result as a binary code. Henceforth, we will refer to the output of the hashing layer from the DH component as an *intermediate binary code*. At this stage, we can cryptographically hash the intermediate binary code and store it as a secure template. However, cryptographic hashes are extremely sensitive to noise and there is always some inherent noise and distortion in biometric measurements such as variations in pose, illumination, or noise due to the biometric capturing device, leading to differences in enrollment and probe biometrics from the same subject. Due to these differences, enrollment and probe biometrics from the same subject may lead to different intermediate binary codes at the output of the hashing layer, which may result in the cryptographic hash mismatch and excessive false rejections. We need to compensate for this distortion in biometric measurements to make the system more robust, improve the matching performance, and provide another layer of security. This is achieved by using ECC. ECC can compensate for the biometric distortion by forcing the enrollment and probe biometric to decode to the same message, making the system more robust to noise in the biometric measurements, which helps in improving the matching performance.

While we could use a conventional ECC decoder at the output of the hashing layer, such a decoder is only suitable when the codewords are corrupted by Gaussian noise. Generally, the distortion in biometric measurements may not necessarily be characterized by Gaussian noise. For this reason, we need to be able to train our ECC decoder to be optimized for biometric measurements.

Recent research in the field of ECC has focused on designing a neural network architecture as an ECC decoder [43, 45]. We can adapt such a neural network based ECC decoder, train it, and use it as an ancillary component to refine the intermediate binary codes generated by hashing layer in the DH component. The advantage of using NND instead of a conventional decoder is that it allows for a common architectural framework to be used for both the hashing framework (i.e., the DH) and the decoder, and it provides an opportunity to jointly learn and optimize with respect to biometric datasets, which are not necessarily characterized simply by Gaussian noise as is assumed by conventional decoders. For our application, we have chosen the NND described in [43] as the basis for our neural network ECC decoder to be integrated with the DH component. Due to space limitations, we do not provide full details on the operation of NND, as details can be found in the original paper. Rather, we focus our discussion on how the NND decoder is integrated with our architecture and also discuss the differences in training the NND relative to [43].

2.2.3 Enrollment and Authentication

During enrollment, the facial image of the user is captured, resized and is given as input ($\mathbf{x}_\mathrm{i}$) to the trained DH network as shown in Fig. 2.1. The intermediate binary code $\mathbf{v}_\mathrm{i}^{\mathbf{(H)}}$ for the user is generated at the output of the hashing layer of the DH network. This intermediate binary code is fed through a trained NND network and the final binary code $\mathbf{c}_\mathrm{i}$ for a given user is generated by simple thresholding of the output of NND at 0.5. The final binary code is cryptographically hashed using SHA3-512 to create the enrollment face template $\mathbf{T}_\mathrm{e}$ to be stored in the database. The final binary code is not provided to the user or stored in an unprotected form. Only the cryptographic hash of the final binary code is stored as a template in the database. During the authentication phase, a new sample of the enrolled user is fed through the DH network and the output of the hashing layer is fed through the NND to get the final binary code $\mathbf{c}_\mathrm{p}$ for the probe. This final binary code is cryptographically hashed using SHA3-512 to generate the probe template $\mathbf{T}_{\mathrm{p}}$, which is compared with the enrollment template $\mathbf{T}_{\mathrm{e}}$ in the matcher to generate a binary score of accept/reject nature.

2.3 Training of the Proposed Architecture

The architecture described in Sec. 2.2 is trained in three stages. In Stage 1, we use a novel loss function to train and learn the parameters of the DH component to generate intermediate binary codes at the output of the hashing layer; in Stage 2, the intermediate binary codes from Stage 1 are passed through a conventional ECC decoder to generate the ground truth, which will be used to fine-tune a NND; in Stage 3, the NND decoder is trained using the ground truth from Stage 2 and this NND is then integrated with the DH component followed by a joint optimization of the overall system.

2.3.1 Stage 1: Training the DH component

As discussed in the Sec. 2.2, the DH component consists of Face-CNN (*Conv1-Conv5+fc1*), hashing layer H and the output softmax layer. Before adding the hashing layer to the DH component, the Face-CNN with the output softmax layer is trained with the CASIA-Webface [71], which contains 494,414 facial images corresponding to 10,575 subjects. For training the Face-CNN, all the raw facial images are first aligned in 2-D and cropped to a size of 224×224 before passing through the network [72]. The only other pre-processing is subtracting the mean RGB value, computed on the training set, from each pixel. The training is carried out by optimizing the multinomial logistic regression objective using mini-batch gradient descent with momentum. The number of nodes in the fully connected layer *fc1* before the softmax layer depends upon the size of the binary code K as we want to gradually reduce the size of the feature vector from high dimensions to the required size of the binary code K. Therefore, if K is 255, then the length of *fc1* is 512, and if K is 1023, then the length of *fc1* is 2048.

After the training of Face-CNN, the hashing layer is added on top of the *fc1* layer (i.e. just before the output softmax layer) to form the DH component. The database used for training the DH component have been discussed in detail in Sec. 2.4.1. For training the full DH component, we have used a novel objective function, which helps in reducing the quantization loss and also maximize the entropy to generate optimal and discriminative binary codes at the output of the hashing layer. The objective function used for training the DH component is a combination of classification loss, quantization loss and entropy maximization loss. The classification loss has been added into the DH network by using the *softmax* layer as shown in Fig. 2.2. $E_1(\mathbf{w})$ denotes the objective function required to fulfill the classification task:

$$E_1(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} L_i(f(x_i, \mathbf{w}), y_i) + \lambda ||\mathbf{w}||_2^2, \tag{2.1}$$

where the first term $L_i(.)$ is the classification loss for a training instance i and is described below, N is the number of training images in a mini-batch. $f(x_i, \mathbf{w})$ is the predicted softmax output of the network and is a function of the input training image x_i and the weights of the network $\mathbf{w}$. The second term is the regularization function where λ governs the relative importance of the regularization. Let the predicted softmax output $f(x_i, \mathbf{w})$ be denoted by $\hat{y}_i$. The classification loss for the i^{th} training instance is given as:

$$L_i(\hat{y}_i, y_i) = -\sum_{m=1}^{M} y_{i,m} \ln \hat{y}_{i,m}, \tag{2.2}$$

where $y_{i,m}$ and $\hat{y}_{i,m}$ is the ground truth and the prediction result for the m^{th} class (out of total M classes) for the i^{th} training instance, respectively and M is the number of output units.

The output of the hashing layer is a K-dimensional vector denoted by $\mathbf{v}_i^{(H)}$, corresponding to the i-th input image. The n-th element of this vector is denoted by $v_{i,n}^{(H)} (n = 1, 2, 3, \cdots, K)$. The value of $v_{i,n}^{(H)}$ is in the range of $[0, 1]$ because it has been activated by the sign activation. To capture the quantization loss of thresholding at 0.5 and make the codes closer to either 0 or 1, we add a constraint of binarization loss between the hashing layer activations and 0.5, which is given by $\sum_{i=1}^{N} ||\mathbf{v}_i^{(H)} - 0.5\mathbf{e}||^2$, where N is the number of training images in a mini-batch and $\mathbf{e}$ is the K-dimensional vector with all elements equal to 1. Let $E_2(\mathbf{w})$ denote this constraint to boost the activations of the units in the hashing layer to be closer to 0 or 1 and this constraint needs to be maximized in order to push the binary codes closer to 0 and 1:

$$E_2(\mathbf{w}) = -\frac{1}{K}\sum_{i=1}^{N} ||\mathbf{v}_i^{(H)} - 0.5\mathbf{e}||^2, \tag{2.3}$$

In the state-of-the-art face template protection methods [63, 64], pre-defined maximum entropy binary codes have been used as labels to train the deep CNNs. Maximum entropy an important requirement for improving the discrimination as well as improving the security. To include this requirement into our architecture, it is important that the binary codes at the output of the hashing layer have equal number of 0's and 1's, which maximizes the binary entropy of the discrete distribution and results in binary codes with better discrimination. It provides more discrimination because the codewords are further apart in the Hamming space from one another compared to other choices of codewords. Let $E_3(\mathbf{w})$ denote the loss function that forces the output of each node to have a 50% chance of being 0 or 1; $E_3(\mathbf{w})$ needs to be minimized to maximize the entropy:

$$E_3(\mathbf{w}) = \sum_{i=1}^{N}(\text{mean}(\mathbf{v}_i^{(H)}) - 0.5)^2. \tag{2.4}$$

The overall objective function to be minimized for a semantics-preserving efficient binary codes is given as:

$$\alpha E_1(\mathbf{w}) + \beta E_2(\mathbf{w}) + \gamma E_3(\mathbf{w}), \tag{2.5}$$

where α, β, and γ are the tuning parameters of each term.

The objective function given in (2.5) can be minimized by using stochastic gradient descent (SGD) efficiently by dividing the training samples into batches.

2.3.2 Stage 2: Generating the Ground Truth for Training the Neural Network Decoder

As already mentioned, we have used the NND from [43] as our ECC component for added security and also to make the system robust to variations in biometric measurements, which would make the architecture applicable even for zero-shot enrollment. The NND in [43] is optimized for a Gaussian noise channel and the database used for training the NND reflects various channel output realizations when the zero codeword has been transmitted.

However, for our proposed system, the NND needs to be optimized for use with biometric data, where the channel noise is characterized by the image distortions (e.g., pose variations, illumination variations and noise due to biometric capturing device) in different biometric images of the same subject. We can create the input dataset for training the NND by using the different facial images for the same subject. However, there is a major issue that needs to be addressed for training the NND with the facial images, and that is we do not have the labels or the ground truth codewords that these input images need to be mapped to. An external conventional ECC decoder can be used to generate the ground truth codewords for this input dataset of facial images.

After training DH network in Stage 1, a number of facial images of subjects disjoint from the subjects used for training the DH network are used for generating the ground truth for training the NND. First, we use the trained DH network to extract the binary vectors at the output of the hashing layer (threshold the sigmoid activations at 0.5) for this disjoint dataset. These extracted binary vectors are used as input to a conventional ECC decoder for soft-decision decoding. Hard-limiting the output of the ECC decoder generates ground truth codewords that are used as labels for optimizing the NND in Stage 3. While usually all the binary vectors of a given subject are mapped by the decoder to the same codeword, it is possible that the vectors could be mapped to different codewords. This is especially the case when there are substantial differences in the variations of the facial images for a given subject, or when the ECC code is not sufficiently strong. In the case that the input binary vectors get mapped to a plurality of codewords, the most common of these codewords is used as the ground truth for that subject.

2.3.3 Stage 3: Joint Optimization of Deep Hashing and Neural Network Decoder

In Stage 3 of the training, first we train the NND using the same procedure and database outlined in the original paper [43]. Next, the NND is fine-tuned for our facial biometric data. For fine-tuning the NND, we use the same disjoint dataset that was used for the ECC coventional decoder in Stage 2. Similar to Stage 2, the input to NND is given by the feature vectors generated at the output of the hashing layer of the DH network and the labels are provided by the decoded codewords generated by the conventional ECC decoder in Stage 2. We have used sigmoid activation for the last layer of NND so that the final network output is in the range $[0, 1]$. As the network output layer is activated by sigmoid function, it generates continuous output, which is probability that a particular output component is equal to 1. Due to the continuous output activation function, we cannot use Hamming distance as a loss function. Therefore, we train and fine-tune the NND using binary cross-entropy loss function:

$$L(o, y) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log(o_i) + (1 - y_i)\log(1 - o_i)\right), \tag{2.6}$$

where o_i, y_i are the actual ith component of the NND output and ground truth codeword (label), respectively.

After fine-tuning the NND, we integrate DH and NND by discarding the softmax layer in the DH network and connecting the output of the hashing layer from DH as input to NND to create an end-to-end face template protection architecture. This overall system is then optimized end-to-end using the same dataset as used for fine-tuning NND and also the same cross-entropy loss function given in (2.6).

Indeed a key benefit of using NND over conventional ECC decoding is that the NND can be trained to force all the binary vectors of the same subject to be decoded to the corresponding common codeword. This also helps for zero-shot enrollment, where, even for the subjects not seen during training, the trained NND will generally compensate for the biometric distortion by forcing the enrollment and probe biometric of a subject to decode to a common codeword.

2.4 Implementation and Evaluation

2.4.1 Databases and Data Augmentation

We use the following databases for our training and testing of our proposed system. The first 3 datasets have been used for a fair comparison with the state-of-the-art in face template protection. The last dataset is used to test our system on a modern large scale face recognition dataset:

1. The CMU PIE [73] database consists of 41,368 images of 68 subjects. These images have been taken under 13 different poses, 43 different illumination conditions, and 4 different expressions. We use 5 poses (p05, p07, p09, p27, p29) and all illuminations variations for our training and testing. We have used 50 subjects for training the DH network with the hashing layer. Out of the remaining 18 subjects, 12 subjects have been used for training the NND and fine-tuning the overall system end-to-end, and the remaining 6 subjects (1458 images) have been used for testing of zero shot enrollment. For one shot and multi-shot, the train and test data split is consistent with [64] for the 50 subjects.

2. The extended Yale Face [74] database contains 2,432 images corresponding to 38 subjects with frontal pose and under different illumination conditions. We have used the cropped version of the database as used in [63]. Out of the 38 subjects, we have used 22 subjects for training the DH network with the hashing layer. Out of the remaining 16 subjects, 10 subjects have been used for training the NND and fine-tuning the overall system from end-to-end, and the remaining 6 subjects (400 images) have been used for testing of zero shot enrollment. Again, for multi-shot enrollment, the train and test data split is consistent with [63] for the 22 subjects.

3. For the CMU Multi-PIE [75] database, we have just used it to test the robustness of our overall system to change in session and lighting conditions as in [63]. CMU Multi-PIE database contains 750,000 images corresponding to 337 subjects under 4 different sessions, 15 view points and 19 illumination conditions. As in [63], we have used the session 3 and session 4 with total of 198 common subjects. We have chosen session 3 for enrollment and session 4 for testing. We have used 125 subjects for training the DH network with the hashing layer. Out of the remaining 73 subjects, 53 subjects have been used for training the NND and fine-tuning the overall system from end-to-end, and the remaining 20 subjects have been used for testing of zero shot enrollment.

4. The WVU multimodal dataset[1] was collected at West Virginia University in year 2012 and 2013. The data for the year 2013 and 2012 contain 61,300 and 70,100 facial images in different poses corresponding to 1063 and 1200 subjects, respectively. There are 294 common subjects in year 2012 and 2013 data. The unique 769 (1063-294) subjects from 2013 dataset

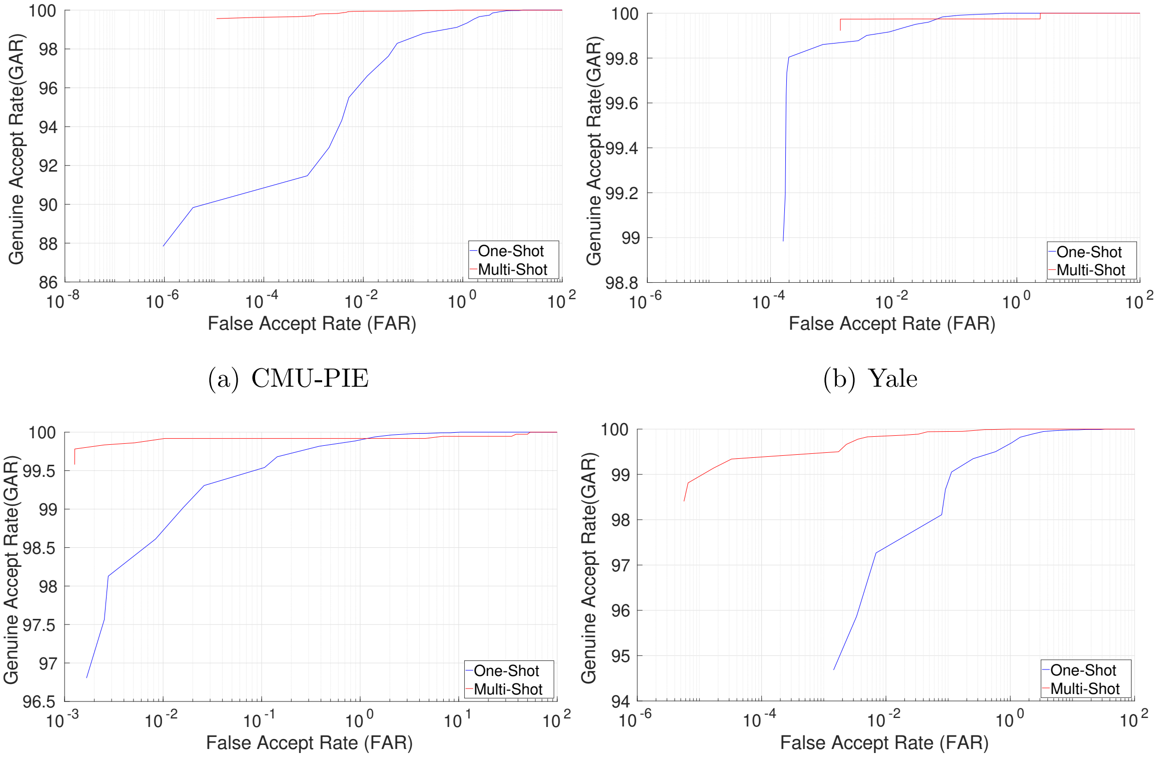
[1] http://biic.wvu.edu

are used for training the DH component. The remaining 294 common subjects from 2013 dataset are used for training the NND and fine-tuning the overall system from end-to-end. For one shot, we randomly select one image per user for training and the rest are used for testing. For multi-shot, 10 images are randomly chosen for training and the rest are used for testing. A subset of 50 subjects of the unique 900 subjects from the 2012 dataset are used for testing of zero shot enrollment.

In order to have sufficient data for training our deep learning algorithm, we use data augmentation [76]. For each facial image, we apply horizontal flip, scaling to $60\%, 70\%, 80\%$, and 90% to generate five augmented images [77,78]. For each augmented image of size $m \times m$ we extract all possible crops of size $n \times n$ yielding a total of $(m-n+1) \times (m-n+1)$ crops. Therefore, data augmentation yields a total of $5 \times (m-n+1) \times (m-n+1)$ images for each face image. For our experiments, we have chosen value of $m = 224$ and $n = 221$.

2.4.2 Details of the Code and Decoder

The intermediate binary code generated at the output of hashing layer of the DH network is considered to be the noisy codeword of some ECC that we can select and this noisy codeword can be decoded using the NND to generate the final hash code that is cryptographically hashed and stored as template in the database. We have used a BCH code as an ECC for our experiments. The size of the BCH code that we can use for NND depends upon the size of the intermediate binary code that we want. For comparison with the state-of-the-art methods in [63, 64], we have used 255 and 1023 as the size of the intermediate binary codes, which is also the size of the final binary code used. The BCH codes that we have used for NND are BCH(255,187), and BCH(1023,933). A code used for decoder is normally described in terms of (N, k), where N signifies the codeword length (i.e. input to the decoder), and k signifies the actual message size (output of decoder). Therefore in our case, if $N = 255$, $k = 187$, and if $N = 1023$, $k = 933$. For the external ECC decoder, we have used the BCH decoder from the communication toolbox in MATLAB®.



(a) CMU-PIE

(b) Yale

(c) Multi-PIE

(d) WVU

Figure 2.3: ROC curves with One-Shot and Multi-Shot enrollment for different datasets for K=255. GAR and FAR are reported in %.

2.4.3 Experimental Set Up and Results

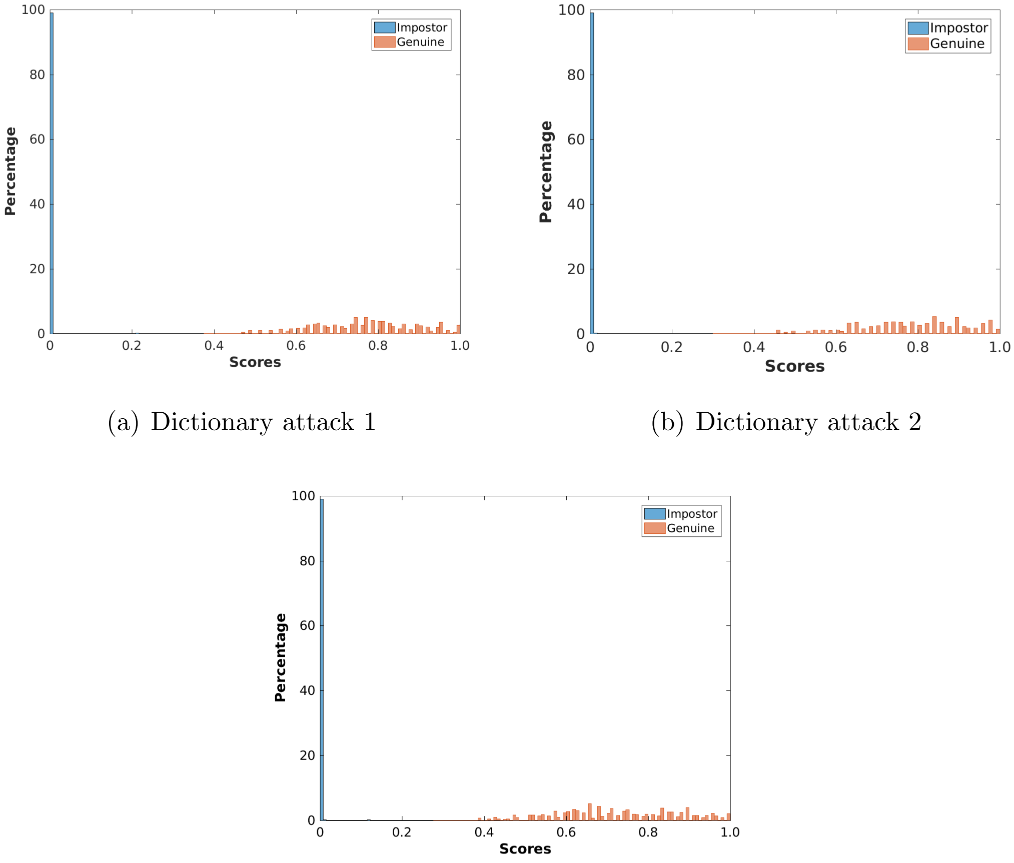
We use the genuine accept rate (GAR) at different FAR as the evaluation metric and we also report the equal error rate (EER). Since the train-test splits are randomly generated, we report the mean and standard deviation of the results for 5 random splits.

As discussed in Sec. 2.2.3, authentication is based on binary accept/reject by comparing the probe template $\mathbf{T}_{\mathrm{p}}$ with the enrolled template $\mathbf{T}_{\mathrm{e}}$. However, this is not an ideal scenario for an experimental testing of biometric authentication system as we need a tunable metric to adjust the FAR and FRR of the system. For this reason, we use several augmented images (as described in Sec. 2.4.1) of each image presented for authentication, and $\mathbf{T}_{\mathrm{p}}$ is calculated for each augmented image, yielding a set of templates $\mathcal{T}$. Therefore, as given in [63], the final matching score can be defined by the number of templates $\mathbf{T}_{\mathrm{p}}$ in $\mathcal{T}$ that match the stored template $\mathbf{T}_{\mathrm{e}}$, scaled by the cardinality of $\mathcal{T}$. A threshold can then be applied to the matching score to achieve a desired value of FAR/FRR.

Table 2.1: Authentication results for different datasets.

Database	Enrollment Type	K	GAR@0.01%FAR	EER
PIE	Zero-Shot	255	83.6 ± 2.1%	14.71 ± 0.83%
		1023	82.8 ± 1.83%	14.89 ± 0.77%
	One-Shot	255	96.2 ± 0.98%	0.99 ± 0.12%
		1023	96.0 ± 1.12%	1.32 ± 0.40%
	Multi-Shot	255	99.9 ± 0.06%	0.051 ± 0.022%
		1023	99.0 ± 0.88%	0.078 ± 0.016%
Yale	Zero-Shot	255	87.4 ± 1.46%	12.5 ± 0.98%
		1023	85.1 ± 1.98%	14.65 ± 1.23%
	One-Shot	255	99.9 ± 0.03%	0.052 ± 0.023%
		1023	99.1 ± 0.18%	0.072 ± 0.034%
	Multi-Shot	255	99.98 ± 0.005%	0.049 ± 0.015%
		1023	99.8 ± 0.09%	0.039 ± 0.012%
Multi-PIE	Zero-Shot	255	81.4 ± 2.32%	15.43 ± 1.08%
		1023	81.2 ± 2.18%	16.69 ± 1.21%
	One-Shot	255	98.7 ± 0.99%	0.263 ± 0.10%
		1023	97.4 ± 0.94%	0.34 ± 0.13%
	Multi-Shot	255	99.8 ± 0.17%	0.93 ± 0.11%
		1023	98.5 ± 0.43%	1.14 ± 0.13%
WVU Multimodal	Zero-Shot	255	88.7 ± 1.87%	10.32 ± 0.83%
		1023	88.1 ± 1.68%	11.32 ± 0.92%
	One-Shot	255	97.5 ± 0.98%	0.42 ± 0.14%
		1023	97.23 ± 0.89%	0.51 ± 0.11%
	Multi-Shot	255	99.7 ± 0.16%	0.11 ± 0.02%
		1023	98.5 ± 0.57%	0.48 ± 0.10%

For our experiments, the mean and standard deviation of the EER, and the GAR at 0.01% FAR for 5 different train-test splits, with zero-shot, one-shot, and multi-shot enrollment, using binary code dimensions $K = 255, 1023$ have been reported in Table 2.1. With zero-shot enrollment, we achieve GARs up to $\approx 83\%$ on CMU-PIE, $\approx 86\%$ on Extended Yale, $\approx 81\%$ on Multi-PIE, and $\approx 88\%$ on Multi-PIE with up to $K = 1023$ at the strict operating point of 0.01% FAR. We also get high GARs in the range of $98 - 99\%$ for one shot and multi-shot on all the datasets. It can be observed from the Table 2.1 that the results are stable with respect to parameter K as there is not drastic change in GAR or the EER as K changes from 255 to 1023. Therefore, this makes the parameter K totally selectable purely on the basis of the required template security. The verification performance using receiver



(a) Dictionary attack 1

(b) Dictionary attack 2

(c) Dictionary attack 3

Figure 2.4: Genuine and impostor distribution of different dictionary attacks for K=255.

operating characteristic (ROC) curves is also shown in Fig. 2.3 for all the datasets using one-shot and multi-shot.

A comparison of our results with other face template protection algorithms on PIE dataset is shown in Table 2.2. Our proposed method values are all shown in bold. For security level, we compare our code dimensionality parameter K to the equivalent parameter in the shown approaches. It can be noted that we get a better matching performance and a lower EER when compared to the other face template protection schemes for both one-shot and multi-shot enrollment. For one shot enrollment, we achieve 96.0%GAR@0.01%FAR for $K = 1023$, which is $\approx$ 4.5% improvement in matching performance compared to 91.34%GAR@0.1%FAR reported in [64]([64] does not report the GAR@0.01%FAR, however, our proposed system still provides better matching performance at even lower FAR). Even with zero-shot enroll-

Table 2.2: Performance comparison for PIE dataset

Method	Enrollment Type	K	GAR@FAR	EER
Hybrid Approach [79]	Multi-Shot	210	90.61%@1%FAR	6.81%
BDA [80]	Multi-Shot	76	96.38%@1%FAR	-
MEB Encoding [63]	Multi-Shot	256	93.22%@0%FAR	1.39%
		1024	90.13%@0%FAR	1.14%
Deep CNN [64]	Multi-Shot	256	97.35%@0%FAR	0.15%
		1024	96.53%@0%FAR	0.35%
	One-Shot	256	91.91%@0.1%FAR	4.00%
		1024	91.34%@0.1%FAR	3.60%
Our Method	Multi-Shot	**255**	**99.9%@0.01%FAR**	**0.051%**
		1023	**99.0%@0.01%FAR**	**0.078%**
	One-Shot	**255**	**96.2%@0.01%FAR**	**0.99%**
		1023	**96.0%@0.01%FAR**	**1.32%**
	Zero-Shot	**255**	**83.6%@0.01%FAR**	**14.71%**
		1023	**82.8%@0.01%FAR**	**14.89%**

Table 2.3: EER comparison with other variations for WVU dataset for 1023 bits

Enrollment Type	Method		
	DH^-	$DH + Decoder$	$DH + NND$(Our)
One-Shot	8.2%	1.83%	0.51%
Multi-Shot	6.61%	1.54%	0.48%

ment, we get good matching performance and a respectable EER.

In order to show the importance of NND, we have compared our complete architecture "$DH + NND$" with two variations of our proposed system:1)Using the deep hashing component with no NND, denoted as "DH^-". 2)DH component with an external conventional decoder, denoted as "DH + $Decoder$". We tested these variations on WVU multimodal dataset for one -shot and multi-shot enrollment for 1023 bits. The EER results are shown in Table 2.3. We can observe $DH + NND$ gives the best result followed by $DH + Decoder$ and lastly DH^-. We can see that using an external decoder improves the performance by reducing the EER by about 5% and using the NND further reduces EER by 1.1%. The advantage with NND over external conventional decoder is that it provides an opportunity to jointly learn and optimize with respect to biometric datasets, which are not necessarily characterized simply by Gaussian noise as is assumed by a conventional decoder.

2.5 Security Analysis

This security analysis is based on a stolen template scenario as reported in [63]. Given the template, the attacker's goal is to extract biometric information of the user. However, the template is generated using a cryptographic hash function SHA-3 512 of the binary codes generated by the deep CNN (DH + NND). Morover, the binary codes generated are neither stored nor provided to the user. Since the SHA-3 512 is a one-way transformation, the attacker will not be able to extract any information about the binary codes from the protected template. The only way the attacker can get the codes is by a brute force attack, where the attacker would need to try all possible values of the code, hash each one and compare them to the template.

There are two scenarios that need to be explored in the case of brute force attack. The first scenario is when the attacker has no access to the CNN parameters and the second scenario is when the attacker has access to the CNN parameters. In the scenario where the attacker has no access to the CNN parameters, the search space for the brute force attack would be 2^K, i.e., the number of binary codes. In this scenario, it is very important that the final binary code should posses high entropy. The objective function required to train the DH component also includes a loss function for maximizing the entropy, which is shown in (2.3). Therefore, the high entropy requirement is captured in the training of the DH component. Additionally, to make the search space larger for a brute force attack, we use a final binary code with a minimum dimensionality of $K = 255$. The brute force attack in this scenario would be computationally infeasible because even with $K = 255$, the search space would be the order of $\binom{255}{128}$ (due to equal number of 0's and 1's).

Now, let's analyze the scenario where the attacker has access to both the stolen face protected template and also the CNN parameters. In this scenario, the attacker would try to generate attacks in the input domain and exploit the FAR of the system. To exploit the system FAR, the attacker would try a dictionary attack using large set of faces. In the proposed method, it is not straightforward to analyze the reduction in the search space due to the knowledge of the CNN parameters. However, measuring the minimal FAR of the proposed method is a good indicator of the template security. To evaluate the template security, we have used the genuine and impostor distribution to measure the false accepts, where all other users other than genuine are considered as impostors. The genuine and impostor distributions for three types of dictionary attacks are shown in Fig. 2.4. The three attacks are:(1) CMU-PIE as the genuine database and Ext Yale as the attacker database. (2) Ext Yale as a genuine database and frontal images of Multi-PIE as attacker database (3) Multi-PIE as the genuine database and Ext Yale as the attacker database. It can be seen that the impostor scores are always zero and genuines tend to one, indicating there are no false accepts in this scenario and the proposed method does not easily accept external faces for enrolled faces even if they are preprocessed under same conditions.

2.6 Summary

In this chapter, we presented an algorithm that uses a combination of deep hashing and neural network based error correction to be implemented for face template protection. The novelty of this algorithm is that it can even be used for zero-shot enrollment, where the subject has not been seen during training of the deep CNN and still can be enrolled. Additionally, we show a matching performance improvement of $\approx 4.5\%$ for one-shot enrollment and $\approx 3\%$ for multi-shot enrollment when compared to related work, while providing high template security.

Chapter 3

Multibiometric Secure System Based on Deep Learning

When compared to unimodal systems, multimodal biometric systems have several advantages, including lower error rate, higher accuracy, and larger population coverage. However, multimodal systems have an increased demand for integrity and privacy because they must store multiple biometric traits associated with each user. In this chapter, we present a deep learning framework for feature-level fusion that generates a secure multimodal template from each user's multiple biometrics. We integrate a deep hashing (binarization) technique into the fusion architecture to generate a robust binary multimodal shared latent representation. Further, we employ a hybrid secure architecture by combining cancelable biometrics with secure sketch techniques and integrate it with a deep hashing framework, which makes it computationally prohibitive to forge a combination of multiple biometrics that passes the authentication. The efficacy of the proposed approach is shown using a multimodal database and it is observed that the matching performance is improved due to the fusion of multiple biometrics.

3.1 Introduction

Multimodal biometric systems use a combination of different biometric traits such as face and iris, or face and fingerprint. Multimodal systems are generally more resistant to spoofing attacks [3]. Moreover, multimodal systems can be made to be more universal than unimodal systems, since the use of multiple modalities can compensate for missing modalities in a small portion of the population. Multimodal systems also have an advantage of lower error rates and higher accuracy when compared to unimodal systems [2]. Consequently, multimodal systems have been deployed in many large scale biometric applications including the FBI's Next Generation Identification (NGI), the Department of Homeland Security's US-VISIT, and the Government of India's Unique Identity (UID). However, Multimodal systems have an increased demand for integrity and privacy because the system stores multiple biometric traits of each user. Hence, multimodal template protection is the main focus of this chapter.

As already pointed out in Chapter 1, there are multiple issues when implementing a multimodal biometric secure system. One issue is compatibility of feature level representations between multiple biometrics and the template protection scheme. Another issue is related to the quantization error when binarizing using thresholding for feature vectors. These both issues could be addressed by using deep hashing methods for developing a multimodal biometric secure system

Inspired by the recent success of deep hashing methods, the objective of this chapter is to examine the feasibility of integrating deep hashing with a secure architecture to generate a secure multimodal template.

The rest of the chapter is organized as follows. The deep hashing secure multibiometric framework and the associated algorithms are introduced in Section 3.2. Implementation details of the proposed framework are presented in Section 3.3. In Section 3.4, we present a performance evaluation of the cancelable biometric module, which is a part of the overall proposed secure multimodal system. The performance evaluation of the overall proposed secure *multimodal* system is discussed in Section 3.5. The conclusions are summarized in Section 3.6.

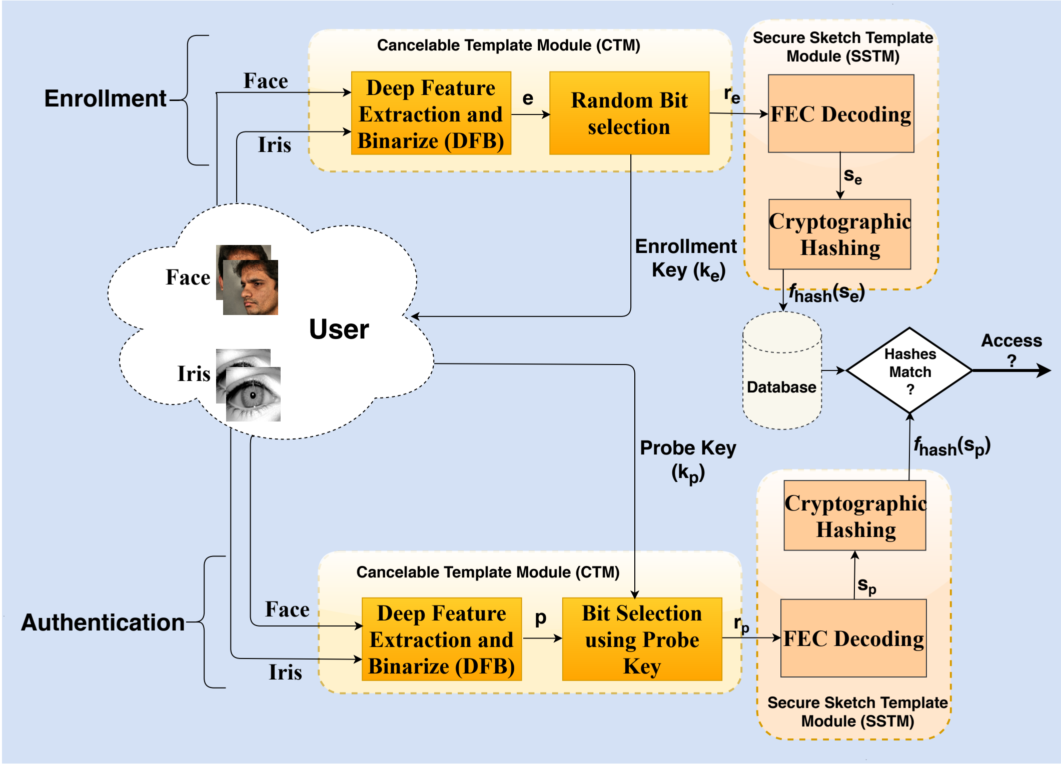


Figure 3.1: Block diagram of the proposed system.

3.2 Proposed Secure Multibiometric System

3.2.1 System Overview

In this section, we present a system overview including descriptions of the enrollment and authentication procedures. We propose a feature-level fusion and hashing framework for the secure multibiometric system. The general framework for the proposed secure multibiometric system is shown in Fig. 3.1. During enrollment, the user provides their biometrics (e.g., face and iris) as an input to the deep feature extraction and binarization (DFB) block. The output of the DFB block is an J-dimensional binarized joint feature vector $\mathbf{e}$. A random selection of feature components (bits) from the binarized joint feature vector $\mathbf{e}$ is performed. The number of random components that are selected from the binarized joint feature vector $\mathbf{e}$ is G. The indices of these randomly selected G components forms the enrollment key $\mathbf{k_e}$, which is given to the user. The cancelable multimodal template $\mathbf{r_e}$ is formed by selecting the values from the binarized joint feature vector $\mathbf{e}$ at the corresponding location or indices as specified by the user-specific key $\mathbf{k_e}$.

This random selection of G components from the binarized joint feature vector $\mathbf{e}$ helps in achieving revocability, because if a key is compromised, a new key can be issued with a different set of random indices. In the next step, $\mathbf{r_e}$ is passed through a FEC decoder to generate the multimodal sketch $\mathbf{s_e}$. The cryptographic hash of this sketch $f_{\mathsf{hash}}(\mathbf{s_e})$ is stored as a secure multimodal template in the database.

During authentication, the probe user presents the biometrics and the key $\mathbf{k_p}$ where $\mathbf{k_p}$ could be same as the enrollment key $\mathbf{k_e}$ in the case of a genuine probe or it could be a synthesized key in case of an impostor probe. Using the biometrics, the probe biometrics are passed through the DFB block to obtain a binary vector $\mathbf{p}$, which is the joint feature vector corresponding to the probe. Using the key $\mathbf{k_p}$ provided by the user, the multimodal probe template $\mathbf{r_p}$ is generated by selecting the values from $\mathbf{p}$ at the locations given by the key $\mathbf{k_p}$. In the next step, $\mathbf{r_p}$ is passed through a FEC decoder with the same code used during enrollment to generate the probe multimodal sketch $\mathbf{s_p}$. If the cryptographic hash of the enrolled sketch $f_{\mathsf{hash}}(\mathbf{s_e})$ matches the cryptographic hash of the probe sketch $f_{\mathsf{hash}}(\mathbf{s_p})$, then the access is granted. If the hash codes do not match, then access is denied.

The proposed secure multibiometric system consists of two basic modules: *cancelable template module (CTM)* and *secure sketch template module (SSTM)*, which are described more fully in the following subsections.

3.2.2 Cancelable Template Module

The CTM consists of two blocks: DFB block and random-bit selection block. The primary function of the CTM is to perform non-linear feature extraction, fusion, and binarization using the proposed DFB architecture shown in Figs. 3.2 and 3.3. The DFB consists of two layers: domain-specific layer (DSL) and joint representation layer (JRL).

Domain-Specific Layer

The DSL consists of a CNN for encoding the face ("Face-CNN") and a CNN for encoding the iris ("Iris-CNN"). For each CNN, we use VGG-19 [22] pre-trained on ImageNet [81] as a starting point and then fine-tune it with an additional fully connected layer *fc3* as

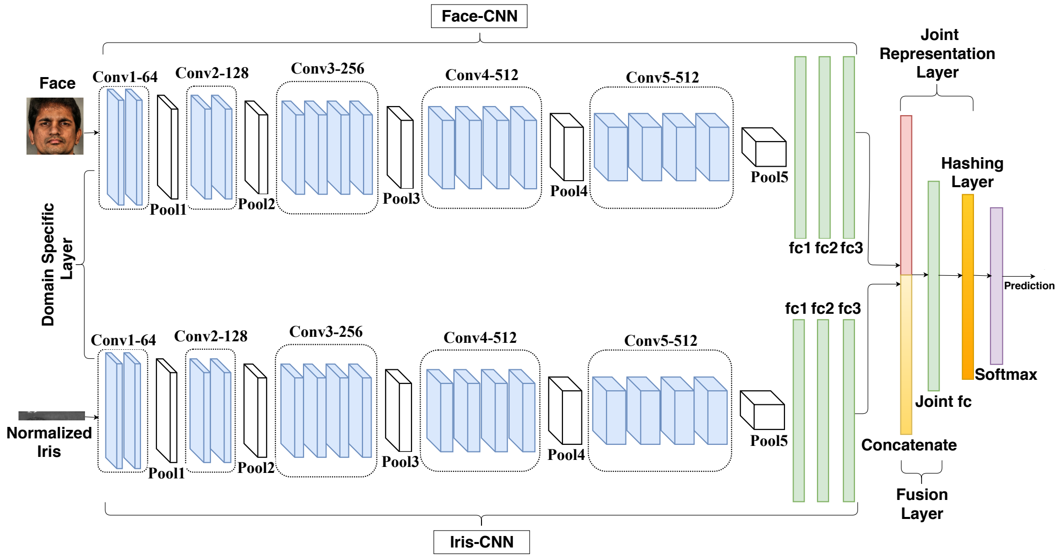


Figure 3.2: Proposed deep feature extraction and binarization (DFB) model for the FCA.

described in Sec. 3.3.2 and 3.3.3. There are multiple reasons for using VGG-19 pre-trained on the ImageNet dataset for encoding the face and iris. In the proposed method, the VGG-19 is only used as feature-extractor for face and iris modalities. It can be seen from the previous literature [82–87] that the features provided by a VGG-19 pre-trained on ImageNet and fine-tuned on face/iris images are very discriminative and therefore can be used for face/iris recognition. Moreover, starting with a well-known architecture and using the same architecture for both modalities makes the work highly reproducible.

Joint Representation Layer

The output feature vectors of the Face-CNN and Iris-CNN are fused and binarized in the JRL, which is split into two sub-layers: fusion layer and hashing layer. The main function of the fusion layer is to fuse the individual face and iris representations from domain-specific layers into a shared multimodal feature embedding. The hashing layer binarizes the shared multimodal feature representation that is generated by the fusion layer.

Fusion layer: We have implemented two different architectures for the fusion layer: (1) FCA, and (2) BLA. These two architectures differ in the way the face and iris feature vectors are fused together to generate the joint feature vector.

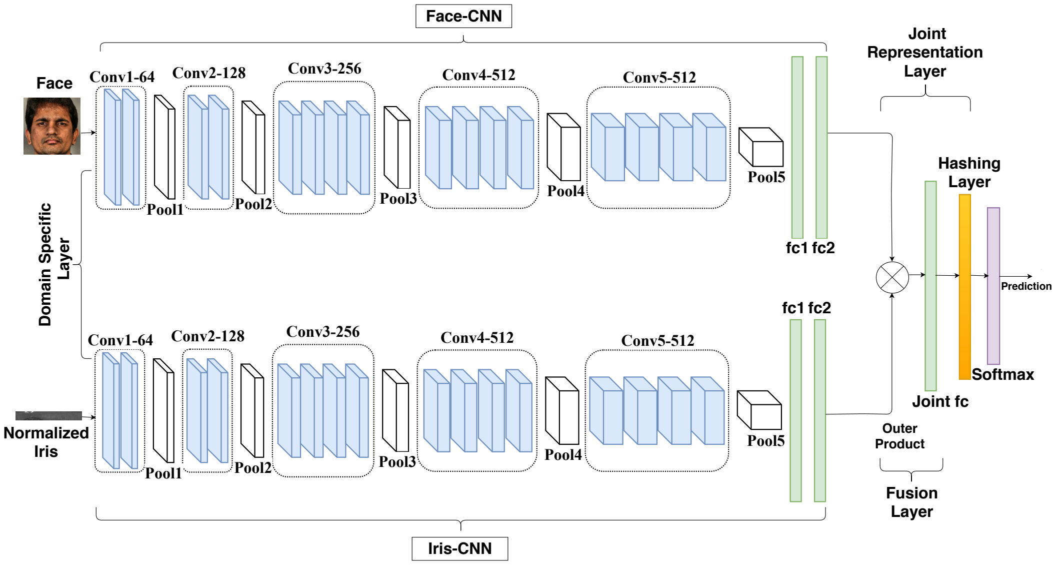


Figure 3.3: Proposed deep feature extraction and binarization (DFB) model for the bilinear architecture (BLA).

In the FCA shown in Fig. 3.2, the outputs of the Face-CNN and Iris-CNN are concatenated vertically using a concatenation layer. The concatenated feature vector is passed through a fully connected layer (hereon known as *joint fully connected layer*) which reduces the feature dimensionality (i.e., the number of dimensions is reduced) and also fuses the iris and face features. In the FCA, the concatenation layer and the joint fully connected layer together constitute the fusion layer.

In the BLA shown in Fig. 3.3, the outputs of the Face-CNN and Iris-CNN are combined using the matrix outer product; i.e., the bilinear feature combination of column face feature vector $\mathbf{f}_{\mathsf{face}}$ and column iris feature vector $\mathbf{f}_{\mathsf{iris}}$ is given by $\mathbf{f}_{\mathsf{face}}\mathbf{f}_{\mathsf{iris}}^T$. Similar to the FCA, the bilinear feature vector is also passed through a joint fully connected layer. In the BLA, the outer product layer and the joint fully connected layer together constitute the fusion layer.

In addition to the two techniques (FCA, BLA) used in this chapter, there could be other fusion techniques for combining multiple modalities [88]. The rationale behind implementing FCA is that we wanted to use a fusion technique that involves just simple concatenation where there is no interaction between the two modalities being fused before the joint fully connected layer (Joint f_c). As evident from Fig. 3.2, the iris and face extracted features do not interact with each other and have their own network parameters before passing through the joint fully connected layer. On the other hand, we also wanted to test a fusion technique that involves high interactions between the two modalities feature vectors at every element before being passed through the joint fully connected layer. That is the reason we have used BLA, which is based on *bilinear fusion* [89]. Bilinear fusion exploits the higher-level dependencies of the modalities being combined by considering the pairwise multiplicative interactions between the modalities at each feature element (i.e., matrix outer product of modalities feature vector). Moreover, bilinear fusion is widely being used in many CNN applications such as fine-grained visual recognition and video action recognition [88, 89].

Hashing layer: The output of the fusion layer produces a J-dimensional shared multimodal feature vector of real values. We can directly binarize the output of the fusion layer by thresholding at any numerical value or thresholding at the population mean. However, this kind of thresholding leads to a quantization loss, which results in sub-optimal binary codes. To account for this quantization loss, we have included another latent layer after the fusion layer, which is known as the hashing layer (shown in orange in Fig. 3.2 and 3.3). The main function of the hashing layer is to binarize (hash) the shared multimodal feature representation generated by the fusion layer.

One key challenge of implementing deep learning to hash end-to-end is converting deep continuous representations, which are real-valued and continuous, to exactly binary codes. The sign activation function $h = \mathrm{sgn}(z)$ can be used by the hashing layer to generate the binary hash codes. However, the use of the non-smooth sign-activation function makes standard back-propagation impracticable as the gradient of the sign function is zero for all non zero inputs. The problem of zero gradient at the hashing layer due to a non-smooth sign activation can be diminished by using the idea of continuation methods [66].

We circumvent the zero-gradient problem by starting with a smooth activation function $y = \tanh(\beta x)$ and making it sharper by increasing the bandwidth β as the training proceeds. We have utilized a key relationship between the sign activation function and the scaled tanh function using limits:

$$\lim_{\beta \to \infty} \tanh(\beta x) = \mathrm{sgn}(x), \tag{3.1}$$

where $\beta > 0$ is a scaling parameter. The scaled function $\tanh(\beta x)$ will become sharper and more saturated as we increase β during training. Eventually, this non-smooth tanh function with $\beta \to \infty$ converges to the original, difficult to optimize, sign activation function. For training the network, we start with a $\tanh(\beta x)$ activation for the hashing layer with $\beta = 1$ and continue training until the network converges to zero loss. We then increase the value of β while holding other training parameters equal to the previously converged network parameters, and start retraining the network for convergence. This process is repeated several times by increasing the bandwidth of the tanh activation as $\beta \to \infty$ until the hashing layer can generate binary codes.

In addition to using this continuation method for training the network, we have used additional cost functions for efficient binary codes. The overall objection function used for training the deep hashing network is discussed in Sec. 3.3.1

Random-Bit Selection

One of the most prevalent methods for generating cancelable template involves random projections of the biometric feature vector [12], in which the random projection is a revocable transformation. Similarly, the DFB architecture is considered to be the projection of the biometric images in a J-dimensional space. The randomness and revocability is added by performing a random bit selection of G bits from the J-dimensional output vector $\mathbf{e}$ of the DFB. After the selection, these random bits are then arranged in descending order of reliability. The reliability of each bit is computed as $((1 - p_g^e)p_i^e)$, where p_i^e and p_g^e are the impostor and genuine bit error probabilities, respectively [2]. A different set of random bits is selected for every user and these randomly selected G bits form the cancelable multimodal template $\mathbf{r_e}$ and the indices of the selected bits forms the key for that user $\mathbf{k_e}$. This key is revocable and a new set of random bits can be selected in case the key gets compromised. Selecting a new set of bits requires that either the original vector $\mathbf{e}$ be retrieved from a secure location or else the user is re-enrolled, thereby presenting a new instance of $\mathbf{e}$. This method of using the DFB architecture with a random bit selection is analogous to a random projection as a revocable transformation to generate a cancelable template [12].

It is important to note that even if multiple users end up having the same key $\mathbf{k_e}$ (i.e., the same set of G random bits), their final templates will still be distinct because the final template depends on the values at those G bits (i.e., $\mathbf{r_e}$) from the enrollment vector $\mathbf{e}$, and not only on the indices of the G bits. The situation when a second user happens to have the same key $\mathbf{k_e}$ is equivalent to the stolen key scenario, which is analyzed in Sec. 3.4.2.

3.2.3 Secure Sketch Template Module

As shown in Fig. 3.1, the cancelable template (output of CTM) $\mathbf{r_e}$ is an intermediate template and is not stored in the database. The cancelable template is passed through the SSTM to generate the secure multimodal template, which is stored in the database. As the name suggests, the SSTM module is related to the secure sketch biometric template protection scheme. The SSTM contains two important blocks: FEC decoding and cryptographic hashing. The main function of the SSTM is to generate a multimodal secure sketch by using the cancelable template as an input to the FEC decoder. This multimodal secure sketch (output of the FEC decoder) is cryptographically hashed to generate the secure multimodal template, which is stored in the database.

The FEC decoding implemented in our framework is the equivalent of a secure-sketch template protection scheme. In a secure-sketch scheme, sketch or helper data is generated from the user's biometrics and this sketch is stored in the access-control database. There are many methods of implementing this secure sketch scheme. However, a common method is to use error control coding. In this method error control coding is applied to the biometrics or the feature vector to generate a sketch which is stored in the database. Similarly, in our proposed framework, the FEC decoding is considered to be the error control coding part required to generate the secure sketch. Our approach is different from other secure sketch approaches using error correcting codes as we do not have to present any other side information to the decoder like a syndrome or a saved message key [14].

The cancelable template $\mathbf{r_e}$ generated from the CTM is considered to be the noisy codeword of some error correcting code that we can select. This noisy codeword is decoded with a FEC decoder and the output of the decoder is the multimodal secure sketch $\mathbf{s_e}$ that corresponds to the codeword closest to the cancelable template. This multimodal sketch $\mathbf{s_e}$ is cryptographically hashed to generate $f_{\mathsf{hash}}(\mathbf{s_e})$, which is stored in the database.

During authentication, the same process is performed. The probe user provides the biometrics and the key which are used to generate the probe template $\mathbf{r_p}$. The probe template $\mathbf{r_p}$ is passed through an FEC decoder for the same error correcting code used during the enrollment. The output of the FEC decoder is the probe multimodal sketch $\mathbf{s_p}$ which is cryptographically hashed and access is granted only if this hash matches the enrolled hash. During authentication, if it is a genuine probe, the enrollment $\mathbf{r_e}$ and the probe vector $\mathbf{r_p}$ would usually decode to the same codeword in which case the hashes would match and access would be granted.

3.3 Implementation

3.3.1 Objective Function for Training the Deep Hashing Network

In this section, the objective function used for training the deep hashing network is described.

Semantics-preserving binary codes: In order to construct semantics-preserving binary codes, we propose to model the relationship between the labels and the binary codes. Every input image is associated with a semantic label, which is derived from the hashing layer's binary-valued outputs, and the classification of each image is dependent on these binary outputs. Consequently, we can ensure that semantically similar images belonging to the same subject are mapped to similar binary codes through an optimization of a loss function defined on the classification error. The classification formulation has been incorporated into the deep hashing framework by adding the *softmax* layer as shown in Fig. 3.2 and Fig. 3.3. Let E_1 denote the objective function required to fulfill the classification formulation:

$$E_1(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} L_n(f(x_n, \mathbf{w}), y_n) + \lambda||\mathbf{w}||^2, \tag{3.2}$$

where the first term $L_n(.)$ is the classification loss for a training instance n and is described below, N is the number of training images in a mini-batch. $f(x_n, \mathbf{w})$ is the predicted softmax output of the network and is a function of the input training image x_n and the weights of the network $\mathbf{w}$. The second term is the regularization function where λ governs the relative importance of the regularization.

The choice of the loss function $L_n(.)$ depends on the application itself. We use a classification loss function that uses softmax outputs by minimizing the cross-entropy error function. Let the predicted softmax output $f(x_n, \mathbf{w})$ be denoted by $\hat{y}_n$. The classification loss for the n^{th} training instance is given as:

$$L_n(\hat{y}_n, y_n) = -\sum_{m=1}^{M} y_{n,m} \ln \hat{y}_{n,m}, \tag{3.3}$$

where $y_{n,m}$ and $\hat{y}_{n,m}$ is the ground truth and the prediction result for the m^{th} unit of the n^{th} training instance, respectively and M is the number of output units.

Additional cost constraints for efficient binary codes: The continuation method that has been described in 3.2.2 forces the activations of the hashing layer closer to -1 and 1. However, we need to include additional cost constraints to obtain more efficient binary codes.

Let the J-dimensional vector output of the hashing layer be denoted by $\mathbf{o}_n^H$ for the n-th input image, and let the i-th element of this vector be denoted by $o_{n,i}^H (i = 1, 2, 3, \cdots, J)$. The value of $o_{n,i}^H$ is in the range of $[-1, 1]$ because it has been activated by the tanh activation. To make the codes closer to either -1 or 1, we add a constraint of maximizing the sum of squared errors between the hashing layer activations and 0, which is given by $\sum_{n=1}^{N} ||\mathbf{o}_n^H - \mathbf{0}||^2$, where N is the number of training images in a mini-batch and $\mathbf{0}$ is the J-dimensional vector with all elements equal to 0. However, this is equivalent to maximizing the square of the length of the vector formed by the hashing layer activations, that is $\sum_{n=1}^{N} ||\mathbf{o}_n^H - \mathbf{0}||^2 = \sum_{n=1}^{N} ||\mathbf{o}_n^H||^2$. Let $E_2(\mathbf{w})$ denote this constraint to boost the activations of the units in the hashing layer to be closer to -1 or 1:

$$E_2(\mathbf{w}) = -\frac{1}{J} \sum_{n=1}^{N} ||\mathbf{o}_n^H||^2. \tag{3.4}$$

In addition to forcing the codes to become binarized, we also require that the codes satisfy a balance property whereby they produce an equal number of -1's and 1's, which maximizes the entropy of the discrete distribution and results in hash codes with better discrimination. To achieve the balance property, we want each bit to fire 50% of the time by minimizing the sum of the squared error between the mean of the hashing layer activations and 0. This is given by $\sum_{n=1}^{N}(\text{mean}(\mathbf{o}_n^H) - 0)^2$, which is equivalent to $\sum_{n=1}^{N}(\text{mean}(\mathbf{o}_n^H))^2$ where mean(.) computes the average of the elements of the vector. This criterion helps to obtain binary codes with an equal number of -1's and 1's. Let $E_3(\mathbf{w})$ denote this constraint that forces the output of each node to have a 50% chance of being -1 or 1:

$$E_3(\mathbf{w}) = \sum_{n=1}^{N}(\text{mean}(\mathbf{o}_n^H))^2. \tag{3.5}$$

Combining the above two constraints (binarizing and balance property constraints) makes $\mathbf{o}_n^H$ close to a length J binary string with a 50% chance of each bit being -1 or 1.

Overall objective function: The overall objective function to be minimized for a semantics-preserving efficient binary codes is given as:

$$\alpha E_1(\mathbf{w}) + \beta E_2(\mathbf{w}) + \gamma E_3(\mathbf{w}), \tag{3.6}$$

where α, β, and γ are the tuning parameters of each term. The optimization to be performed to minimize the overall objective function is given as:

$$\mathbf{w} = \arg\min_{\mathbf{w}}(\alpha E_1(\mathbf{w}) + \beta E_2(\mathbf{w}) + \gamma E_3(\mathbf{w})) \tag{3.7}$$

The optimization given in (3.7) is the sum of the losses form and can be performed via the stochastic gradient descent (SGD) efficiently by dividing the training samples into batches. For training the JRL we adopt a two-step training procedure where we first train only the JRL using the objective function in (3.6) greedily with softmax by freezing the Face-CNN and Iris-CNN. After training the JRL, the entire model is fine-tuned end-to-end using the same objective function with back-propagation at a relatively small learning rate.

For tuning the hyper-parameters α, β, and γ of the objective function (3.6), we have utilized an iterative grid search. To start, consider a cubic grid with all possible values for each parameter. Each point on this grid (α,β,γ) represents a combination of the three hyper-parameters. Because exhaustively searching over all combinations is computationally expensive, we adopted an iterative and adaptive grid search.

In the iterative and adaptive grid search, for each hyper-parameter, we considered the set of values $\mathcal{S} = \{1, 2i\}$ for $i = \{1, ..., 15\}$; i.e., the set containing 1 and all positive even integers from 2 to 30. This grid search is performed iteratively, where each iteration is a combination of 3 steps. In the first step, we fixed α, and γ to be 1 and β is chosen from the set $\mathcal{S}$. Therefore the set of points considered for this step is:

$$(\alpha, \beta, \gamma) = (1, \beta_i, 1), \text{where } \beta_i \in \mathcal{S}. \tag{3.8}$$

For each point in the above set $(1, \beta_i, 1)$, we trained our DFB network and calculated the GAR for the overall system for a security of 104 bits using a 5-fold cross validation. Using this method, we found the best value for hyper-parameter β that gave us the highest GAR with the values of α and γ fixed as 1. This best value of β will be denoted as β^t where the superscript t signifies the iteration number.

In the second step, we repeated the same process with α and β fixed at 1 and choosing γ from the set $\mathcal{S}$:

$$(\alpha, \beta, \gamma) = (1, 1, \gamma_i), \text{where } \gamma_i \in \mathcal{S}. \tag{3.9}$$

Again using a 5-fold cross validation, we found the best value for hyper-parameter γ, which is denoted by γ^1, that gave us the highest GAR with the values of α and β fixed as 1. In the third step, the same procedure was performed by keeping β, and γ fixed at 1 and found the best value for hyper-parameter α, which is denoted by α^1, from the set $\mathcal{S}$. These three steps together complete one iteration of the iterative grid search.

In the next iteration, we again performed the above 3 steps but instead of fixing the values of the two parameters to 1, we fixed the value of the two parameters to be the best value found in the previous iteration for those parameters. To explain this, consider the best value of the 3 parameters found in the first iteration, denoted by α^1,β^1,γ^1. In the first step of the second iteration, we fixed α, and γ to be α^1 and γ^1 respectively and chose β from the set $\mathcal{S}$. Therefore the set of points considered for this step is:

$$(\alpha, \beta, \gamma) = (\alpha^1, \beta_i, \gamma^1), \text{where } \beta_i \in \mathcal{S}. \tag{3.10}$$

Again, using a 5-fold cross validation, we found the best value for hyper-parameter β with the other parameters set to α^1 and γ^1. This best value of β will be denoted as β^2 since this is the second iteration. Similarly, we performed the second and third steps of the second iteration to find the γ^2 and α^2 respectively.

We continued performing these iterations until the parameters converged, which implies that the best value of each parameter did not change from one iteration to the other; i.e., $\alpha^t = \alpha^{t-1}, \beta^t = \beta^{t-1}, \gamma^t = \gamma^{t-1}$.

Using the above procedure for hyperparameter tuning, we have found the values of α^t, β^t, and γ^t to be 8, 2, 2 for FCA and 6, 4, 2 for BLA respectively. The importance of each term will be further discussed in the ablation study in Section 3.5.4.

3.3.2 Network parameters for the Face-CNN

The network used for the Face-CNN is the VGG-19 with an added fully connected layer *fc3* (shown in Fig. 3.2). The Face-CNN is fine-tuned end-to-end with the CASIA-Webface [71], which contains 494,414 facial images corresponding to 10,575 subjects. After fine-tuning with CASIA-Webface, the Face-CNN is next fine-tuned with the 2013 session of the WVU-Multimodal face 2012-21013 dataset [90]. The WVU-Multimodal face dataset for the year 2012 and 2013 together contain a total of 119,700 facial images corresponding to 2263 subjects with 294 common subjects. All the raw facial images are first aligned in 2-D and reduced to a fixed size of 224×224 before passing through the network [72]. The only other pre-processing is subtracting the mean RGB value, computed on the training set, from each pixel. The training is carried out by optimizing the multinomial logistic regression objective using mini-batch gradient descent with momentum. The batch size was set to 40, and the momentum to 0.9. The training was regularized by weight decay (the L2 penalty multiplier set to 0.0005) and dropout regularization for the first three fully-connected layers (dropout ratio set to 0.5). We used batch normalization for fast convergence. The learning rate was initially set to 0.1, and then decreased to 90% of its value every 10 epochs. The number of nodes in the last fully connected layer *fc3* before the softmax layer is 1024 for the FCA and 64 for the BLA. This implies that the feature vector that is extracted from the Face-CNN and fused with the feature vector from Iris-CNN has 1024 dimensions for the FCA and 64 for the BLA.

3.3.3 Network parameters for the Iris-CNN

The network used for the Iris-CNN is the VGG-19 with an added fully connected layer *fc3*. First, the Iris-CNN has been fine-tuned end-to-end using the combination of CASIA-Iris-Thousand[1] and ND-Iris-0405 [91] with about 84,000 iris images corresponding to 1355 subjects. Next, the Iris-CNN is fine-tuned using the 2013 session of the WVU-Multimodal iris 2012-2013 dataset[2]. The WVU-Multimodal iris dataset for the year 2012 and 2013 together contain a total of 257,800 iris images corresponding to 2263 subjects with 294 common subjects. All the raw iris images are segmented and normalized to a fixed size of 64×512 using Osiris (Open Source for IRIS) which is an open source iris recognition system developed in the framework of the BioSecure project [92]. There is no other pre-processing for the iris images. The other hyper-parameters are consistent with the fine-tuning of the Face-CNN. As with the Face-CNN, the iris network has an output of 1024 for FCA and 64 for BLA.

3.3.4 Network parameters for the Joint Representation Layer

The details of the network parameters for the two JRL architectures are discussed in this subsection:

Fully Concatenated Architecture

In the FCA, the 1024-dimensional outputs of the Face-CNN and Iris-CNN are concatenated vertically to give a 2048-dimensional vector. The concatenated feature vector is then passed through a fully connected layer which reduces the feature dimensionality from 2048 to 1024 and also fuses the iris and face features. The hashing layer is also a fully connected layer that outputs a 1024-dimensional vector and includes a tanh activation function.

For the training of the DFB model, we have used a two-step training procedure. First, only the JRL was trained for 65 epochs on a batch size of 32. The learning rate was initially set to 0.1, and then decreased to 90% of its value every 20 epochs. The other hyperparameters

[1] http://biometrics.idealtest.org/

[2] http://biic.wvu.edu

are consistent with the fine-tuning of the Face-CNN. After training of the JRL, the entire DFB model was fine-tuned end-to-end for 25 epochs on a batch size of 32. The learning rate was initialized to 0.07 which is the final learning rate in the training process of the joint fully connected layer in the first step. The learning rate was decreased to 90% of its value every 5 epochs. For this two-step training process, we have used the 2013 session of the overlap subjects in the 2012 and 2013 sessions from the WVU-Multimodal dataset. This common subset consists of 294 subjects with a total of 18700 face and 18700 iris images with the same number of face and iris images per subject.

Bilinear architecture

For the BLA, we do not add *fc3* (i.e., the additional fully connected layer) to either the Face-CNN or the Iris-CNN. In addition, the number of nodes in the first and second fully connected layers $fc1$ and $fc2$ are reduced to 512 and 64, respectively. This means that the output feature vector of the face and iris networks have 64 dimensions rather than the 1024 dimensions of the FCA. The 64-dimensional outputs of the Face-CNN and Iris-CNN are combined in the bilinear (outer product) layer using the matrix outer product as explained in Sec. 3.2.2. The bilinear layer produces an output of dimension $64 \times 64 = 4096$ fusing the iris and face features. The bilinear feature vector is then passed through a fully connected layer, which reduces the feature dimension from 4096 to 1024 followed by a hashing layer which produces a binary output of 1024 dimensions.

In the first step of the two-step training process, only the JRL was trained for 80 epochs on a batch size of 32. The momentum was set to 0.9. The learning rate was initially set to 0.1, and then decreased by a factor of 0.1 every two epochs. The other hyperparameters and the input image sizes are consistent with the training process used in FCA. After training of the JRL, the entire DFB model was fine-tuned for 30 epochs on a batch size of 32. The learning rate was initialized to 0.0015 which is the final learning rate in the training process of the JRL in the first step. The learning rate was decreased by a factor of 0.1 every five epochs. The other hyper-parameters are consistent with the training of the JRL in FCA.

3.3.5 Parameters for the FEC Decoding

The cancelable template generated from the CTM is considered to be the noisy codeword of some error correcting code that we can select. Due to its maximum distance seperable (MDS) property, we have selected Reed-Solomon (RS) codes and used RS decoder for FEC decoding in SSTM. The G-dimensional cancelable template is passed through a RS decoder to identify the closest codeword, which is the multimodal secure sketch.

RS codes use symbols of length m bits. The input to the RS decoder is of length $N' = 2^{m-1}$ in symbols, which means the number of bits per input codeword to the decoder is $n' = mN'$. For example, if the symbol size $m = 6$ then $N' = 63$ is the codeword length in symbols and $n' = 378$ is the codeword length in bits. Let's assume the size of the cancelable template is $G = 378$ bits, which is the number of bits at the input to the RS decoder. This 378-dimensional vector is decoded to generate a secure sketch whose length is K' symbols or, equivalently, $k' = mK'$ bits. K' can be varied depending on the error correcting capability required for the code and k' also signifies the security of the system in bits [49].

We have used *shortened* RS codes. A shortened RS code is one in which the codeword length is less than 2^{m-1} symbols. In standard error control coding, the shortening of the RS code is achieved by setting a number of data symbols to zero at the encoder, not transmitting them, and then re-inserting them at the decoder. A shortened $[N, K]$ RS code essentially uses an $[N', K']$ encoder, where $N' = 2^m - 1$, where m is the number of bits per symbol (symbol size) and $K' = K + (N' - N)$. In our experiments we have used $m = 8$ and $N' = 255$. In the case of using shortened RS codes, the size of the cancelable template is considered equal to N symbols rather than N' symbols. For example, the output of the cancelable template block could be 768 bits which equals to $N = 768/8 = 96$ symbols. The security of the secure multimodal template depends on the selected value of K, implying that the security of the system is k bits, where $k = mK$. The output of the decoder is a length-k binary message, which is cryptographically hashed and stored as the secure multimodal template in the database. When a query is presented for authentication, the system approves the authentication only if the cryptographic hashes of the query matches with the specific enrolled identity.

3.4 Experimental Results for the cancelable multimodal template

We have evaluated the matching performance and the security of our proposed secure multibiometric system using the WVU multimodal database [90] containing images for face and iris modalities. Note that all the experiments have been performed with optimized hyper-parameters. We have used $\{\alpha, \beta, \gamma\}$ as $\{8, 2, 2\}$ for FCA and $\{6, 4, 2\}$ for BLA, respectively.

In this section, we analyze the cancelable multimodal template, which is the output of the CTM. Analyzing the output of the CTM helps us to gain insight into the requirements and the strength of the error correcting code to be used in the SSTM. In the next section, we analyze the secure multimodal template, which is the output of the overall secure multimodal system.

3.4.1 Evaluation Protocol

For the cancelable multimodal template, EER has been used as one of the metrics to evaluate the matching performance for various levels of random bit selection (values of G). EER indicates a value that the proportion of false acceptances is equal to the proportion of false rejections. The lower the equal error rate value, the higher the accuracy of the biometric system. We have also used the genuine and impostor distribution curves along with the ROC curves to evaluate the matching performance of the cancelable template.

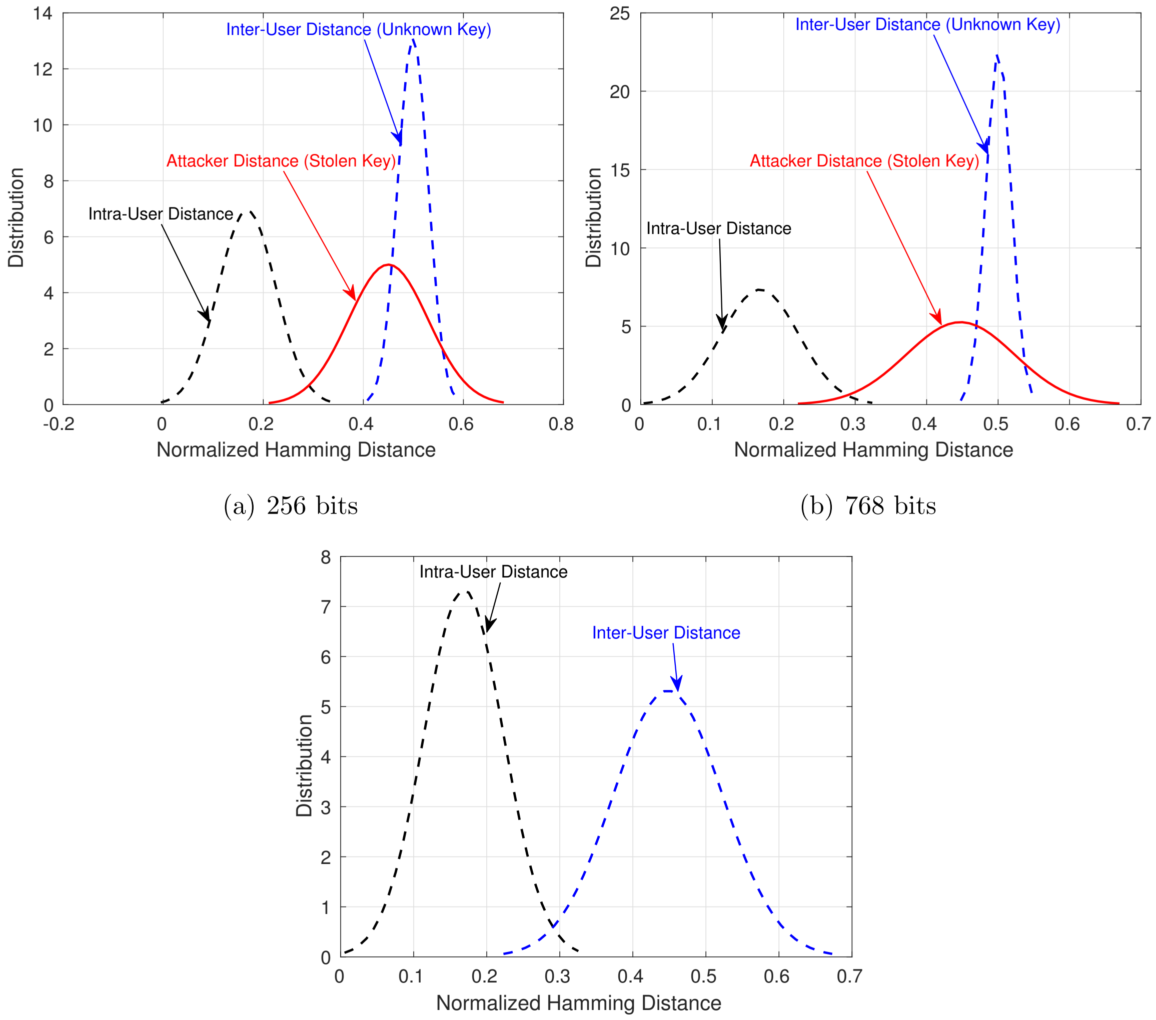
3.4.2 Performance Evaluation

After fine-tuning the entire DFB, we test this network by extracting features using the JRL of the DFB. In both the FCA and BLA architectures, the output is a 1024-dimensional joint binarized feature vector. For testing, we have used 50 subjects from the WVU-Multimodal 2012 dataset. The training and testing set are completely disjoint which means these 50 subjects have never been used in the training set. 20 face and 20 iris images are chosen randomly for each of these 50 subjects. This will give us 20 pairs (face and iris) per subject with no repetitions. These 1,000 pairs (50 × 20) are forward passed through the DFB and 1024-dimensional 1,000 fused feature vectors are extracted. A user-specific random-bit selection is performed using the fused feature vector to generate the cancelable multimodal template. The number of randomly selected bits G that we have used in our experiments is equal to 128, 256, 512, 768 bits out of the 1024 dimensional binary fused vector to generate the cancelable multimodal template.

In this section, we present the results for the statistical analysis of the cancelable multimodal template. We have used two different architectures (FCA and BLA) for fusing the face and iris features. The performance evaluation for each architecture is also discussed here.

Two scenarios have been considered for the evaluation of the secure templates. One is the unknown key scenario. In this scenario, the impostor does not have access to the key of the legitimate user. The impostor tries to break into the system by posing as a genuine user by presenting an artificially synthesized key (which is different from the actual key of the genuine user) and also presenting impostor biometrics. This means that the impostor will try to present random indices for our random-bit selection method in the CTM. These random indices are different from the actual indices that were selected during the enrolment for the legitimate user. The other scenario is the stolen key scenario. In this scenario the impostor has access to the actual key of the genuine user and tries to break the system by presenting actual key but with an impostor biometrics.

The genuine and impostor distributions for the cancelable template for FCA in the unknown key and stolen key scenarios generated by varying the number of random bits selected by the CTM is given in Fig. 3.4. The genuine and impostor distributions shown in Fig. 3.4 have been generated by fitting a normal distribution curve to the histogram. We first observe that there is no overlap between the inter-user (impostor) and intra-user (genuine) distributions. These distributions assume that every user employs his own key. Also plotted is an attacker (stolen key) distribution in which a user (attacker) decides to use the key of another user (victim). In this case, the attacker distribution slightly overlaps with the genuine distribution, but the overlap between the two is still reasonably small. In addition, we also observe that as the number of random bits selected grows from 256 to 768, the overlap be-



(a) 256 bits

(b) 768 bits

(c) 1024 bits

Figure 3.4: Genuine and impostor distribution of cancelable template distances using FCA for varying number of random bits.

tween the genuine and impostor distributions reduces in both the scenarios. However, when all the 1024 bits are used, the overlap again is increased. This clearly shows the trade-off between the security (selection of 'G' random bits) and the matching performance (overlap of the distributions). Notice that there is no "stolen key" curve in Fig. 3.4(c) as all the 1024 bits are used and there is no down-selection of bits, and hence, no key.

The EER plots for FCA and BLA are given in Fig. 3.5 and Fig. 3.6, respectively. EER plot is obtained by calculating the value of EER by varying the length of the cancelable template (number of randomly selected bits). In general, it can be observed from the EER

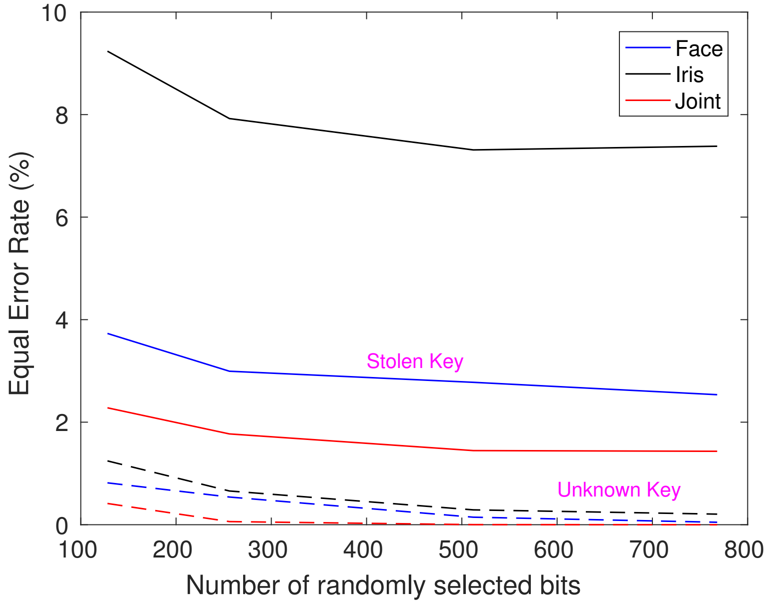


Figure 3.5: EER curves for face, iris, joint-FCA modalities in unknown key (dashed lines) and stolen key (solid lines) scenarios using different sizes of cancelable template.

plots that there is an increase in performance by using additional biometric features and the multimodality (joint) template performs better than the individual modalities (face and iris).

As seen from the curves, the EER for the joint modality is lower than the EER for face or iris. For example, the EER for joint modality using FCA and BLA at 512 bits for stolen key scenario is 1.45% and 1.99%, respectively. Using the same settings, the EER for face and iris is 2.6% and 7.4%, respectively. This clearly shows that there is an improvement by fusing multiple modalities.

The ROC curves for both the architectures have been compared in Fig. 3.7 and 3.8 for unknown and stolen key scenarios, respectively, when the number of randomly selected values (security) is 768 bits. Again, we can clearly observe that the joint modality performs better than the individual modality. For a FAR of 0.5%, the GAR for stolen key scenario using FCA and BLA is 98.25% and 96.33%, respectively. For face and iris, the GAR is 90.8% and 62.5%, respectively at an FAR of 0.5%.

As observed from the plots, the matching performance is not compromised for high security and the multimodality gives us better performance than unimodality.

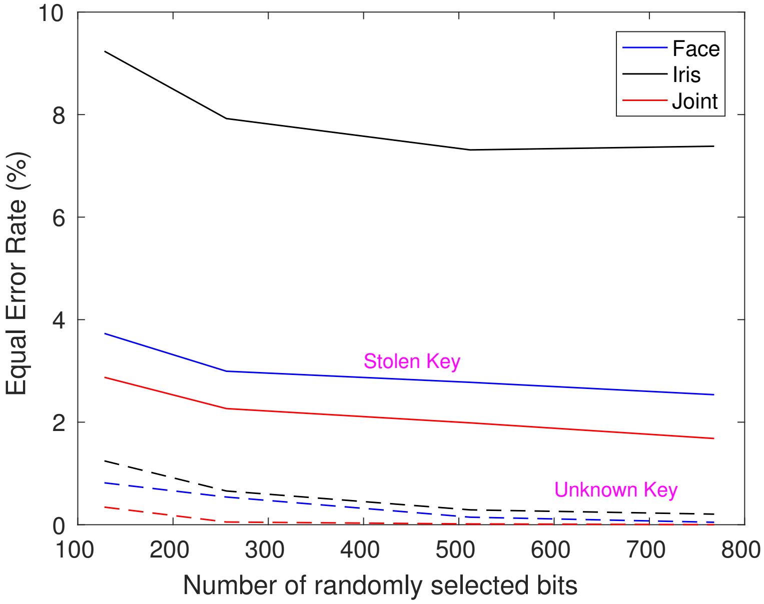


Figure 3.6: EER curves for face, iris, and joint-BLA modalities in unknown key (dashed lines) and stolen key (solid lines) scenarios for different sizes of cancelable template.

3.5 Experimental Results for the Overall System

In this section, we analyze the performance at the output of the overall system, where the output of the overall system is the secure multimodal template that is stored in the database.

3.5.1 Evaluation Protocol

We evaluate the trade-off between the matching performance and the security of the proposed secure multimodal system using the curves that relate the GAR to the security in bits (i.e., the G-S curves). The G-S curve is acquired by varying the error correcting capability of the RS code used for FEC decoding in the SSTM. The error correcting capability of a code signifies the number of bits (or symbols) that a given ECC can correct. The error correcting capability of a RS code is given by $\frac{(N-K)}{2}$ symbols or $\frac{(n-k)}{2}$ bits. We vary the error correcting capability of the code by using different code rates (K/N).

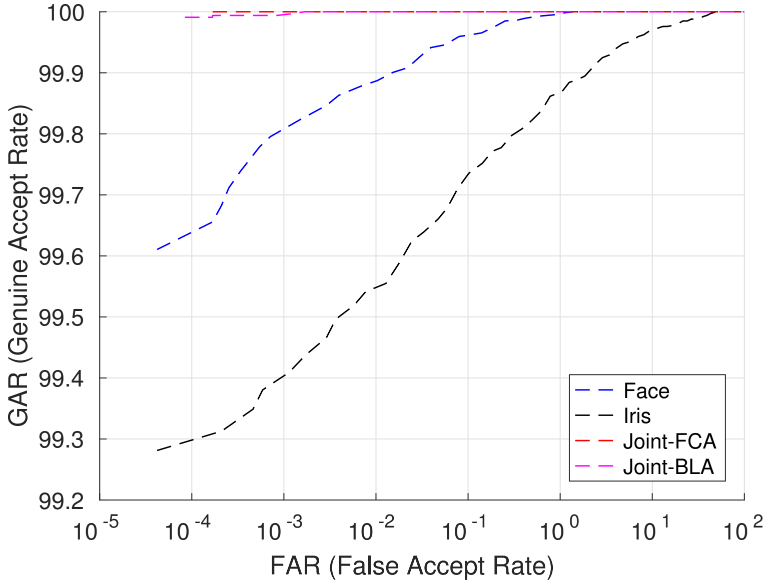


Figure 3.7: ROC curves for face, iris, joint-FCA, and joint-BLA modalities in unknown key scenario for a random selection of 768 bits. The FAR and GAR are in %

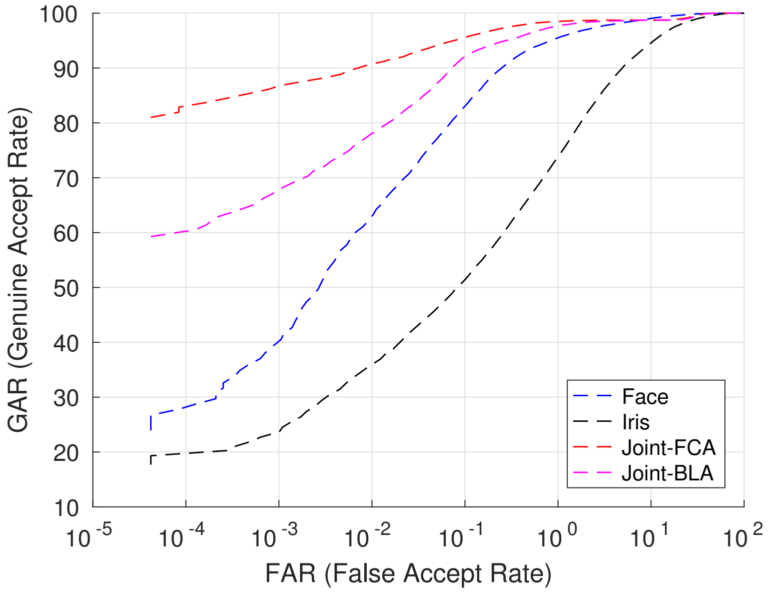


Figure 3.8: ROC curves for face, iris, joint-FCA, and joint-BLA modalities in stolen key scenario for a random selection of 768 bits. The FAR and GAR are in %

3.5.2 Performance Evaluation

As explained in Sec. 3.3.5, the output of the cancelable template block (n bits) is decoded in order to generate a multimodal secure sketch of length k bits, where k also represents the security of the proposed secure multibiometric system. This multimodal sketch is cryptographically hashed and stored as the secure multimodal template in the database. When a query is presented for authentication, the system authenticates the user only if the cryptographic hash of the query matches that of the specific enrolled identity.

We have experimented with different values of N symbols with $m = 8$ and $N' = 255$ symbols using shortened RS codes. The G-S curves for different values of n bits (equivalent to N symbols) for unknown and stolen key scenarios using FCA and BLA are given in Fig. 3.9 and Fig. 3.10, respectively. We can observe from the curves that as the size of the cancelable template in bits (n) increases, the GAR for a given level of security in bits (k) also increases.

Table 3.1: GARs of FCA and BLA in unknown and stolen key scenarios at a security level of 56, 80 and 104 bits using different cancelable template size (N).

N (symbols)	n (bits)	Security (K) (symbols)	Security (k) (bits)	$\frac{(n-k)}{2}$	FCA-GAR		BLA-GAR	
					Unknown	Stolen	Unknown	Stolen
32	256	7	56	100	82.30%	82.15%	82.25%	80.66%
		10	80	88	31.32%	32.68%	36.67%	35.92%
		13	104	76	4.3%	4.33%	6.77%	6.07%
64	512	7	56	228	99.65%	99.68%	98.95%	99.77%
		10	80	216	97.85%	94.95%	94.63%	94%
		13	104	204	84.63%	82.05%	84.41%	85.15%
96	768	7	56	356	99.93%	99.99%	99.55%	99.22%
		10	80	344	99.37%	99.44%	99.04%	99.04%
		13	104	332	98.95%	99.16%	96.51%	96.75%

For example at a security (k) of 104 bits (equivalent to $K = 13$ symbols) using FCA with the stolen key scenario, the GAR for n=128, 256, 512, and 768 bits is equal to $0.62\%, 4.33\%, 82.05\%,$ and 99.16%, respectively. Similarly for the unknown key scenario and FCA, the GAR for n=128, 256, 512, and 768 bits is equal to $0.74\%, 4.3\%, 84.63\%,$ and

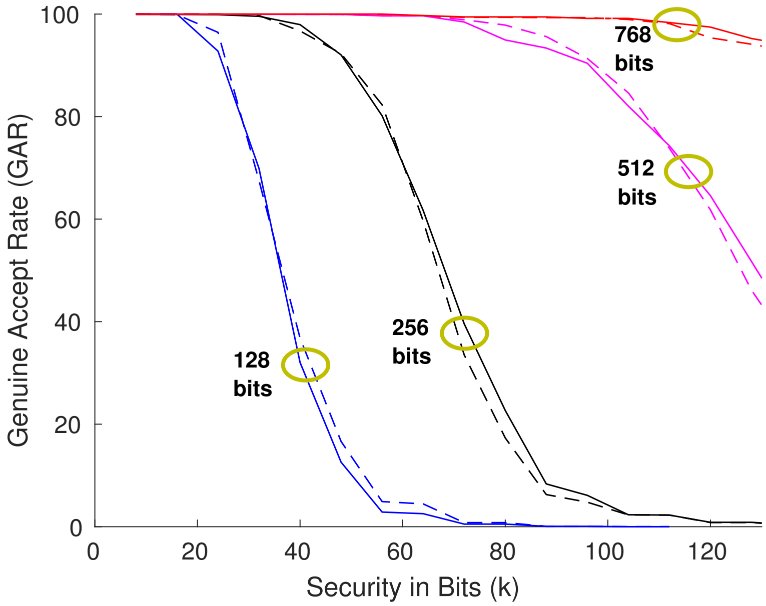


Figure 3.9: G-S curves using FCA in unknown key (dashed) and stolen key (solid) scenarios for different values of n bits.

98.95%, respectively. It can be observed that the use of a larger cancelable template results in better performance. This performance improvement can be attributed to the fact that an increase in n at a fixed value of k (security) improves the error correcting capability of the RS codes which is given by $\frac{(n-k)}{2}$ and hence a better matching performance.

Table 3.1 summarizes the GAR for different values of n at security levels of 56, 80, and 104 bits using both FCA and BLA. The error correcting capabilities in bits $\left(\frac{(n-k)}{2}\right)$ for the RS codes at different security levels are also given in the table. From the Table 3.1, it can be observed that for a given size of the cancelable template in bits (n), the error correcting capability decreases with an increase in the required security levels in bits (k) of the system, which results in a decrease in GAR. This implies that the code cannot correct the intra-class variations at high code rates (k/n) (higher value of k), which results in a reduced GAR. This is the trade-off between the matching performance (GAR) and the security (k) of the system. We have chosen a minimum security level of 56 bits for comaprison in Table 3.1 which is higher when compared to those reported in the literature [2].

The plot in Fig. 3.11 gives a comparison of G-S curves for face, iris, joint-FCA, and joint-BLA modalities using $m = 8$, $N' = 255$ and $n = 768$ bits (equivalent to $N = 96$ symbols) for unknown and stolen key scenario, respectively. The security for the iris modality in stolen

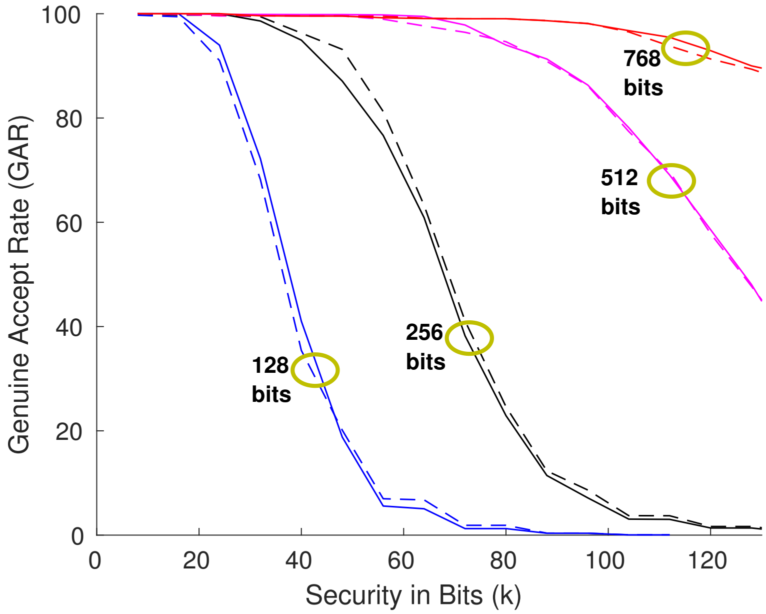


Figure 3.10: G-S curves using BLA in unknown key (dashed) and stolen key (solid) scenarios for different values of n bits.

key scenario at a GAR of 95% is 20 bits. However, by incorporating additional biometric features (face), the security of the multibiometric system using FCA increases to 128 bits at the same GAR.

3.5.3 Comparison with State-of-the-Art Hashing Techniques

As a further experiment, we compare the proposed hashing technique with other hashing techniques. This is done by replacing our hashing method with two other hashing methods [93] and [66], and then training and testing the multimodal authentication system using the same WVU multimodal dataset. The rest of the system is kept the same for comparison purposes. We have compared our hashing technique with supervised semantics-preserving deep hashing (SSDH) [93], and HashNet [66] and evaluated the overall system to produce G-S curves. We have used the FCA system for comparison. We denote the system with our proposed hashing technique as “FCA”, use “FCA+SSDH” to denote the FCA architecture with our hashing function replaced by the SSDH hashing, and use “FCA+HashNet” to denote our FCA architecture with the HashNet hashing function. Fig. 3.12 shows G-S curves for stolen key and unknown key scenarios. It can clearly be seen that our proposed

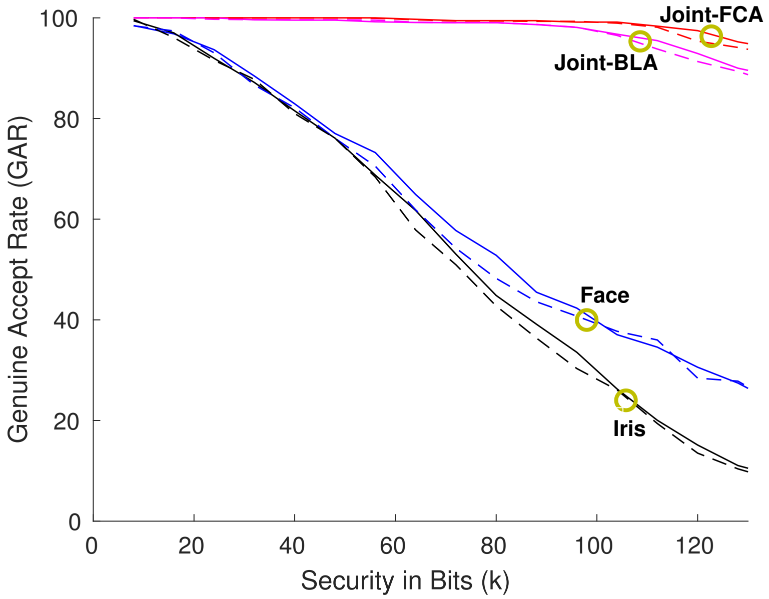


Figure 3.11: G-S curves for face, iris, joint-FCA, and joint-BLA modalities in unknown key (dashed lines) and stolen key (solid lines) scenarios for $n = 768$ bits.

hashing method performs better than the other two deep hashing techniques for the given multimodal biometric security application. Compared to the other two hashing techniques, our proposed method improves the GAR by at least 1.15% at a high security of 104 bits.

3.5.4 Ablation Study

The objective function defined in (3.6) contains 3 constraints, one for the semantics-preserving binary codes (i.e., for classification) and two constraints for efficient binary codes (i.e., for binarization and entropy maximization). In this section, we study the relative importance of each of these terms.

First, we measure the influence of the classification term E_1 by setting $\alpha = 1$, $\beta = 0$, and $\gamma = 0$. Using this setting, we train our DFB model and evaluate the overall system by calculating the GAR for a security of $k = 56$, 80, and 104 bits for $n = 768$ bits (similar to Table 3.1) on the test data for the WVU-Multimodal 2012 dataset. We also study the effect of the binarization constraint along with classification term by setting $\alpha = 1$, $\beta = 1$, and $\gamma = 0$, train our DFB model and again evaluate the overall system by calculating the GARs. Finally, we set $\alpha = 1$, $\beta = 1$, and $\gamma = 1$, and train the DFB model and evaluate the overall system. We performed this experiment for both FCA and BLA architectures only for stolen

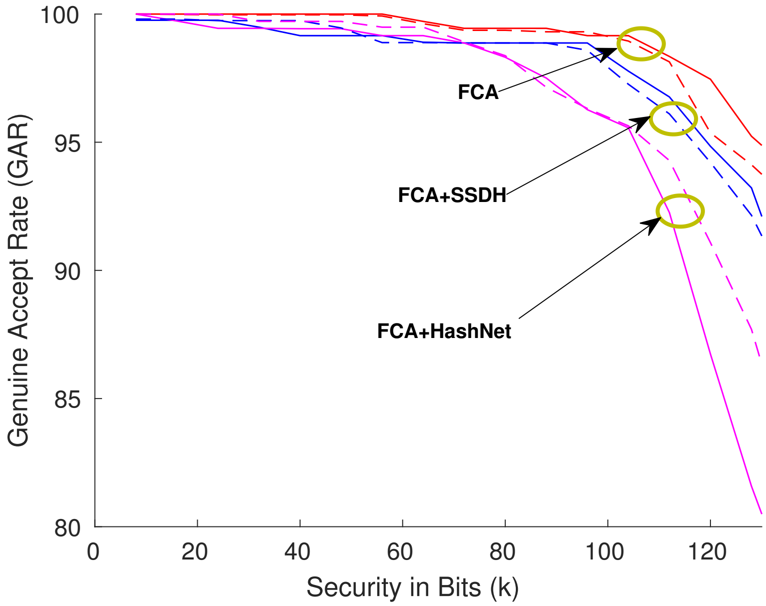


Figure 3.12: G-S curves that compare the performance of the proposed hashing technique with two other hashing techniques for FCA in unknown key (dashed lines) and stolen key (solid lines) scenarios.

key scenario because we can see from Table 3.1 that unknown key and stolen key scenarios give very similar results. The GAR results for this experiment are shown in Table 3.2.

It can be observed from Table 3.2 that the classification term E_1 is the most important term. However, adding the binarization and the entropy constraints E_2 and E_3 (i.e., $\alpha = 1, \beta = 1, \gamma = 1$) definitely help to improve the matching performance (i.e., GAR) by at least 1.25% at a high security of 104 bits in our proposed system. We also note that this performance improvement is evident for both FCA and BLA architectures. Therefore, using all the terms proves beneficial to improve the matching performance evident at higher level of security for both FCA and BLA architectures.

3.5.5 Privacy Analysis

The objective of our work is to design a multimodal authentication system that maximizes the matching performance while keeping the biometric data secure. However, the problem is complicated by the possibility that the adversary may gain access to the enrollment key $\mathbf{k_e}$, the multimodal secure sketch $\mathbf{s_e}$, the enrollment feature vector $\mathbf{e}$, or any combination

Table 3.2: GARs of FCA and BLA in stolen key scenario showing the influence of each term in the objective function.

Hyper-parameters	n (bits)	Security (k) (bits)	FCA-GAR	BLA-GAR
$\alpha = 1, \beta = 0, \gamma = 0$	768	56	99.16%	98.71%
		80	98.32%	96.87%
		104	95.26%	93.29%
$\alpha = 1, \beta = 1, \gamma = 0$	768	56	99.73%	98.76%
		80	98.8%	97.14%
		104	95.72%	94.72%
$\alpha = 1, \beta = 0, \gamma = 1$	768	56	99.52%	98.70%
		80	98.41%	97.02%
		104	95.43%	93.98%
$\alpha = 1, \beta = 1, \gamma = 1$	768	56	99.9%	99%
		80	99.8%	97.6%
		104	96.5%	95.6%

thereof. Using this information, the adversary could not only compromise the authentication integrity of the system, but may also extract information about the biometric data. The system should be robust in these scenarios and the system design should minimize the privacy leakage, which is the leakage of the user biometric information from the compromised data, and preserve authentication integrity of the system.

The G-S curves which have been discussed in Sec. 3.5.2 quantify the security of the system. In this subsection, we will quantify the privacy leakage of the user's biometric information for our proposed system. The privacy of the user is compromised if the adversary gains access to the enrollment feature vector $\mathbf{e}$ as we assume that the enrollment feature vector can be de-convolved to recover the biometric data of the user. The information leaked about the user's enrollment feature vector $\mathbf{e}$ can be quantified in the form of mutual information:

$$I(\mathbf{e}; \mathbf{V}) = H(\mathbf{e}) - H(\mathbf{e}|\mathbf{V}), \tag{3.11}$$

where $\mathbf{e}$ represents the enrollment feature vector, and $\mathbf{V}$ represents the information that adversary has access to. $\mathbf{V}$ could be the enrollment key $\mathbf{k_e}$ and/or the multimodal secure sketch $\mathbf{s_e}$. $H(\mathbf{e})$ represents entropy of $\mathbf{e}$ and quantifies the number of bits required to specify $\mathbf{e}$. In particular, $H(\mathbf{e}) = J$ because the optimization described in Sec. 3.3.1 is designed to ensure that the J bits in the encoded template are independent and equally likely to be 0 or 1. $H(\mathbf{e}|\mathbf{V})$ is the entropy of $\mathbf{e}$ given $\mathbf{V}$ and quantifies the remaining uncertainty about $\mathbf{e}$ given knowledge of $\mathbf{V}$. The mutual information $I(\mathbf{e};\mathbf{V})$ is the reduction in uncertainty about $\mathbf{e}$ given $\mathbf{V}$ [1].

Let's first assume that the adversary gains access to the enrollment key $\mathbf{k_e}$. In this case $\mathbf{V} = \mathbf{k_e}$ and mutual information is given as:

$$I(\mathbf{e};\mathbf{k_e}) = H(\mathbf{e}) - H(\mathbf{e}|\mathbf{k_e}) = 0, \tag{3.12}$$

because $H(\mathbf{e}|\mathbf{k_e}) = H(\mathbf{e}) = J$ as the key $\mathbf{k_e}$ does not give any information about the enrollment feature vector $\mathbf{e}$. $\mathbf{k_e}$ just gives the indices of the random values selected from $\mathbf{e}$ but does not provide values at those indices.

The information leakage when the multimodal secure sketch $\mathbf{s_e}$ or the pair $(\mathbf{k_e}, \mathbf{s_e})$ is compromised can be quantified using the conditional mutual information because $\mathbf{s_e}$ is dependent on $\mathbf{r_e}$ which is driven by $\mathbf{k_e}$. Hence, the information leakage when the secure sketch is compromised is conditionally dependent on $\mathbf{k_e}$ and is given as:

$$I(\mathbf{e};\mathbf{s_e}|\mathbf{k_e}) = H(\mathbf{e}|\mathbf{k_e}) - H(\mathbf{e}|\mathbf{s_e},\mathbf{k_e}), \tag{3.13}$$

where $H(\mathbf{e}|\mathbf{k_e})$ quantifies the remaining uncertainty about $\mathbf{e}$ given knowledge of $\mathbf{k_e}$ and $H(\mathbf{e}|\mathbf{s_e},\mathbf{k_e})$ quantifies the remaining uncertainty about $\mathbf{e}$ given knowledge of $\mathbf{k_e}$ and $\mathbf{s_e}$. This conditional mutual information is measured under two scenarios discussed below.

Both $\mathbf{s_e}$ and $\mathbf{k_e}$ are compromised: In this scenario the adversary gains access to both $\mathbf{s_e}$ and $\mathbf{k_e}$. As previously discussed, $H(\mathbf{e}|\mathbf{k_e}) = H(\mathbf{e}) = J$ because knowing $\mathbf{k_e}$ does not provide any information about $\mathbf{e}$. If the adversary knows $\mathbf{s_e}$, the information leakage of $\mathbf{r_e}$ due to $\mathbf{s_e}$ is equal to the length of $\mathbf{s_e}$ which is k bits. The adversary can use this information of $\mathbf{r_e}$ with the additional knowledge of the enrollment key $\mathbf{k_e}$ and exactly know the indices and the values for the k bits in the enrollment vector $\mathbf{e}$. However, there is still uncertainity about the remaining $J-k$ bits of the enrollment feature vector $\mathbf{e}$, which implies

$H(\mathbf{e}|\mathbf{s_e}, \mathbf{k_e}) = J - k$. Therefore, the information leakage about enrollment feature vector when both secure sketch and enrollment key are compromised is given as:

$$I(\mathbf{e}; \mathbf{s_e}|\mathbf{k_e}) = H(\mathbf{e}|\mathbf{k_e}) - H(\mathbf{e}|\mathbf{s_e}, \mathbf{k_e}) = J - (J - k) = k. \tag{3.14}$$

Only $\mathbf{s_e}$ is compromised: In this scenario the adversary gains access to only $\mathbf{s_e}$. Even in this case if the adversary knows $\mathbf{s_e}$, the information leakage of $\mathbf{r_e}$ due to $\mathbf{s_e}$ is k bits. However, the adversary does not have any information about the enrollment key $\mathbf{k_e}$ which means that there is added uncertainty in the information about the enrollment feature vector $\mathbf{e}$ as the adversary does not know the exact locations of the k bits given by $\mathbf{s_e}$. This added uncertainity is measured by $H(\mathbf{k_e})$ which is calculated using combinatorics and is given as:

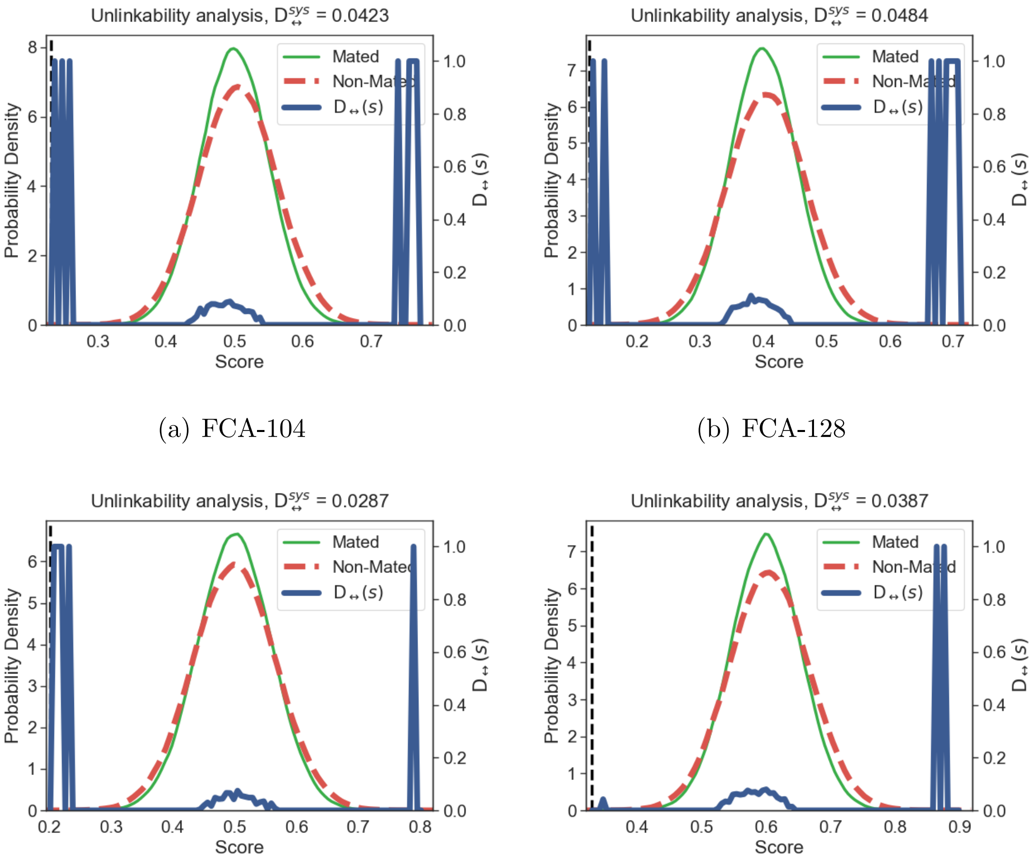
$$H(\mathbf{k_e}) = \log_2 \binom{J}{n}, \tag{3.15}$$

where n is the size of the key and (3.15) provides all the combinations that n bits could be selected from J. Therefore, the conditional mutual information is given as:

$$\begin{aligned} I(\mathbf{e}; \mathbf{s_e}|\mathbf{k_e}) &= H(\mathbf{e}|\mathbf{k_e}) - H(\mathbf{e}|\mathbf{s_e}, \mathbf{k_e}) \\ &= J - \left(J - k + \log_2 \binom{J}{n}\right) \\ &= k - \log_2 \binom{J}{n} \\ &= \max\left(0, k - \log_2 \binom{J}{n}\right), \end{aligned} \tag{3.16}$$

where the max function is applied in the last equation as information leakage cannot be negative. We have evaluated (3.16) using different values of n and k for $J = 1024$ bits. We know that n ranges from 1 to J depending on the number of random bits selected from the enrollment feature vector $\mathbf{e}$ and k ranges from 1 to n depending on the rate of the error correcting code. We found that information leakage is zero for all the values of k for n ranging from 1 to 792 bits. However, if n is above 792, there is a positive information leakage when value of k is taken above 780.

From (3.14) and (3.16), we can conclude that for $J = 1024$, the ideal value of n should be less than 792 and ideal value of k should be small. This would make the information leakage to be zero or small in case if $\mathbf{s_e}$ or the pair $(\mathbf{s_e}, \mathbf{k_e})$ gets compromised. These values of n and k would also keep the matching performance high as shown in Fig. 3.11.



(a) FCA-104 (b) FCA-128

(c) BLA-104 (d) BLA-128

Figure 3.13: Unlinkability analysis of the proposed system for FCA and BLA for different quantities of security bits (104, 128).

3.5.6 Unlinkability Analysis

According to ISO/IEC International Standard 24745, transformed templates generated from the same biometric references should not be linkable across applications or databases. By using the protocol defined in [94], we have evaluated the unlinkability of the proposed system. The protocol in [94] is based on mated (H_m) and non-mated (H_{nm}) samples distributions. Mated samples correspond to the templates extracted from the samples of the same subject using different user-specific keys. Non-mated samples correspond to the templates extracted from the samples of different subjects using different keys. For an unlinkable system, there must exist a significant overlap between mated and non-mated score distributions [94].

Using these distributions, two measures of unlinkability are specified: i) Local measure $D_{\leftrightarrow}(s)$ evaluates the linkability of the system for each specific linkage score s and is dependent upon the likelihood ratio between score distributions. $D_{\leftrightarrow}(s) \in [0, 1]$ and is defined over the entire score domain. $D_{\leftrightarrow}(s) = 0$ denotes full unlinkability, while $D_{\leftrightarrow}(s) = 1$ denotes full linkability of two transformed templates at score s. All values of $D_{\leftrightarrow}(s)$ between 0 and 1 indicate an increasing degree of linkability. ii) Global measure $D_{\leftrightarrow}^{sys}$ provides an overall measure of the linkability of the system independent of the score domain and is a fairer benchmark for unlinkability comparison of two or more systems. $D_{\leftrightarrow}^{sys} \in [0, 1]$, where $D_{\leftrightarrow}^{sys} = 1$ indicates full linkability for all the scores of the mated samples distribution and $D_{\leftrightarrow}^{sys} = 1$ indicates full unlinkability for the whole score domain. All values of $D_{\leftrightarrow}^{sys}$ between 0 and 1 indicates an increasing degree of linkability.

According to the benchmark protocol defined in [94], six transformed databases were generated from WVU Multimodal face and iris test dataset by using different set of random bits (enrollment key) in the CTM for each template of a subject. The linkage score we have used is the Hamming distance between the $\mathbf{s_e}$ and $\mathbf{s_p}$. The mated samples distribution and the non-mated samples distribution were computed across these six databases. These score distributions are used to calculate local measure $D_{\leftrightarrow}(s)$, which is further used to compute the global measure $D_{\leftrightarrow}^{sys}$ (overall linkability of the system). Fig. 3.13 shows unlinkability curves when transformed templates are generated for joint-FCA, and joint-BLA modalities using $m = 8, N' = 255$, and $n = 768$. We have tested with two quantities of security bits $k = 104$ and $k = 128$ bits. With significant overlap, the overall linkability of the system is close to zero for both joint-FCA ($D_{\leftrightarrow}^{sys} = 0.048$) and joint-BLA ($D_{\leftrightarrow}^{sys} = 0.038$). Based on this discussion, the proposed system can be considered to be unlinkable.

3.6 Summary

We have presented a feature-level fusion and binarization framework using deep hashing to design a multimodal template protection scheme that generates a single secure template from each user's multiple biometrics. We have employed a hybrid secure architecture combining the secure primitives of *cancelable biometrics* and *secure-sketch* and integrated it with a deep hashing framework, which makes it computationally prohibitive to forge a combination of multiple biometrics that passes the authentication. We have also proposed two deep learning based fusion architectures, *fully connected architecture* and *bilinear architecture* that could be used to combine more than two modalities. Moreover, we have analyzed the matching performance and the security of the proposed secure multibiometric system. Experiments using the WVU multimodal dataset,, which contain face and iris modalities, demonstrate that the matching performance does not deteriorate with the proposed protection scheme. In fact, both the matching performance and the template security are improved when using the proposed secure multimodal system.

Chapter 4

Learning to Authenticate with Deep Multibiometric Hashing and Neural Network Decoding

In this chapter, we propose a novel MDHND architecture, which integrates a deep hashing framework with a NND to create an effective multibiometric authentication system. The MDHND consists of two separate modules: a MDH module, which is used for feature-level fusion and binarization of multiple biometrics, and a NND module, which is used to refine the intermediate binary codes generated by the MDH and compensate for the difference between enrollment and probe biometrics (variations in pose, illumination, etc.)

4.1 Introduction

Multimodal biometric authentication systems use a combination of different biometric traits such as face and iris, or face and fingerprint to authenticate the user. One of the major challenges of multimodal systems is the selection of the fusion algorithm required to combine the multiple modalities. The fusion of the modalities can be performed at sensor, feature, score, or decision levels [3, 16, 49, 65, 95]. However, fusion-level authentication systems have better performance because they leverage the richer information available about the biometric data in the feature-set [96]. Feature-level fusion involves combining of feature vectors that are obtained from multiple feature sources. Multibiometric systems can combine biometrics in many ways, including: (i) feature vectors obtained from different sensors for the same biometric; (ii) feature vectors obtained from different entities for the same biometric, such as combination of feature vectors obtained from left and right eyes; or (iii) feature vectors obtained from multiple biometric traits, such as face and iris. Fusion at the feature level is relatively difficult to achieve in practice due to incompatibility of feature sets extracted from multiple modalities and because the relationship between different feature spaces may be unknown [97]. To overcome this issue, we propose a feature-level fusion of multiple biometrics using CNNs.

In this chapter, we have combined multiple modalities at feature-level using two different CNN architectures. Similar work has been done in [98], however, the CNN features are high-dimensional and real-valued, and usually require high computational cost [99]. In order to reduce the computational complexity, in this chapter, we have also integrated a novel deep hashing algorithm into a feature-level fusion CNN architecture to design a MDH network for combining multiple modalities. Deep hashing is the application of deep learning to generate compact binary vectors from raw image data [52, 53]. In addition to using deep hashing for feature-level fusion, we have also used ECC as an additional component to compensate for the difference in enrollment and probe biometrics (arising from variation in pose, illumination, noise in biometric capture). Due to this difference, enrollment and probe biometrics lead to different hash code at the output of the MDH network and therefore a failure to authenticate. ECC can compensate for this difference by forcing the enrollment and probe biometric to decode to the same message, then using that message for authentication, making the system more robust to distortion in the biometric measurements, which helps in improving the authentication performance.

Recent work has shown that the same kinds of neural network architectures used for classification can also be used to decode ECC [100]. In this work, we integrate a NND [100] into our deep hashing architecture as an ECC component to improve the authentication performance. The NND is a formulation of the BP algorithm as a neural network. The input to the NND can be considered to be a corrupted codeword of an ECC and this corrupted codeword is within a certain distance of a correct codeword of an ECC. The corrupted codeword can be decoded using the NND to generate the correct codeword. It can be argued that a conventional ECC decoder can be used instead of using a NND, however, implementing the decoder as a neural network has the benefit of providing the same architecture as the classifier (MDH network) that extracts the hashing code, and hence, it can be more efficiently jointly optimized and implemented within a common framework. Another advantage with NND is that it provides an opportunity to jointly learn and optimize with respect to biometric datasets, which are not necessarily characterized simply by Gaussian noise as is assumed by a conventional decoder. Motivated by this, in this chapter, we have integrated our MDH network with NND and a joint optimization process to formulate our novel MDHND framework for an end-to-end multimodal biometric authentication system.

To summarize the main contributions of this chapter include:

1. Conversion of the involved biometric traits into a common feature space by using modality-specific CNNs.

2. Fusion and binarization of the individual modality features to generate a robust binary latent shared representation.

3. Inclusion and optimization of the neural network based decoder to compensate for the distortion in biometric measurements.

4. End-to-end joint optimization of the overall system.

4.2 Multimodal Deep Hashing Neural Decoder

In this section, we present a system overview of the MDHND shown in Fig. 4.1. MDHND consists of two important modules: MDH module and NND module. We have considered face and iris as the two modalities for this authentication system. However this system could be used with other modalities and can be extended to more than two modalities.

4.2.1 Multimodal Deep Hashing Module

The MDH module, which is shown in Fig. 4.2, consists of a DSL containing face and iris CNNs and the JRL. The primary functions of the MDH module are the non-linear feature-level fusion and binarization of the fused features.

The DSL of the MDH module consists of a CNN for encoding the face ("Face-CNN") and another CNN for encoding the iris ("Iris-CNN"). The output feature vectors of the face and iris CNNs are fused and binarized in the JRL, which is split into two sub-layers: a fusion layer and a hashing layer. The main function of the fusion layer is to fuse the individual face and iris representations from domain-specific layers into a non-linear multimodal feature embedding. The hashing layer binarizes the shared multimodal feature representation that is generated by the fusion layer.

Fusion layer: We have implemented two different architectures for the fusion layer: FCA and BLA. In the FCA, the outputs of the Face-CNN and Iris-CNN are concatenated vertically in the concatenation layer and passed through a fully connected layer to fuse the iris and face features. In FCA, the concatenation layer and the fully connected layer together constitute the fusion layer. In the BLA (Fig. 4.2), the outputs of the Face-CNN and Iris-CNN are combined using the matrix outer product of the face and iris feature vectors to create bilinear feature vector. Similar to FCA, the bilinear feature vector is also passed through a fully connected layer. In BLA, the outer product layer and the fully connected layer together constitute the fusion layer.

Hashing layer: The output of the fusion layer is a shared multimodal feature vector of unquantized values. The output of the fusion layer can be directly binarized by thresholding at any numerical value or thresholding at the population mean. However, this kind of thresholding leads to a quantization loss, which results in sub-optimal binary codes. To account for this quantization loss, we have included another latent layer after the fusion layer, which is known as the hashing layer (shown in orange in Fig. 4.2). The main function of the hashing layer is to capture the quantization loss incurred while converting the shared multimodal representation (output of fusion layer) into binary codes.

To generate the binary hash codes, we can directly use the sign activation function

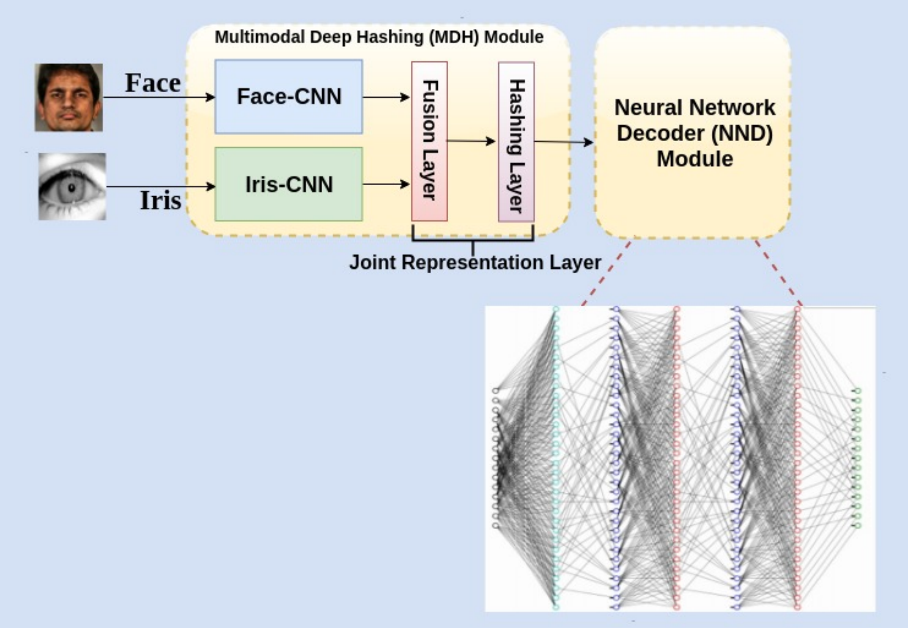

Multimodal Deep Hashing (MDH) Module
Face
Iris
Face-CNN
Iris-CNN
Fusion Layer
Hashing Layer
Joint Representation Layer
Neural Network Decoder (NND) Module

Figure 4.1: Block diagram of the proposed system. The NND is shown in the figure. The MDH module is shown in Fig. 4.2

$h = \text{sgn}(z)$ for the hashing layer. However, the use of the non-smooth sign-activation function makes standard back-propagation impracticable as the gradient of the sign function is zero for all non-zero inputs. We have used a continuation method to overcome this zero-gradient problem by starting with a smooth activation function $y = \tanh(\beta x)$ and making it gradually sharper by increasing the bandwidth β as the training proceeds. This continuation utilizes the relationship between the sign activation function and the scaled tanh function:

$$\lim_{\beta \to \infty} \tanh(\beta x) = \text{sgn}(x), \tag{4.1}$$

where $\beta > 0$ is a scaling parameter. For training the network, we start with a $\tanh(\beta x)$ activation for the hashing layer with $\beta = 1$ and continue training until the network converges to zero loss. We then increase the value of β while holding other training parameters equal to the previously converged network parameters, and start retraining the network. This process is repeated several times by increasing the bandwidth of the tanh activation allowing $\beta \to \infty$ until the hashing layer can generate an output very close to binary values. In addition to using this continuation method, the overall objection function used for training the deep hashing network is discussed in Sec. 4.3.1

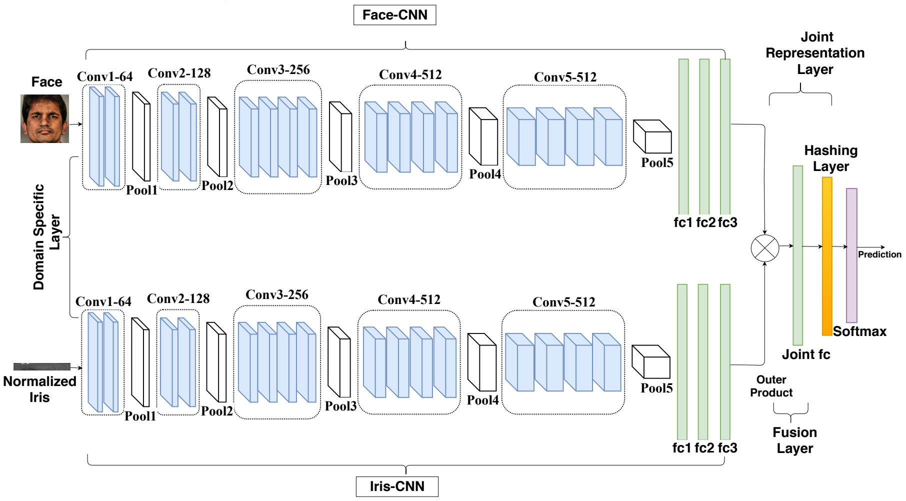


Figure 4.2: Proposed multimodal deep hashing (MDH) framework for the bilinear architecture (BLA).

4.2.2 Neural Network Decoder Module

The output of the MDH module after training can be binarized by directly using a sign function. Henceforth, we will refer to the output of the MDH network as *intermediate binary code*. Even though we still use a threshold of 0.5, the quantization loss is much lower and authentication performance is improved when compared to values with no hashing layer. However, there is still room to make the system more robust and improve the performance. This is achieved by using ECC. There could be distortion in biometric measurements such as variations in pose, illumination, or noise due to the biometric capturing device, which leads to difference in enrollment and probe biometrics. Due to these differences, enrollment and probe biometrics may lead to different hash codes at the output of the MDH network and therefore a failure to authenticate. ECC can compensate for this difference by forcing the enrollment and probe biometric to decode to the same message, then using that message for authentication, making the system more robust to noise in the biometric measurements, which helps in improving the authentication performance.

Recent research in the field of ECC has focused on designing a neural network architecture as an ECC decoder [45,100]. We can adapt such a neural network based ECC decoder, train it, and use it as an ancillary component to refine the intermediate binary codes generated by MDH. The advantage of using NND instead of a conventional decoder is that it allows for a common architectural framework to be used for both the classifier (i.e., the MDH) and the decoder and it provides an opportunity to jointly learn and optimize with respect to biometric datasets, which are not necessarily characterized simply by Gaussian noise as is assumed by conventional decoders. For our application, we have chosen the NND described in [100] as the basis for our neural network ECC decoder to be integrated with the MDH network. Due to space limitations, we do not provide full details on the operation of NND, details can be found in the original paper. Rather, we focus the discussion on the changes in how the decoder is trained relative to [100].

4.3 Training Steps for Multimodal Deep Hashing Neural Decoder

The MDHND framework described in Sec. 4.2 is trained in 3 steps. In Step 1, we use a novel loss function to train and learn the MDH parameters to generate binary shared multimodal latent code; in Step 2, the latent binary codes from Step 1 are passed through an ECC decoder to generate the ground truth, which will be used to fine-tune a NND; in Step 3, the NND decoder is trained using the ground truth from Step 2 and this NND is then integrated with the MDH network followed by a joint optimization of the overall system MDHND. In this section, we discuss the 3 steps used during training of the proposed MDHND system.

4.3.1 Step 1: Training of the Multimodal Deep Hashing Module

In this step, the MDH network is trained for feature-level fusion and binarization of the multiple biometrics. For modality-specific CNNs, VGG-19 pre-trained on ImageNet dataset is used as a starting point followed by fine-tuning the entire VGG-19 with additional fully connected layer $fc3$. The Face-CNN is fine-tuned end-to-end with the CASIA-Webface [71], which contains 494,414 facial images corresponding to 10,575 subjects. The Iris-CNN is fine-tuned end-to-end using the combination of the CASIA-Iris-Thousand[1] and ND-Iris-0405 [91]

[1] http://biometrics.idealtest.org/

datasets with about 84,000 iris images corresponding to 1355 subjects.The Face-CNN and Iris-CNN are also fine tuned on the 2013 face and iris subsets of the WVU-Multimodal 2012-2013 datasets[2] respectively. The WVU-Multimodal dataset for the year 2012 and 2013 together contain a total of 119,700 facial images and 257,800 iris images corresponding to 2263 subjects with 294 common subjects. For fine-tuning the Face-CNN and Iris-CNN, we have used 58,200 facial images and 121,200 iris images, respectively, corresponding to 1,060 subjects from the WVU-Multimodal 2012-2013 dataset. For fine-tuning the CNNs, we have not used any of the common subjects from 2012-2013 WVU dataset, so that these common subjects can be used for fine-tuning the NND and also for testing the overall system.

For the Face-CNN, all the raw facial images are first aligned in 2-D and cropped to a size of 224 × 224 before passing through the network [72]. The only other pre-processing is subtracting the mean RGB value, computed on the training set, from each pixel. The training is carried out by optimizing the multinomial logistic regression objective using mini-batch gradient descent with momentum. The number of nodes in the last fully connected layer *fc3* before the softmax layer is 1,024 for the FCA and 64 for the BLA. This implies that the feature vector that is extracted from the Face-CNN and fused with the feature vector from Iris-CNN has 1,024 dimensions for the FCA and 64 for the BLA. For the iris-CNN, all the raw iris images are segmented and normalized to a fixed size of 64 × 512 using Osiris (Open Source for IRIS) which is an open source iris recognition system developed in the framework of the BioSecure project [92]. As with the Face-CNN, the iris network has an output of 1,024 for FCA and 64 for BLA.

Next, we train the JRL, which is a combination of the fusion layer and the hashing layer. For training the JRL we have used a novel objective function, which helps in reducing the quantization loss and also maximize the entropy to generate optimal and discriminative binary shared multimodal representation. For training the JRL, we adopt a two-step training procedure where we first train only the JRL greedily by freezing the face and iris CNNs followed by fine tuning the entire MDH module end-to-end using back-propagation at a relatively small learning rate.

The objective function used for training the JRL is a combination of classification loss,

[2] http://biic.wvu.edu

quantization loss and entropy maximization loss. The classification loss has been added into the MDH network by using the *softmax* layer as shown in Fig. 4.2. Let $E_1(\mathbf{w})$ denote the objective function required to fulfill the classification task:

$$E_1(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} L_n(f(x_n, \mathbf{w}), y_n) + \lambda||\mathbf{w}||^2, \tag{4.2}$$

where the first term $L_n(.)$ is the classification loss for a training instance n and is described below, N is the number of training images in a mini-batch. $f(x_n, \mathbf{w})$ is the predicted softmax output of the network and is a function of the input training image x_n and the weights of the network $\mathbf{w}$. The second term is the regularization function where λ governs the relative importance of the regularization. Let the predicted softmax output $f(x_n, \mathbf{w})$ be denoted by $\hat{y}_n$. The classification loss for the n^{th} training instance is given as:

$$L_n(\hat{y}_n, y_n) = -\sum_{m=1}^{M} y_{n,m} \ln \hat{y}_{n,m}, \tag{4.3}$$

where $y_{n,m}$ and $\hat{y}_{n,m}$ is the ground truth and the prediction result for the m^{th} unit of the n^{th} training instance, respectively and M is the number of output units. In addition to using classification loss and the continuation method described in Sec. 4.2.1, we add constraint of quantization loss and entropy maximization to generate efficient binary codes at the output of the MDH module. The output of the hashing layer is a J-dimensional vector denoted by $\mathbf{o}_n^H$, corresponding to the n-th input image. The i-th element of this vector is denoted by $o_{n,i}^H(i = 1, 2, 3, \cdots, J)$. The value of $o_{n,i}^H$ is in the range of $[-1, 1]$ because it has been activated by the tanh activation. To make the codes closer to either -1 or 1, we add a constraint of quantization loss between the hashing layer activations and 0, which is given by $\sum_{n=1}^{N} ||\mathbf{o}_n^H - \mathbf{0}||^2$, where N is the number of training images in a mini-batch and $\mathbf{0}$ is the J-dimensional vector with all elements equal to 0. Let $E_2(\mathbf{w})$ denote this constraint to boost the activations of the units in the hashing layer to be closer to -1 or 1:

$$E_2(\mathbf{w}) = -\frac{1}{J}\sum_{n=1}^{N} ||\mathbf{o}_n^H - \mathbf{0}||^2 = -\frac{1}{J}\sum_{n=1}^{N} ||\mathbf{o}_n^H||^2. \tag{4.4}$$

In addition to forcing the codes to become binarized, we also require that the binary codes have equal number of -1's and 1's, which maximizes the entropy of the discrete distribution and results in hash codes with better discrimination. Let $E_3(\mathbf{w})$ denote this constraint that forces the output of each node to have a 50% chance of being -1 or 1:

$$E_3(\mathbf{w}) = \sum_{n=1}^{N} (\text{mean}(\mathbf{o}_n^H))^2. \tag{4.5}$$

Therefore, the overall objective function that is required to be minimized to generate discriminative efficient binary codes is given as:

$$\alpha E_1(\mathbf{w}) + \beta E_2(\mathbf{w}) + \gamma E_3(\mathbf{w}), \tag{4.6}$$

where α, β, and γ are the tuning parameters of each term.

4.3.2 Step 2: Generating the Ground Truth for Training the Neural Network Decoder

In [100], the goal is to optimize the NND for a gaussian noise channel and the database used reflects various channel output realizations when the zero codeword has been transmitted. However, for our proposed system, we want to optimize the NND to be used with biometric data, where the channel noise for our database is characterized by the image distortions (e.g., pose variations, illumination variations and noise due to biometric capturing device) in different biometric images of the same subject. We can create the input dataset for training the NND by using the different facial and iris images for the same subject. However, we do not have the labels or the ground truth codewords that these input images need to be mapped to. An external conventional ECC decoder can be used to generate the ground truth codewords for this input dataset.

After training the MDH network in Step 1, facial and iris image pairs for a different set of subjects disjoint from the training dataset are used for generating the ground truth for training the NND. We use the MDH network to extract the joint binary features for this disjoint dataset. These extracted features are used as input to a conventional ECC decoder for soft-decision decoding. Hard-limiting the output of the ECC decoder generates ground truth codewords that are used as labels for optimizing the NND in Step 3.

For this step, we have used the facial and iris image pairs of 294 common subjects from the 2013 year of the WVU multimodal 2012-2013 dataset. This implies that we have extracted the features for all the face and iris image pairs of the 294 subjects using the MDH network and decoded it using soft-decision decoding with an ECC decoder to generate the ground truth to be used in Step 3. While usually all the joint feature vectors of a given subject are mapped by the decoder to the same codeword, it is possible that the vectors could be mapped to different codewords. This is especially the case when there are substantive differences in the latent subspaces of the different face/image pairs for the subject or when the ECC is not sufficiently strong. In the case that the feature vectors get mapped to a plurality of codewords, the most common of these codewords is used as the ground truth for that subject.

4.3.3 Step 3: Joint Optimization of the Multimodal Deep Hashing Neural Decoder

In Step 3 of the training, first the NND is trained using the procedure and database from the original paper [100]. In the next step, the NND is fine-tuned for our multibiometric data. For fine-tuning the NND, we use the same database used for the ECC decoder in Step 2, where the input to NND is given by the feature vectors generated by the MDH network and the labels are provided by the decoded codewords generated by the conventional ECC decoder in Step 2. Using this database, the NND is fine-tuned for our joint biometric data. We have used sigmoid activation for the last layer of NND. The sigmoid is added so that the final network output is in the range $[0, 1]$. This makes it possible to train and fine-tune the NND using cross-entropy loss function:

$$L(o, y) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(o_i) + (1 - y_i)\log(1 - o_i), \tag{4.7}$$

where o_i,y_i are the deep neural network output and the actual ith component of the ground truth codeword (label), respectively.

After fine-tuning the NND, we integrate MDH and NND by discarding the softmax layer in the MDH network and connecting the output of the hashing layer from MDH as input to NND to create and end-to-end MDHND network. This MDHND is optimized end-to-end using the same dataset as used for fine-tuning NND and also the same cross-entropy loss function given in (4.7). For fine-tuning the NND and end-to-end optimization of the MDHND, we use the same 294 subjects from the 2013 year of the WVU multimodal dataset used in Step 2. The total number of facial and iris images corresponding to the 294 subjects is equal to 15,500 and 33, 200, respectively. We use the Adam optimizer [101] with the default hyper-parameter values ($\epsilon = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) to train all the parameters. The batch size in all the experiments is fixed to 32. Our MDHND is implemented in TensorFlow with python API and all the experiments are conducted on two GeForce GTX TITAN X 12GB GPUs.

4.4 Performance Evaluation

4.4.1 Details of the Code and Decoder

The intermediate binary code generated from the MDH module is considered to be the noisy codeword of some error correcting code that we can select and this noisy codeword can be decoded using the NND. We have used BCH code for our experiments. The size of the intermediate binary codes, which is the output of the MDH network depends upon the size of the error correcting code being used for the NND. We have experimented with a few different sizes of the BCH code for NND including BCH(63,45), BCH(127,85), BCH(255,187), and BCH(511,376). We have tried to keep the code rate to be around 0.7. For the external ECC decoder, we have used the BCH decoder from the communication toolbox in MATLAB®.

4.4.2 Evaluation Protocol for Authentication

For evaluation of authentication performance, we use a disjoint dataset of 70 subjects, which have never been seen during training with 20 facial and 20 iris images per subject leading to a total of 1400 face and iris image pairs with no repetitions. For authentication, all the 1400 pairs are forward passed through our proposed MDHND system and genuine and impostor scores are calculated using Hamming distance as our score measure. Based on the number of subjects ($N = 70$) and the number of image pairs ($t = 20$) per subject we obtain $Nt(t-1)/2 = 13,300$ genuine scores and $(N(N-1)t^2)/2 = 966,000$ imposter scores.

We have used the ROC curve and the EER as our performance metric for authentication. The ROC curve is plotted between the GAR and the FAR at all possible values of score thresholds. EER indicates a value that the proportion of false acceptances is equal to the proportion of false rejections.

4.4.3 Evaluation Protocol for Identification

We have also evaluated our system for identification application. The proposed system can also be used for identification as long as the database of individuals is not too large because if we make the ECC too strong, it may lead to ambiguity, where multiple users are resolved to the same codeword and the system may not be too discriminative and may not provide good performance. For identification performance, MDHND is trained on 294 mutual subjects from year 2013, and is tested on the same subjects from year 2012 of the WVU multimodal dataset. The total number of image pairs in the test set for identification is equal to 2940 corresponding to 294 subjects. For identification, the score can be calculated against each of the 294 stored codes and the class yielding the best score can be identified as the subject class. We have used the classification accuracy as our performance metric for identification. The classification accuracy is defined as the ratio of the correct predictions made to the the total number of predictions for a given testing set.

4.4.4 Baselines

We have compared our algorithms with some of the state-of-the-art score, decision, and fusion-level algorithms. CNN-Sum is a score-level fusion algorithms, which use the probability outputs for the test sample of each modality, added together to give the final score vector. CNN-Major is a decision-level fusion algorithm, which chooses the maximum number of modalities taken to be from the correct class. The feature level fusion techniques include serial feature fusion [102], parallel feature fusion [103], CCA-based feature fusion [104], and discriminant correlation analysis (DCA/MDCA) [105] methods. For the serial, parallel and CCA fusion techniques, we have used principal component analysis (PCA) [106] and linear discriminant analysis (LDA) for dimensionality reduction followed by K-nearest neighbor

(KNN) classifier [107]. To compare the results for the proposed system, with the state-of-the-art algorithms, we extract Gabor features in five scales and eight orientations for face and iris modalities. For each face, and iris image 31, 360, and 36, 630 features are extracted respectively. These hand-crafted Gabor features are used only for serial, parallel, CCA and DCA fusion techniques. For all the other algorithms being compared we have used CNN features.

We have also compared our multibiometric system with single modality system denoted as Iris-CNN and Face-CNN. As stated previously, we have implemented two different fusion architectures FCA and BLA in our MDH network. MDH-FCA/BLA denotes a stand-alone MDH network with feature level fusion using FCA/BLA and with no ECC decoder of any kind being used. For such cases in comparison, when no error correcting code is used, the length of the output feature vector is equal to the codeword length n of the BCH(n,k) code. MDH-FCA+Ext.Decoder/MDH-BLA+Ext.Decoder denotes feature level fusion using FCA/BLA in the MDH network integrated with an external conventional decoder. MDH-FCA+NND/MDH-FCA+NND denotes feature level fusion using FCA/BLA in the MDH network integrated with a non optimized NND (NND is just trained as in [100] with AWGN channel but not fine-tuned with biometric data) and also with no joint optimization. MDHND-FCA/MDHND-BLA denotes our proposed overall system with feature level fusion using FCA/BLA in the MDH network integrated and jointly optimized with a NND for biometric data.

4.4.5 Authentication Results

Table 4.1 shows a comparison of authentication performance for our proposed system with other methods in terms of EER. As it can be seen that our proposed MDHND framework outperforms single modality by at least 1.5%. It can be observed that there is an improvement in performance by using additional biometric features and the multimodality system performs better than the unimodal systems. We can also observe that there is an improvement in performance when ECC decoding is used, and that this improvement improves when the NND decoder is used. For example, for BCH(255,187) and the BLA architecture, the EER

Table 4.1: EER for different algorithms using different BCH codes.

Algorithm	BCH(63,45)	BCH(127,85)	BCH(255,187)	BCH(511,376)
CNN-Sum	1.71%	1.64%	1.32%	1.29%
CNN-Major	2.11%	2.19%	1.61%	1.53%
Iris-CNN	2.74%	2.34%	2.10%	2.04%
Face-CNN	1.91%	1.72%	1.63%	1.58%
MDH-FCA	1.84%	1.67%	1.48%	1.43%
MDH-BLA	1.76%	1.66%	1.45%	1.42%
MDH-FCA+Ext.Decoder	1.73%	1.64%	1.41%	1.39%
MDH-BLA+Ext.Decoder	1.62%	1.58%	1.32%	1.28%
MDH-FCA+NND	1.39%	1.34%	1.30%	1.23%
MDH-BLA+NND	1.30%	1.26%	1.24%	1.19%
MDHND-FCA	**1.05%**	**0.99%**	**0.90%**	**0.86%**
MDHND-BLA	**1.01%**	**0.94%**	**0.84%**	**0.79%**

decreases from 1.45% without ECC decoding to 1.32% when a conventional decoder is used. The EER further decreases to 1.24% when an unoptimized NND is used (i.e., trained on AWGN) and to 0.84% when an optimized NND is used (i.e., trained on biometric data) for a net improvement of about 0.5% relative to the conventional decoder.

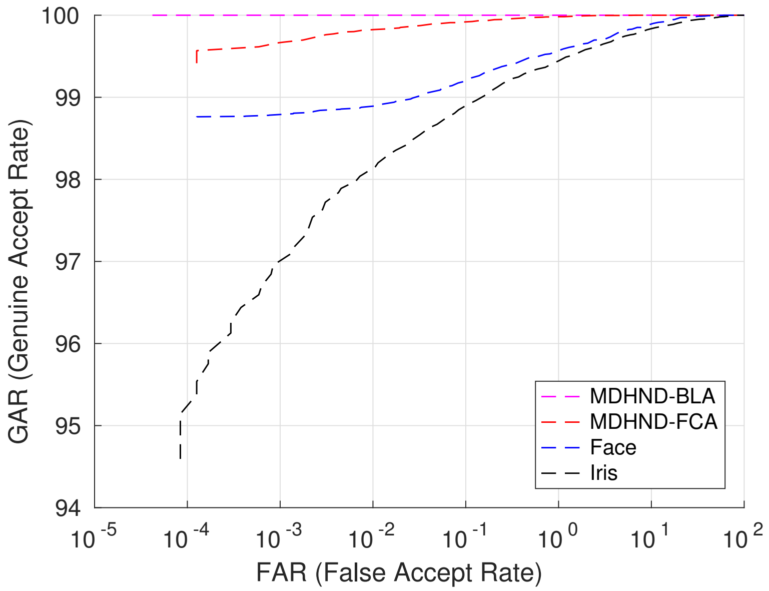


Figure 4.3: ROC curves for face, iris, MDHND-FCA, and MDHND-BLA modalities using BCH(511,376). FAR and GAR are given in %.

Fig. 4.3 shows the comparison of ROC curves for unimodal and a multimodal system. It is evident that the multimodal feature-level fusion using BLA integrated with NND and

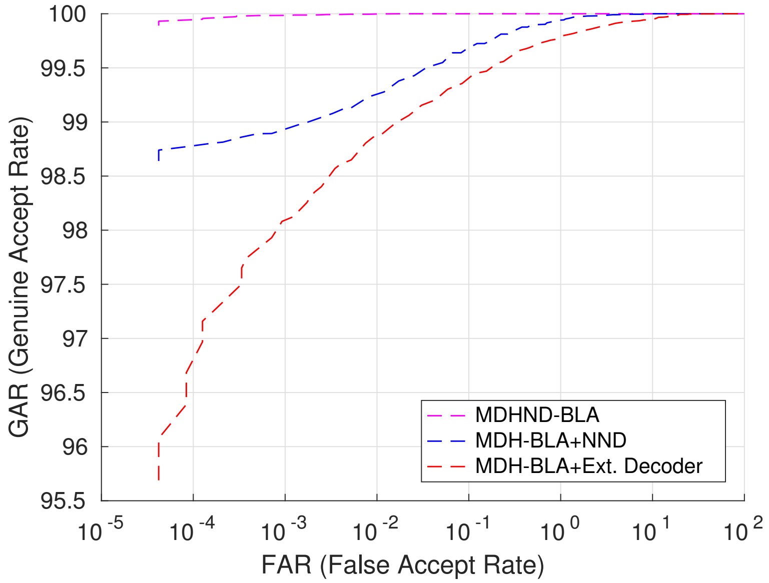


Figure 4.4: ROC curves for face, iris, MDHND-FCA, and MDHND-BLA modalities using BCH(255,187) and BLA architecture. FAR and GAR are given in %.

joint optimization significantly improves unimodal representation accuracy by using a bilinear formulation for exploiting the captured multiplicative interactions of the low-dimensional modality-dedicated feature representations. Fig. 4.4 shows the comparison of ROC curves showing the improvement in performance by using an optimized NND (MDHND-BLA) relative to a non-optimized NND (MDH-BLA+NND) and an external decoder (MDH-BLA+Ext. Decoder). It can clearly be seen that with a FAR of 0.01% we get an improvement in GAR of about 1.3% by using an optimized NND relative to a conventional external decoder.

We have also evaluated the time/delay required to perform a single authentication. We have measured the time for 1000 authentications using our trained MDHND-BLA with the same hardware as described at the end of section 4.3.3 . Based on this evaluation, the total time required for 1000 authentication is equal to 9.625 secs, which equals to an average of 9.625ms per authentication.

4.4.6 Identification Results

Table 4.2 shows a comparison of our proposed systems with different state of the art fusion algorithms for identification task. It can clearly be seen that the performance of our proposed systems (MDHND-FCA and MDHND-BLA shown in the last two rows) in

Table 4.2: Classification accuracy using different BCH codes.

Algorithm	BCH(63,45)	BCH(127,85)	BCH(255,187)	BCH(511,376)
Serial + PCA + KNN	68.92%	70.47%	70.83%	71.12%
Serial + LDA + KNN	76.11%	78.00%	80.15%	80.52%
Parallel + PCA + KNN	71.2%	73.21%	74.1%	74.69%
Parallel + LDA + KNN	77.4%	80.19%	82.11%	82.53%
CCA + PCA + KNN	84.34%	85.12%	87.45%	87.21%
CCA + LDA + KNN	87.43%	88.65%	88.98%	89.12%
DCA/MDCA + KNN	79.08%	81.67%	82.09%	83.02%
CNN-Sum	95.13%	95.21%	95.54%	96.11%
CNN-Major	93.34%	94.16%	95.23%	95.71%
Iris-CNN	92.03%	92.39%	93.22%	94.91%
Face-CNN	94.19%	94.97%	95.06%	95.65%
MDH-FCA	94.76%	95.45%	95.82%	96.01%
MDH-BLA	94.88%	95.26%	95.98%	96.36%
MDH-FCA+Ext.Decoder	95.16%	95.88%	96.11%	96.23%
MDH-BLA+Ext.Decoder	95.22%	95.34%	96.28%	96.72%
MDH-FCA+NND	96.32%	97.22%	97.28%	97.83%
MDH-BLA+NND	96.98%	97.11%	97.29%	98.12%
MDHND-FCA	**98.02%**	**98.13%**	**99.10%**	**99.11%**
MDHND-BLA	**98.16%**	**98.94%**	**99.13%**	**99.23%**

terms of classification accuracy is significantly better than previous fusion approaches. It can also be observed that using optimized NND (MDHND-BLA) definitely helps to improve the identification performance and we get an improvement of about 3% when compared to a conventional external ECC decoder (MDH-BLA+Ext.Decoder) or an improvement of about 3.5% with no decoder (MDH-BLA) being used at all.

4.5 Summary

We have presented a feature-level fusion and binarization framework using deep hashing, and integrated a neural network decoder into the framework. In this framework, we leveraged a neural network based decoder to refine the codes generated by the deep hashing network to improve the authentication performance. We have also implemented multiple architectures for combining the biometrics at feature-level. The experimental results show that the bilinear architecture is better than linear concatenation of features. Additionally, the experimental results show that the optimized neural network decoder decreases the EER of the multimodal biometric system by 0.7%, relative to not using any decoder at all. Also, optimized neural network decoder significantly improves the authentication performance (GAR) of the multimodal biometric system by about 1.3% at an FAR of 0.01%, when compared to using an external conventional ECC decoder. The current work deals with fusion of two modalities and we plan to extend our model and use more than two modalities. Also we intend to use more recent codes such as Turbo and LDPC codes as our error-correcting code.

Chapter 5

Using Deep Cross Modal Hashing and Error Correcting Codes for Improving the Efficiency of Attribute Guided Facial Image Retrieval

In this chapter, we propose a novel cross-modal hashing architecture — DNDCMH, which uses a binary vector specifying the presence of certain facial attributes as an input query to retrieve relevant face images from the database. The DNDCMH network consists of two separate components: an ADCMH module, which uses a margin (m)-based loss function to efficiently learn compact binary codes to preserve similarity between modalities (i.e., facial attribute modality and image modality) in the Hamming space, and a NECD, which is an error correcting decoder implemented with a neural network. The goal of integrating the NECD network with the ADCMH network is to error correct the hash codes generated by ADCMH to improve the retrieval efficiency.

5.1 Introduction

Due to rapid development of internet and social media over the last decade, there has been tremendous volume of multimedia data, which is generated from different heterogeneous sources and includes modalities like images, videos, and texts. Thus, approximate nearest neighbor (ANN) search has attracted a lot of attention from machine learning and computer vision research community to guarantee the retrieval quality and computing efficiency for content based image retrieval (CBIR) in large-scale multimedia datasets. As a fast and an advantageous solution, hashing has been employed in ANN search for CBIR due to its fast query speed and low storage cost [19, 20, 108–118]. The goal of hashing is to map high-dimensional visual data points into compact binary codes in the Hamming space, where the semantic similarity in the original space is approximately preserved in the Hamming space. The key principle in hashing functions is to maintain the semantic similarity by mapping images of similar content to similar binary codes.

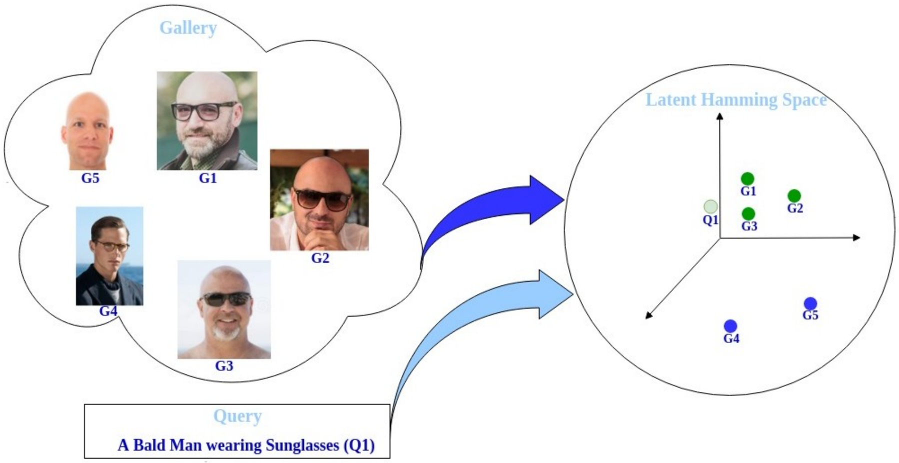


Figure 5.1: Cross modal hashing for facial image retrieval: a bald man wearing sun glass.

Additionally, corresponding data samples from heterogeneous modalities may establish semantic correlations, which leads to CMH. CMH returns relevant information of one modality in response to a query of another modality (e.g., retrieval of texts/images by using a query image/text), where similar hash codes in a shared latent Hamming space are generated for each individual modality. In this chapter, we utilize the cross-modal hashing framework for facial retrieval biometrics application in which the image are retrieved based solely on semantic attributes. For example, in this model, a user can give a query such as “A bald man wearing sunglasses” to retrieve relevant face images from a large-scale gallery. The idea of cross-modal hashing for image retrieval applications is shown in Fig. 5.1. We can note that the relevant points in the gallery G1, G2 and G3 are closer to the query Q1 in the latent hamming space than the points G4 and G5.

There has been a surge in the development of CMH techniques used for ANN search for retrieval on multi-modal datasets. However, capturing the semantic correlation between the heterogeneous data from divergent modalities [119], and bridging the semantic gap between low-level features and high-level semantics for an effective CMH is a challenge. Deep learning techniques for CMH (or "deep cross-modal hashing (DCMH)") [37,38,120] integrate feature learning and hash coding into an end-to-end trainable framework. DCMH frameworks minimize the quantization error of hashing from continuous representation to binary codes and prove the benefit of jointly learning the semantic similarity preserving features.

The main goal in DCMH is to learn a set of hash codes such that the content similarities between different modalities is preserved in the Hamming space. As such, during the learning, a likelihood function [38, 121, 122] or margin-based loss function such as the triplet loss function [123,124] needs to be incorporated into the DCMH framework to improve retrieval performance. In triplet based DCMH [123], the inter-modal triplet embedding loss is applied to model the heterogeneous correlation across different modalities, the intra-modal triplet loss encodes the discriminative power of the hash codes, and the regularization loss forces the *adjacency consistency* to ensure that the hash codes can retain the original cross-modal similarities in the Hamming space. However, in margin-based loss functions, some of the instances of different modalities of the same subject may not be close enough in the Hamming space to guarantee all the correct retrievals. Therefore, it is important to bring the different modalities of the same subject closer to each other in the Hamming space to improve the retrieval efficiency.

In this work, we identified that in addition to the regular DCMH techniques [38,41,42], which exploit entropy maximization and quantization losses in the objective function of the DCMH; an ECC decoder can also be used as an additional component to compensate for the heterogeneity gap and reduce the distance between the different modalities of the same subject in the Hamming space in order to improve the cross-modal retrieval efficiency. We presume that the hash code generated by DCMH is a binary vector that is within a certain distance from a codeword of an ECC. The hash code generated by DCMH can be passed through an ECC decoder, then the closest codeword to this hash code is found. which can be used as a final hash code for the retrieval process. In this process, the attribute hash code and image hash code of the same subject are forced to map to the same codeword, thereby reducing the distance of the corresponding hash codes. This brings more relevant facial images from the gallery closer to the attribute query, which leads to an improved retrieval performance.

Recent work has shown that the same kinds of neural network architectures used for classification can also be used to decode ECC [43–45]. Motivated by this, we have used a NECD [43] as an ECC decoder to improve the cross-modal retrieval efficiency. The NECD is a non-fully connected neural network architecture based on the BP algorithm, which is a notable decoding technique applied in the field of error correcting codes. We have integrated our "ADCMH" with the NECD to formulate our novel DNDCMH framework for cross-modal retrieval (face image retrieval based on semantic attributes), which performs better than other state-of-the-art deep cross-modal hashing methods for facial image retrieval in response to an attribute query.

Specifically, the DNDCMH contains a custom-designed ADCMH network integrated with the NECD. The goal of ADCMH network is to learn pairwise optimized intermediate hash codes for both modalities, while the goal of NECD is to refine the intermediate hash codes generated by ADCMH to improve the cross-modal retrieval efficiency. The entire DNDCMH network is trained end to end by implementing an alternative minimization algorithm in two stages. Stage 1 is split into parts 1(a) and 1(b) such that in Stage 1(a) of this algorithm, we employ a novel cross-modal loss function which uses margin-based distance logistic loss (DLL) to learn the ADCMH parameters. Stage 1(a) of the algorithm generates intermediate cross-modal hash codes. In Stage 1(b), we train a NECD network by relating the error correcting capability e of the ECC used to create the NECD network with the margin m, which is employed in the distance logistic loss for training the ADCMH parameters in Stage 1 (a). In Stage 2 of the alternative minimization algorithm, we pass the intermediate hash codes generated by the ADCMH network (i.e., from Stage 1(a)) through the trained NECD network ((i.e., from Stage 1(b)) to find the closest correct codeword to the intermediate hash codes. The cross-entropy loss between the correct codeword and the intermediate codes is then back-propagated only to the ADCMH network to update its parameters. It should be noted that during the testing, we use only the ADCMH component of the DNDCMH for image retrieval and we do not use the NECD component.

To summarize, the main contributions of this chapter include:

1: Attribute guided deep cross-modal hashing (ADCMH): We utilize deep cross-modal hashing based on margin-based distance logistic loss for face image retrieval in response to a facial attribute query.

2: Integrating a NECD: We exploit the error correcting capability of the ECC and relate it to the margin of distance logistic loss to integrate the NECD network into the ADCMH network to learn error-corrected hash codes using an alternative minimization optimization algorithm.

3: Scalable cross-modal hash: The proposed DNDCMH architecture performs facial image retrieval using point-wise data without requiring pairs or triplets of training inputs, which makes DNDCMH scalable to large scale datasets.

5.2 Proposed Method

In this section, we first formulate the problem; then, we provide the details of our proposed method including the cross-modal hashing framework, training of our proposed scheme, and how it can be leveraged for out-of-sample data.

5.2.1 Problem Definition

We assume that there are two modalities for each sample, i.e., facial attribute and image. Define $\mathbf{X} = \{\mathbf{x}_{\mathrm{i}}\}_{i=1}^{n}$ to represent the image modality, in which $\mathbf{x}_{\mathrm{i}}$ is the raw pixels of image i in a training set of size n. In addition, we use $\mathbf{Y} = \{\mathbf{y}_{\mathrm{i}}\}_{i=1}^{n}$ to represent the attribute modality, in which $\mathbf{y}_{\mathrm{i}}$ is the annotated facial attributes vector related to image $\mathbf{x}_i$. $\mathbf{S}$ is a cross-modal similarity matrix in which $S_{ij} = 1$ if image $\mathbf{x}_{\mathrm{i}}$ contains a facial attribute $\mathrm{y_j}$, and $S_{ij} = 0$ otherwise.

Based on the given training information (i.e., $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{S}$), the goal of our proposed method is to learn modality-specific hash functions: $h^{(x)}(\mathbf{x}) \in \{-1, +1\}^c$ for image modality, and $h^{(y)}(\mathbf{y}) \in \{-1, +1\}^c$ for attribute modality to map the image $\mathbf{x}$ and attribute feature

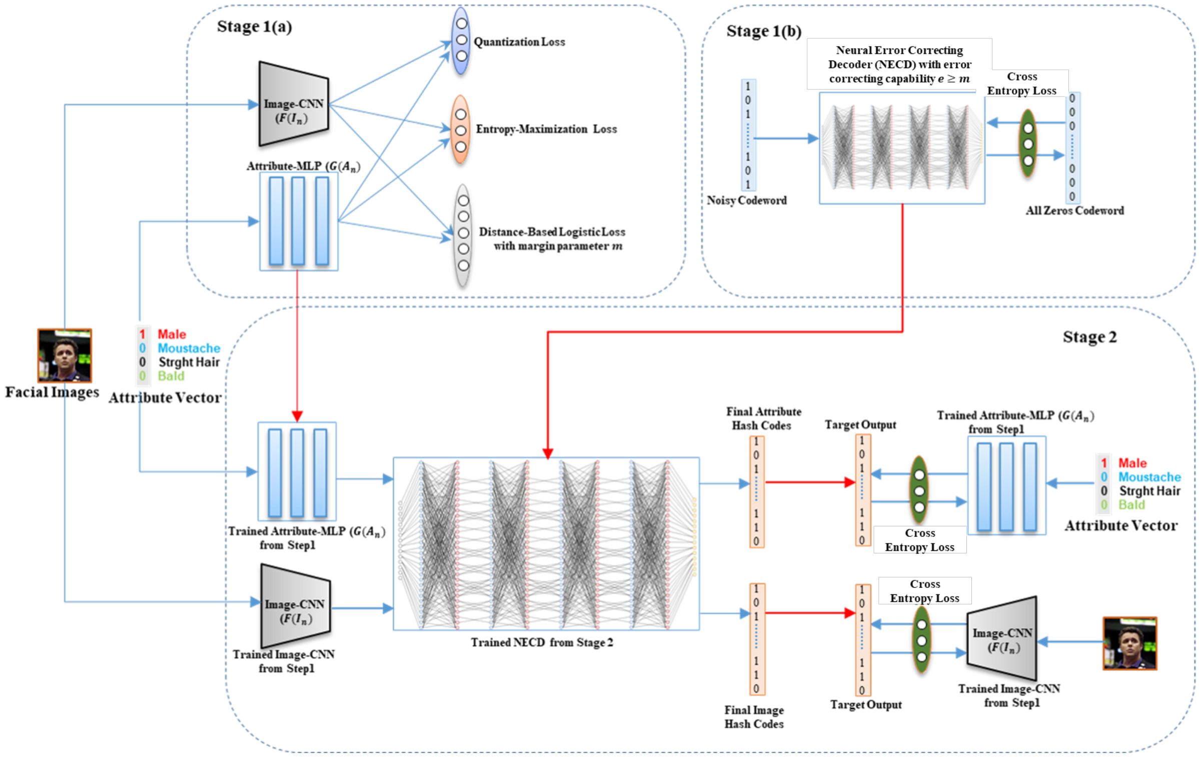


Figure 5.2: Schematic illustration of our DNDCMH.

vector $\mathbf{y}$ into a compact c-bit hash code. Hash codes need to be learned such that the cross-modal similarity in $\mathbf{S}$ is preserved in the Hamming space. Specifically, if $S_{ij} = 1$, the Hamming distance between the binary codes $\mathbf{c}_i^{(\mathbf{x}_i)} = h^{(x)}(\mathbf{x}_\mathrm{i})$ and $\mathbf{c}_j^{(\mathbf{y}_j)} = h^{(y)}(\mathbf{y}_\mathrm{j})$ should be small and if $S_{ij} = 0$, the corresponding Hamming distance should be large.

5.2.2 Deep Neural Decoder Cross-Modal Hashing

The schematic of our proposed DNDCMH is shown in Fig. 5.2. The DNDCMH architecture consists of two important components: 1) A "attribute-based deep cross-modal hashing" (ADCMH) architecture, which contains an Image-CNN, and an Attribute-MLP. 2) A Neural error-correcting decoder (NECD). As shown in Fig. 5.2, the entire DNDCMH network is trained end to end by implementing an alternative minimization algorithm in two stages:

Stage 1 is split into parts 1(a) and 1(b) such that in Stage 1(a) of this algorithm, we employ a novel cross-modal loss function which uses margin-based distance logistic loss (DLL) to learn the ADCMH parameters. Stage 1(a) of the algorithm generates intermediate cross-modal hash codes. In Stage 1(b), we train a NECD by relating the error correcting capability e of the ECC used to create the NECD with the margin m used in the distance logistic loss for training the ADCMH parameters in Stage 1(a). Specifically, in Stage 1(b), in order to force the attribute and image intermediate hash codes of same subject to be pushed closer and decoded to the same codeword, we chose the ECC used to create the NECD in such a way that the error correcting capability e is greather than or equal to m, where m is the margin parameter used in the distance logistic loss for training the ADCMH parameters in Stage 1(a).

In Stage 2 of the alternative minimization algorithm, we pass the intermediate hash codes (real-valued vector) generated by the ADCMH network (i.e., from Stage 1(a)) through the trained NECD network ((i.e., from Stage 1(b)) to find the closest correct codeword (binary) to the intermediate hash codes. The cross-entropy loss between the correct codeword and the intermediate codes is back-propagated only to the ADCMH network to update its parameters. Stage 1(a) and Stage 2 are carried out iteratively until there is not a significant improvement in retrieval efficiency on the training.

Stage 1(a): Learning Intermediate Hash Codes

Training of ADCMH network to learn the intermediate hash codes: In step 1, we train the ADCMH network, which is a coupled deep neural network (DNN), to learn the intermediate hash codes. The ADCMH has three main objectives: 1) to learn a coupled DNN using distance-based logistic loss to preserve the cross-modal similarity between different modalities; 2) to secure a high retrieval efficiency, for each modality, minimize the quantization error due to the hashing of real-valued continuous output activations of the network to binary codes; 3) to maximize the entropy corresponding to each bit to obtain the maximum information provided by the hash codes.

We design the objective function for ADCMH to generate efficient hash codes. Our objective function for ADCMH comprises of three parts: (1) margin-based distance logistic loss; (2) quantization loss; and (3) entropy maximization loss. The ADCMH is composed of two networks: An image convolutional neural network (Image-CNN), which is used to extract features for image modality and an attribute multi-layer perceptron (Attribute-MLP), which is used to extract features for facial attribute modality. A tanh activation is used for the last layer of both the networks so that the network outputs are in the range of [-1,1]. Let $p(\mathbf{w}_x, \mathbf{x}_{\mathrm{i}}) \in \mathbb{R}^d$ denote the learned CNN features for sample $\mathbf{x}_{\mathrm{i}}$ corresponding to the image modality, and $q(\mathbf{w}_y, \mathbf{y}_{\mathrm{j}})$ denote the learned MLP features for sample $\mathbf{y}_{\mathrm{j}}$ corresponding to the attribute modality. $\mathbf{w}_x$ and $\mathbf{w}_y$ are the CNN network weights and the MLP network weights, respectively. The total objective function for ADCMH is defined as follows:

$$\begin{aligned}\mathcal{J} = &\sum_{i=1}^{n}\sum_{j=1}^{n} \ell_c(p(\mathbf{P}_{*i}, \mathbf{Q}_{*j}), S_{ij}) \\ &- \frac{\theta}{c}(\sum_{i=1}^{n} ||\mathbf{P}_{*i}||^2 + \sum_{j=1}^{n} ||\mathbf{Q}_{*j}||^2) \\ &+ \lambda(\sum_{e=1}^{c} ||(\mathbf{P}^{\mathsf{T}})_{*e}||^2 + \sum_{f=1}^{c} ||(\mathbf{Q}^{\mathsf{T}})_{*f}||^2),\end{aligned} \tag{5.1}$$

where $\mathbf{P} \in \mathbb{R}^{c \times n}$ represents the image feature matrix constructed by placing CNN features of training samples column-wise and $\mathbf{P}_{*i} = p(\mathbf{w}_x, \mathbf{x}_{\mathrm{i}})$ is the CNN feature corresponding to sample $\mathbf{x}_{\mathrm{i}}$. $(\mathbf{P}^{\mathsf{T}})_{*e}$ is a column vector, which represents the e-th bit of all the training samples. Likewise, $\mathbf{Q} \in \mathbb{R}^{c \times n}$ represents facial attribute feature matrix and $\mathbf{Q}_{*j} = q(\mathbf{w}_y, \mathbf{y}_{\mathrm{j}})$ is the MLP feature for attribute modality $\mathbf{y}_{\mathrm{j}}$; $(\mathbf{Q}^{\mathsf{T}})_{*f}$ is a column vector, which represents the f-th bit of all the training samples. The objective function in (5.1) needs to be minimized with respect to parameters $\mathbf{w}_x, \mathbf{w}_y$.

The first term in the objective function is margin-based distance logistic loss function, which tries to push modalities referring to the same sample closer to each other, while pushing the modalities referring to different samples away from each other. The term $p(\mathbf{P}_{*i}, \mathbf{Q}_{*j}) = \frac{1+\exp(-m)}{1+\exp(||\mathbf{P}_{*i} - \mathbf{Q}_{*j}|| - m)}$ represents the distance-based logistic probability (DBLP) and defines the probability of the match between the image modality feature vector $\mathbf{P}_{*i}$ and attribute modality feature vector $\mathbf{Q}_{*j}$, given their squared distance. The margin parameter m determines the extent to which matched or non-matched samples are attracted or repelled from each other, respectively. The distance-based logistic loss is then derived from the DBLP by using the cross entropy loss similar to the classification case: $\ell_c(r, s) = -s \log(r) + (s-1)\log(1-r)$.

Fig. 5.3 shows an illustration for the margin-based distance logistic-loss for our application. The dotted circle indicates the margin m to the boundary in terms of hamming

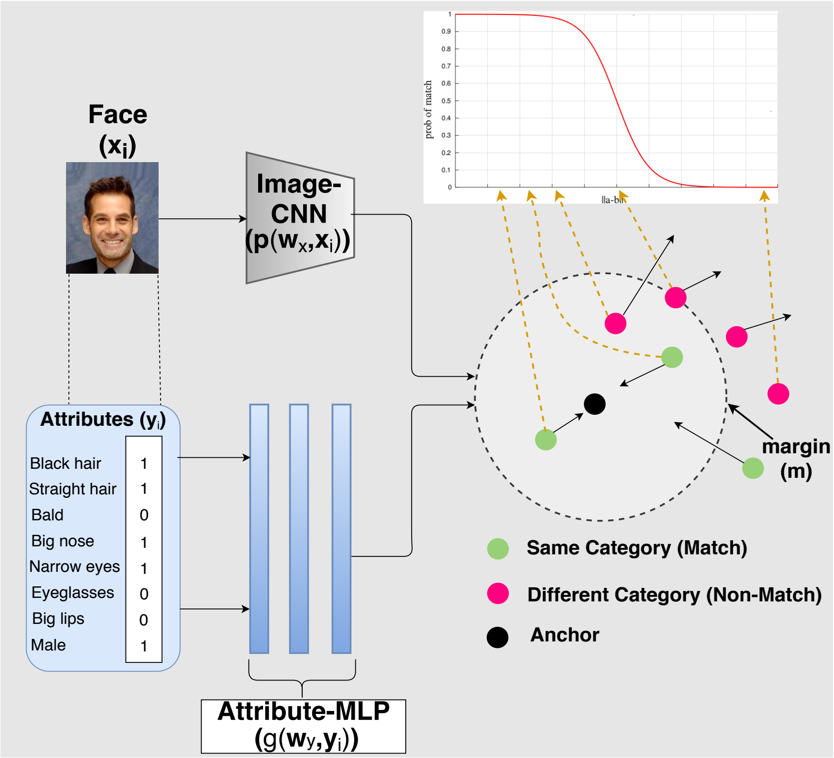


Figure 5.3: Diagram for the first stage showing the importance of margin-based distance logistic loss.

distance. The anchor (black solid circle) indicates a fixed instance and could either be an image hash code or an attribute hash code. The green solid circles indicate the matched instances (i.e., image or attribute hash codes belonging to the same subject as the anchor) and the magenta solid circles indicate non-matched instances (i.e., image or attribute hash codes belonging to different subject than the anchor). The function of margin-based DLL is two-fold. Firstly, it pushes the matched instances (green circles) away from the margin in the inward direction i.e., the hamming distance between the image and attribute hash codes of the same subject should be less than the margin m. Secondly, the margin-based DLL also pushes the non-matched instances (magenta circles) away from the margin in the outward direction i.e., the hamming distance between the image and attribute hash codes belonging to different subjects should be larger than the margin m. It has to be noted that after training the ADCMH network, the image hash codes and the attribute hash codes belonging to the same subject have a hamming distance less than margin m. This margin is a very important parameter in this framework and it will also affect the training of the NECD as is detailed in the description of Stage 1(b) in Sec. 5.2.2 .

The second term in the objective function is the quantization loss that denotes a constraint to boost the activations of the units in the hashing layer to be closer to -1 or 1. The value of $\mathbf{P}_{*i}$ is in the range of $[-1, 1]$ because it has been activated by the tanh activation. To make the codes closer to either -1 or 1, we add a quantization constraint of maximizing the sum of squared errors between the hashing layer activations and 0, which is given by $\sum_{i=1}^{n} ||\mathbf{P}_{*i} - \mathbf{0}||^2$, where n is the number of training images in a mini-batch and $\mathbf{0}$ is the c-dimensional vector with all elements equal to 0. However, this is equivalent to maximizing the square of the length of the vector formed by the hashing layer activations, that is $\sum_{i=1}^{n} ||\mathbf{P}_{*i} - \mathbf{0}||^2 = \sum_{i=1}^{n} ||\mathbf{P}_{*i}||^2$.

The third term, which is the entropy maximization loss, helps to obtain hash codes with an equal number of -1's and 1's, which maximizes the entropy of the discrete distribution and results in hash codes with better discrimination. Precisely, the number of +1 and -1 for each bit on all the training samples should be almost the same.

Learning the ADCMH parameters in Stage 1(a): We used an alternating minimization (optimization) algorithm to learn the ADCMH network parameters $\mathbf{w}_x$, and $\mathbf{w}_y$. In this algorithm, within each epoch, we learn one parameter with other parameters fixed.

Learning ($\mathbf{w}_x$) parameters for ADCMH : We use the back propagation algorithm to first optimize the CNN parameters $\mathbf{w}_x$ for the image modality by fixing the $\mathbf{w}_y$, and compute loss function gradient with respect to the output of image modality network as follows:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{P}_{*i}} = \sum_{j=1}^{n} \frac{\partial \ell_c(p(\mathbf{P}_{*i}, \mathbf{Q}_{*j}), S_{ij})}{\partial \mathbf{P}_{*i}} - \frac{2\theta}{c} \sum_{i=1}^{n} (\mathbf{P}_{*i}) + 2\lambda \sum_{i=1}^{c} (\mathbf{P}^{\mathsf{T}})_{*i}. \tag{5.2}$$

The gradient of the first term in Eq. 5.2 is calculated as:

$$\frac{\partial \ell_c(p(\mathbf{P}_{*i}, \mathbf{Q}_{*j}), S_{ij})}{\partial \mathbf{P}_{*i}} = \frac{-(1+\exp(-m))}{(1+\exp(||\mathbf{P}_{*i} - \mathbf{Q}_{*j}|| - m))^2} \times \left(\frac{S_{ij}}{p(\mathbf{P}_{*i}, \mathbf{Q}_{*j})} + \frac{1 - S_{ij}}{1 - p(\mathbf{P}_{*i}, \mathbf{Q}_{*j})} \right). \tag{5.3}$$

Next, we compute $\frac{\partial \mathcal{J}}{\partial \mathbf{w}_x}$ with $\frac{\partial \mathcal{J}}{\partial \mathbf{P}_{*i}}$ by using the chain rule ($\frac{\partial \mathcal{J}}{\partial \mathbf{w}_x} = \frac{\partial \mathcal{J}}{\partial \mathbf{P}_{*i}} \times \frac{\partial \mathbf{P}_{*i}}{\partial \mathbf{w}_x}$), based on which the back propagation is used to update the parameter $\mathbf{w}_x$.

Learning ($\mathbf{w}_y$) parameters for ADCMH : Similar to the previous optimization, we use the back propagation algorithm to optimize the MLP network parameters $\mathbf{w}_y$ for the facial attribute modality by fixing $\mathbf{w}_x$, and compute the loss function gradient with respect to the output of the facial attribute network as follows:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{Q}_{*j}} = \sum_{i=1}^{n} \frac{\partial \ell_c(p(\mathbf{P}_{*i}, \mathbf{Q}_{*j}), S_{ij})}{\partial \mathbf{Q}_{*j}} - \frac{2\theta}{c} \sum_{j=1}^{n} (\mathbf{Q}_{*j}) + 2\lambda \sum_{j=1}^{c} (\mathbf{Q}^{\mathsf{T}})_{*j}. \tag{5.4}$$

Next, we compute $\frac{\partial \mathcal{J}}{\partial \mathbf{w}_y}$ with $\frac{\partial \mathcal{J}}{\partial \mathbf{Q}_{*j}}$ by using the chain rule ($\frac{\partial \mathcal{J}}{\partial \mathbf{w}_y} = \frac{\partial \mathcal{J}}{\partial \mathbf{Q}_{*j}} \times \frac{\partial \mathbf{Q}_{*j}}{\partial \mathbf{w}_y}$), based on which the back propagation algorithm is used to update the parameters $\mathbf{w}_y$.

The intermediate hash codes after training of the ADCMH are obtained by $\mathbf{c}_i^{(\mathbf{x}_i)} = \text{sign}(p(\mathbf{w}_x, \mathbf{x}_i))$ and $\mathbf{c}_i^{(\mathbf{y}_i)} = \text{sign}(q(\mathbf{w}_y, \mathbf{y}_i))$ for image $\mathbf{x}_i$ and attribute $\mathbf{y}_i$, respectively for a new sample outside of the training set.

Stage 1(b): Training Neural Error Correcting Decoder

As mentioned previously, there is room for improvement of cross-modal retrieval efficiency by reducing the Hamming distance between the intermediate hash codes of different modalities for the same subject. This can be achieved by using an ECC decoder. The concept of ECC decoder is illustrated in Fig. 5.4(a). It is observed from the figure that if the received codewords or the corrupted codewords (green solid circles) fall within the error correcting capability e of the ECC, then the received codeword will be decoded to the correct codeword (black solid circle) by the ECC decoder.

The image or the attribute intermediate hash codes generated by the ADCMH network can be considered to be corrupted codewords within a certain distance d of a correct codeword of an ECC. If this distance d is within the error-correcting capability e of the ECC, then the ECC decoder will decode the intermediate hash codes to corresponding correct codeword of the ECC. However, decoding the intermediate hash codes to a correct codeword does not assure an improvement in cross-modal retrieval efficiency. For improving the retrieval efficiency, in addition to the intermediate hash codes being decoded to the correct codeword of the ECC, we also require the face and attribute intermediate hash codes of the same subject to be decoded to the same codeword.

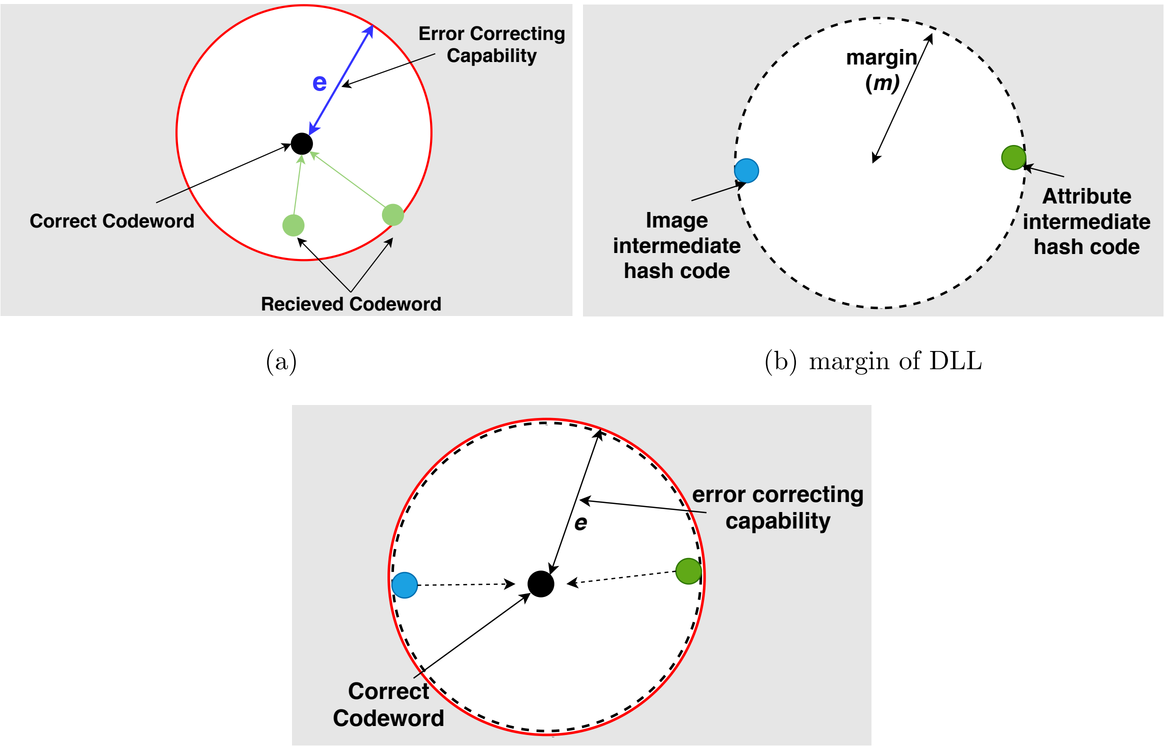
For fulfilling the above requirement, we exploit the error-correcting capability of the ECC decoder and relate it to the margin m used in the DLL for training the ADCMH in Stage 1(a).

Consider the Fig. 5.4(b) which shows the circle representing the margin for the DLL loss in Stage 1(a). The blue and green circles on the margin represent the image intermediate hash code and attribute intermediate hash code, respectively of the same subject. Now consider the Fig. 5.4(c), which shows the margin circle (black dashed line) along with the hamming sphere of the ECC (red solid line) with error correcting capability of e. We notice that the blue and green circle (i.e., image intermediate hash code and attribute intermediate hash code) fall within the error correcting capability of the ECC. Consequently, the image and attribute intermediate hash codes will be decoded to the same correct codeword (i.e. black small circle in the center) by the ECC decoder. This scenario is feasible only when error correcting capability e of the ECC decoder is at-least equal to the margin m used for DLL. Therefore, we chose an ECC decoder in such a way that the error correcting capability of the ECC $e \geq m$.

Recently, some excellent neural network based ECC decoders have been proposed [43–45], which have achieved close to the maximum likelihood (ML) performance for an ECC. These methods can be leveraged to generate high-quality and efficient hash codes. Thus, we can adapt such a neural network based ECC decoder, train it, and use it as an ancillary component to refine the intermediate hash codes generated by ADCMH.

Neural error-correcting decoder: The NECD is a non fully-connected neural network and can be considered to be a trellis representation of the BP decoder in which the nodes of the hidden layer correspond to the edges in the Tanner graph. Let N be the size of the codeword (i.e., the number of variable nodes in the Tanner graph) and E be the number of the edges in the Tanner graph. This implies the input layer of our NECD decoder is a vector of size N, that consists of the log-likelihood ratios (LLRs) of the channel outputs. All the hidden layers of the NECD have size E. A node in each hidden layer is associated with the message transmitted over some edge in the Tanner graph. The output layer of the NECD contains N nodes that output the final decoded codeword.

The number of hidden layers in the NECD depends upon the number of iterations considered for the BP algorithm. One iteration corresponds to message passing from the variable node to the check node and again back from check node to the variable node. Let us consider L iterations of the BP decoder. Then the number of hidden layers in the NECD would


Error Correcting Capability
e
Correct Codeword
Recieved Codeword
margin (m)
Image intermediate hash code
Attribute intermediate hash code
error correcting capability
e
Correct Codeword

(a)

(b) margin of DLL

(c) margin and hamming sphere

Figure 5.4: Relating the error-correcting capability of ECC to the margin of DLL.

be equal to $2L$. Consider the i-th hidden layer, where $i = 1, 2, \cdots, 2L$. For odd (even, respectively) values of i, each node in this hidden layer of the NECD corresponds to an edge $e = (v, c)$ connecting the variable node v (check node c, respectively) to the check node c (variable node v, respectively) and the output of this node represents the message transmitted by the BP decoder over that edge in the Tanner graph.

Next, we will discuss the way the NECD node connections are formed. A node in the first hidden layer (i.e., $i = 1$) corresponding to the edge $e = (v, c)$ is connected to a single node in the input layer, which is the variable node v associated with that edge. A processing node in the hidden layer i, where $i > 1$ and i is odd (even, respectively), corresponds to the edge $e = (v, c)$, and is connected to all the nodes in the layer $i - 1$ associated with the edges $e' = (v, c')$ for $c' \neq c$ ($e' = (v', c)$ for $v' \neq v$, respectively). For an odd node i, a processing node in layer i, corresponding to the edge $e = (v, c)$ is also connected to the vth input node. We denote by $x_{i,e}$, the output of a processing node in the hidden layer i. In terms of the BP decoder, for an odd (even, respectively) i, this is the message produced after

$\lfloor (i-1)/2 \rfloor$ iterations, from variable to check (check to variable, respectively) node. For odd i and $e = (v, c)$, we can use

$$x_{i,e=(v,c)} = \tanh\left(\frac{1}{2}\left(\mathbf{w}_{i,v} l_v + \sum_{e'=(v,c'),c'\neq c} \mathbf{w}_{i,e,e'} x_{i-1,e'}\right)\right), \tag{5}$$

under the initialization, $x_{0,e'} = 0$ for all e' (there is no information in the parity check nodes in the beginning). The summation in (5) is over all the edges $e' = (v, c')$ with variable node v except for the target edge $e = (v, c)$. $\mathbf{w}$ corresponds to the weight parameters of the neural network. Similarly, for even i and $e = (v, c)$, we use

$$x_{i,e=(v,c)} = 2\tanh^{-1}\left(\prod_{e'=(v',c),v'\neq v} x_{i-1,e'}\right). \tag{6}$$

The final vth output of the network is given by

$$o_v = \sigma\left(\mathbf{w}_{2L+1,v} l_v + \sum_{e'=(v,c')} \mathbf{w}_{2L+1,v,e'} x_{2L,e'}\right), \tag{7}$$

where $\sigma(x) = (1+e^{-x})^{-1}$ is a sigmoid function and is added so that the final network output is in the range $[0, 1]$.

In Stage 1(b), we build and train the NECD to be useful for correcting the intermediate hash codes generated by ADCMH in Stage 1(a). In this regard, we select the ECC to build the NECD is such a way that the error correcting capability e of the ECC is at-least equal to the margin m. Based on this condition. We then train our NECD using a dataset of zeros codeword.

Stage 2: Correcting the Intermediate Hash Codes

As shown in Fig. 5.2, the trained ADCMH network and the trained NECD from Stage 1(a) and Stage 1(b), respectively, are utilized in Stage 2 as shown by the red unidirectional arrows. In Stage 2, we use the same dataset used in Stage 1(a) and generate the real-valued vector output of the trained Image-CNN and trained Attribute-MLP (from Stage 1(a)) without thresholding using sign function. This output from both the networks is then passed through the trained NECD from Stage 1(b) to generate the corresponding image and attribute final binary hash codes. Next, the corresponding image and attribute final hash codes are used as target outputs (i.e., ground truths) to fine-tune the Image-CNN and Attribute-MLP, respectively from Stage 1(a). As shown in Fig. 5.2, for fine-tuning the Image-CNN and Attribute-MLP, we use the intermediate hash codes (real-valued vector output of the trained Image-CNN and trained Attribute-MLP) as our predicted outputs and use the corresponding final hash codes (binary) as our ground truths. We use cross-entropy as loss function to fine-tune the Image-CNN and Attribute-MLP in Stage 2.

$$L_C(y,p) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} y_j^{(i)} log(p_j^{(i)}) + (1 - y_j^{(i)}) log(1 - p_j^{(i)}), \tag{5.5}$$

where $y_j^{(i)}$ is the final hash code value for the *i-th* training sample and *j-th* element in the output layer. Similarly, $p_j^{(i)}$ is the intermediate hash code value for the *i-th* training sample and *j-th* element in the output layer. N signifies the number of training samples in a mini-batch, k signifies the hash code length, and $L_C(y,p)$ defines the cross entropy loss function between the intermediate hash code vector p and final hash code vector y.

In this stage of the training algorithm, the cross entropy loss between intermediate hash codes generated by ADCMH and the corrected codewords obtained by NECD, is back-propagated to the ADCMH network to update its parameters (i.e., $\mathbf{w}_x$ and $\mathbf{w}_y$). As mentioned earlier, intermediate hash codes used in this Stage are real valued for both modalities. Here, we formulate Stage 2 of the training algorithm, which is used to update the ADCMH parameters to generate the corrected codes.

As shown in Fig. 5.2, the NECD network is used to correct the intermediate codes generated from both modalities. Assume that $\mathbf{t}_i^{(\mathbf{x}_i)}$ is the output of the NECD network used to correct the intermediate hash code $\mathbf{c}_i^{(\mathbf{x}_i)}$ associated with the sample $\mathbf{x}_i$ for image modality. The overall cross-entropy loss on all the samples is calculated as: $\mathcal{H}_p = L_C(\mathbf{t}_i^{(\mathbf{x}_i)}, \mathbf{c}_i^{(\mathbf{x}_i)})$. To back-propagate the loss of the error-corrected code to the ADCMH parameters related to the image modality (i.e., $\mathbf{w}_x$), we can use the chain rule as discussed in the previous section to update $\mathbf{w}_x$, i.e., $(\frac{\partial \mathcal{H}_p}{\partial \mathbf{w}_x} = \frac{\partial \mathcal{H}_p}{\partial \mathbf{P}_{*i}} \times \frac{\partial \mathbf{P}_{*i}}{\partial \mathbf{w}_x})$.

Likewise, the NECD network is used to update the ADCMH network parameters that are related to the facial attribute modality (i.e., $\mathbf{w}_y$). Therefore, we can formulate it as : $\mathcal{H}_q = L_C(\mathbf{t}_i^{(\mathbf{y}_i)}, \mathbf{c}_i^{(\mathbf{y}_i)})$. To back-propagate the loss of the error-corrected code to update

$\mathbf{w}_y$, we use the chain rule as $(\frac{\partial \mathcal{H}_q}{\partial \mathbf{w}_y} = \frac{\partial \mathcal{H}_q}{\partial \mathbf{Q}_{*i}} \times \frac{\partial \mathbf{Q}_{*i}}{\partial \mathbf{w}_y})$. Note that before back-propagating the cross-entropy loss to update the ADCMH parameters, we scale the obtained loss by a hyper-parameter γ to create a balance between cross-entropy loss and the cross-modal hashing loss (i.e., $\mathcal{J}$ in (5.1)) that is defined in step 1 of the training algorithm.

Note that we utilize alternative optimization algorithm and perform the Stage 1(a) and Stage 2 of the training algorithm iteratively until there is not a significant improvement in retrieval efficiency on the training set. Additionally, note that we use the same trained NECD from Stage 1(b) in Stage 2 for both the Image-CNN and the Attribute-MLP. Therefore, if the NECD in Stage 1(b) has been created using ECC with an error-correcting capability of $e \geq m$, where m is the margin used for DLL in Stage 1(a), then the final hash codes would be the same for both the image and attribute modality at the end of the training of the alternative optimization algorithm. The complete algorithm is given in Fig. 5.2.

Out-of-Sample Extension

After training DNDCMH to convergence (i.e., no improvement in accuracy on the training set), for a new instance that is not in the training set, we can easily generate its error-corrected hash code as long as we can get one of its modalities. Given a query data point with image modality $\mathbf{x}_i$, we directly use it as the input of the Image-CNN part of the trained DNDCMH, then forward propagate the query through the network to generate its hash code as: $\mathbf{t}_i^{(\mathbf{x}_i)} = \text{sign}(p(\mathbf{w}_x, \mathbf{x}_i))$. Similarly, for the Attribute-MLP, we generate the hash code of a data point with only attribute modality $\mathbf{y}_i$ as: $\mathbf{t}_i^{(\mathbf{y}_i)} = \text{sign}(q(\mathbf{w}_y, \mathbf{y}_i))$.

5.3 Experimental Results

Implementation: As mentioned previously, our proposed ADCMH is composed of two networks: An image convolutional neural network (Image-CNN), which is used to extract features for the image modality and an attribute multi-layer perceptron (Attribute-MLP), which is used to extract features for the facial attribute modality. We initialize our Image-CNN parameters with a VGG-19 [22] network pre-trained on the ImageNet [125] dataset. In order to help the Image-CNN learn better and specialized set of facial features, we fine-tune

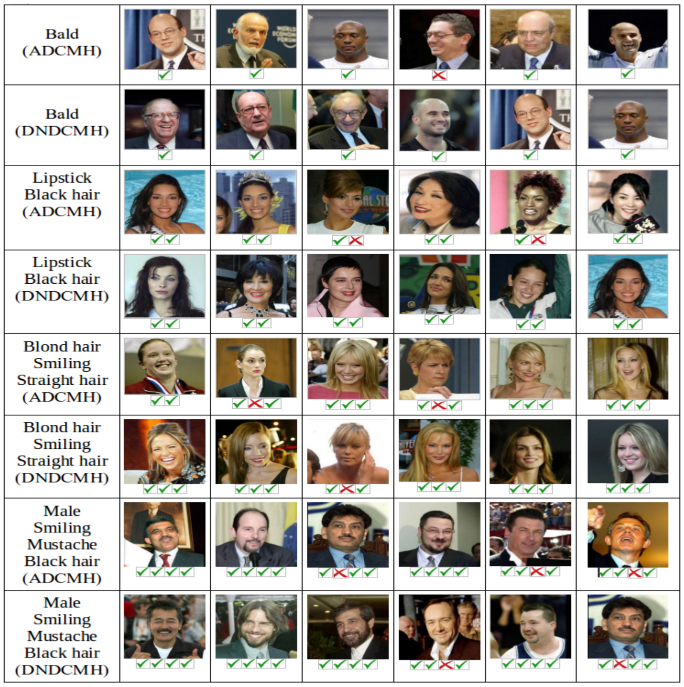

Bald (ADCMH)
Bald (DNDCMH)
Lipstick Black hair (ADCMH)
Lipstick Black hair (DNDCMH)
Blond hair Smiling Straight hair (ADCMH)
Blond hair Smiling Straight hair (DNDCMH)
Male Smiling Mustache Black hair (ADCMH)
Male Smiling Mustache Black hair (DNDCMH)

Figure 5.5: Qualitative results: Retrieved images using DNDCMH and ADCMH by giving different combinations of facial attributes as a query. Tick and cross symbols indicate the correct and wrong image retrieval from the testing set, respectively.

it as a classifier by using the CASIA-Web Face dataset, which contains 10,575 subjects and 494,414 images. The original VGG-19 consists of five convolutional layers ($conv1 - conv5$) and three fully-connected layers ($fc6 - fc8$). We discard the $fc8$ layer and replace the $fc7$ layer with a new f_{ch} layer with c hidden nodes, where c is the required intermediate hash code length. For the convolutional layers and $fc6$ layer we have used ReLU activation, for the f_{ch} layer, we have used tanh activation. The size of the intermediate hash length is equal to the size of the codeword used in the NECD network.

The Attribute-MLP contains three fully connected layers to learn the features for the facial attribute modality. To perform feature learning from the attributes, we first represent the attributes of each training sample as a binary vector, which indicates the presence or absence of corresponding facial attribute. This binary vector serves as a facial attribute vector and is used as an input to the Attribute-MLP. The first and second layers in the MLP network contain 512 nodes with ReLU activation and the number of nodes in the last fully connected layer is equal to the intermediate hash code length c with tanh activation. The weights of the MLP network are initialized by sampling randomly from $\mathcal{N}(0, 0.01)$ except for the bias parameters that are initialized with zeros. We use the Adam optimizer [101] with the default hyper-parameter values ($\epsilon = 10^{-3}$, $\lambda_1 = 0.9$, $\lambda_2 = 0.999$) to train all the parameters using alternative minimization approach. The batch size in all the experiments is fixed to 128. ADCMH is implemented in TensorFlow with python API and all the experiments are conducted on two GeForce GTX TITAN X 12GB GPUs.

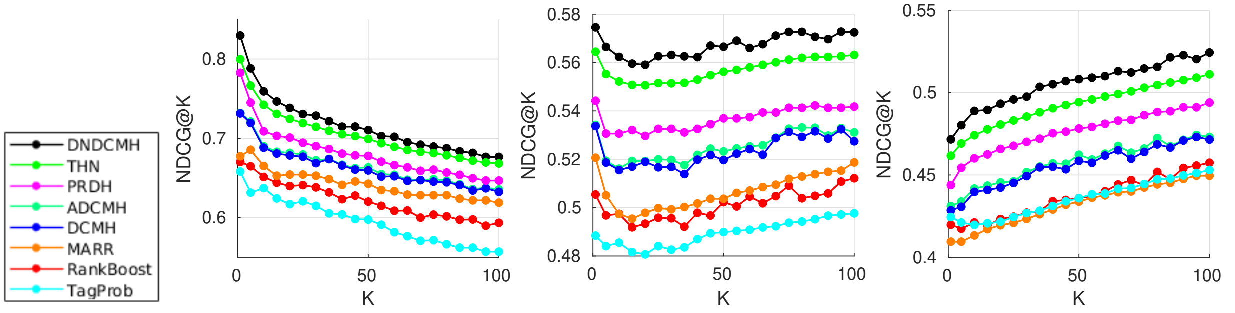
To train this NECD, it is sufficient to use training database constructed using noisy versions of a single codeword [43]. For convenience, we use noisy versions of zero codeword and our training database for NECD contains different channel output realization when the zero codeword is transmitted. The goal is to train NECD to attain N dimensional output word, which is as close as possible to the zero codeword by train the NECD parameters. We have trained our NECD on several codes including BCH(31,21), BCH(63,45), and BCH(127,92). This implies that the intermediate and the target hash code length for our experiments is equal to 31, 63, and 127. Note that the exact codeword size (N, k) depends on the error correcting capability e required for the NECD. Again, the error correcting capability e depends on the margin m of the DLL function used to train the ADCMH, such that $e \geq m$.

Datasets: We evaluated our proposed DNDCMH framework on two face datasets including the Labeled Faces in the Wild (LFW) [126] and Large-scale CelebFaces Attributes (CelebA) Dataset [127] annotated by facial attributes. **LFW** is a notable face database of more than 13,000 images of faces, created for studying the problem of unconstrained face recognition. The faces are collected from the web, detected and centered by the Viola Jones face detector. **CelebA** is a large-scale face attribute and richly annotated dataset containing more than 200K celebrity images, each of which is annotated with 40 facial attributes. CelebA has about ten thousand identities with twenty images per identity on average. For comparison purposes, we have been consistent with the train and test split of these datasets as given on the dataset webpage.

Baselines: We have compared the retrieval and ranking performance of our system with some of the other state-of-the-art face image retrieval and ranking approaches including MARR [128], rankBoost [129], TagProp [130]. For fair comparison, we have exploited the VGG-19 architecture pretrained on Imagenet dataset, which is the same as the initial CNN

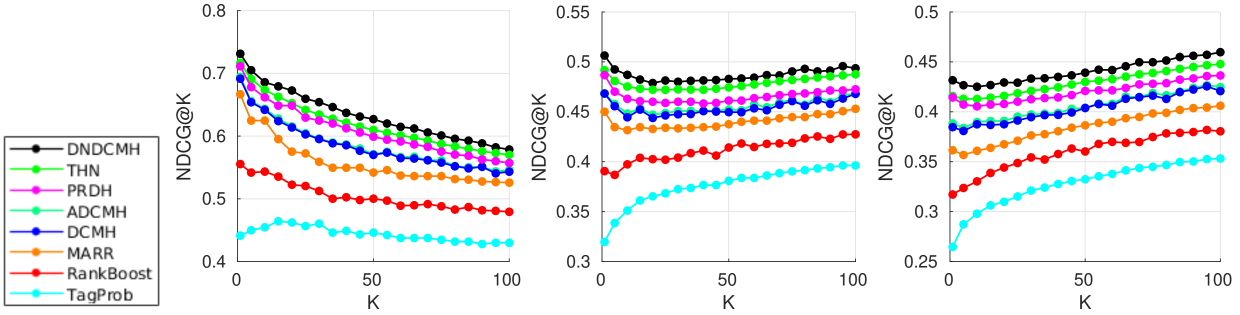
Table 5.1: MAP comparison tabulating the effect of the error correcting capability e for a given margin m. The given retrieval performance is for hash code length of 63 bits.

Margin	Code - Word Size	Error Correcting Capability	CelebA			LFW		
(m)	(N, K)	(e)	Single	Double	Triple	Single	Double	Triple
3	(63,51)	2	63.215	61.779	57.349	60.239	58.322	56.115
	(63,45)	3	71.831	**68.335**	66.131	**67.117**	**65.383**	**64.118**
	(63,39)	4	**72.10**	68.11	**66.63**	65.135	64.446	63.988
	(63,36)	5	66.543	64.671	62.156	61.90	57.783	55.9
	(63,30)	6	61.10	58.831	54.665	56.35	53.139	50.952
5	(63,39)	4	70.55	68.116	66.312	67.515	65.213	63.111
	(63,36)	5	76.311	**74.132**	**70.877**	**71.31**	**69.156**	66.932
	(63,30)	6	**77.19**	72.066	69.992	70.482	67.354	**67.131**
	(63,24)	7	72.634	71.135	68.820	67.111	65.090	63.865
	(63,18)	10	60.225	58.113	56.622	58.35	56.751	54.322
6	(63,36)	5	69.121	66.398	63.1	64.334	63.901	63.178
	(63,30)	6	**79.125**	73.167	**72.913**	**74.873**	**71.242**	**70.643**
	(63,24)	7	76.339	**73.842**	70.190	72.1	70.908	67.231
	(63,18)	10	63.12	62.781	60.613	61.349	60.444	57.609
	(63,16)	11	55.1	56.789	53.334	52.18	50.981	50.4
7	(63,30)	6	77.351	75.568	72.2211	73.181	71.362	71.61
	(63,24)	7	**83.131**	**79.354**	**77.116**	**77.335**	**74.952**	**71.118**
	(63,18)	10	76.541	73.396	70.751	69.117	66.354	62.182
	(63,16)	11	72.225	69.181	66.192	67.111	65.332	63.981
	(63,10)	13	62.25	60.181	58.333	58.05	56.351	54.119



(b) Single (c) Double (d) Triple

Figure 5.6: Ranking performance on the CelebA dataset. Due to space restriction, the legend is shown in the box on the left.



(b) Single (c) Double (d) Triple

Figure 5.7: Ranking performance on the LFW dataset. Due to space restriction, the legend is shown in the box on the left.

of the image modality in DNDCMH, to extract CNN features. All the above baselines are trained based on these CNN features. Additionally, we have also compared our algorithm with the state-of-the-art deep cross modal hashing algorithms DCMH [38], Pairwise Relationship Guided Deep Hashing for Cross-Modal Retrieval (PRDH) [37], and Triplet-based Hashing Network (THN) [123]. We have also compared the DNDCMH framework with only the ADCMH network; i.e., training using only Stage 1(a).

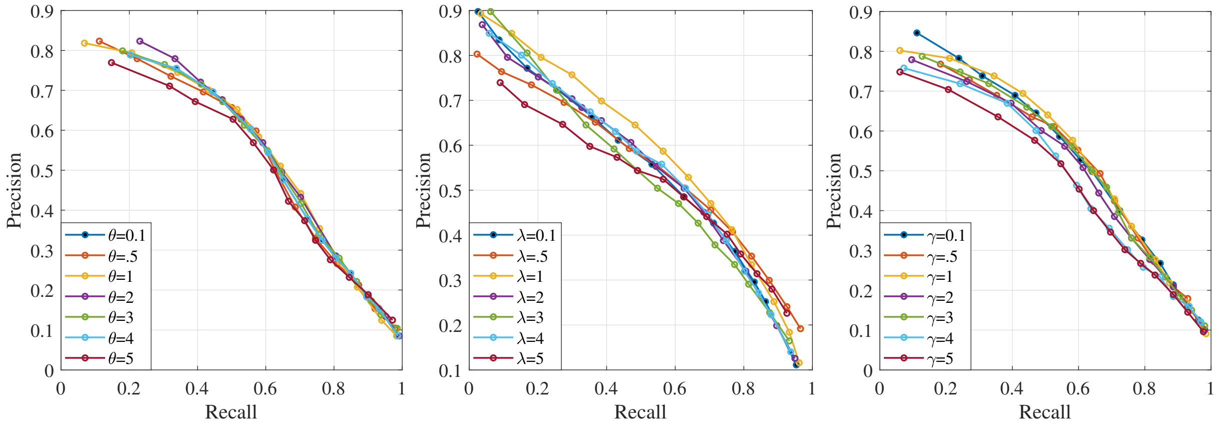
Evaluation Protocols: For hashing-based retrieval, Hamming ranking is a widely used retrieval protocol and we will also evaluate our DNDCMH method and compare it with other baselines using this protocol. The Hamming ranking protocol ranks the points in the database (retrieval set) according to their Hamming distances to a given query point, in an increasing order. Mean average precision (MAP) is the widely used metric to measure the accuracy of the Hamming ranking protocol.

Similarly, for ranking protocol we use normalized discounted cumulative gain (NDCG) to compare ranking performance of DNDCMH with other baselines. NDCG is a standard single-number measure of ranking quality that allows non-binary relevance judgments. It is defined as NDCG@$k = \frac{1}{Z}\sum_{i=1}^{k}\frac{2^{\mathsf{rel}(i)}-1}{\log(i+1)}$, where $\mathsf{rel}(i)$ is the relevance of the i^{th} ranked image and Z is a normalization constant to ensure that the correct ranking results in an NDCG score of 1.

Effect of relation between margin m and error correcting capability e on the retrieval performance: Before extensive performance evaluation of the proposed framework, it is very important to understand how the relation between error-correcting capability e of the NECD used in Stage 1(b) and the margin m of the margin-based distance logistic loss (DLL) in Stage 1(a) affects the overall retrieval performance for the proposed framework. From Sec. 5.2.2, we know that $e \geq m$. For a given value of m we need to analyze the upper limit of e that will give us a reasonable retrieval performance. We use mean average precision (MAP) as our performance metric to evaluate the relation between e and m. Table 5.1 provides the MAP results for both the CelebA and the LFW dataset. We have considered hash code length of 63 bits with margin values as $m = 3, 5, 6, 7$. Table 5.1 also provides the different values of error correcting capability e we have used for a given margin m. We have tried the values of e in the range of $m - 1$ to $m + 6$ depending on the error correcting code word (N, K) possible with hash code length of 63 bits.

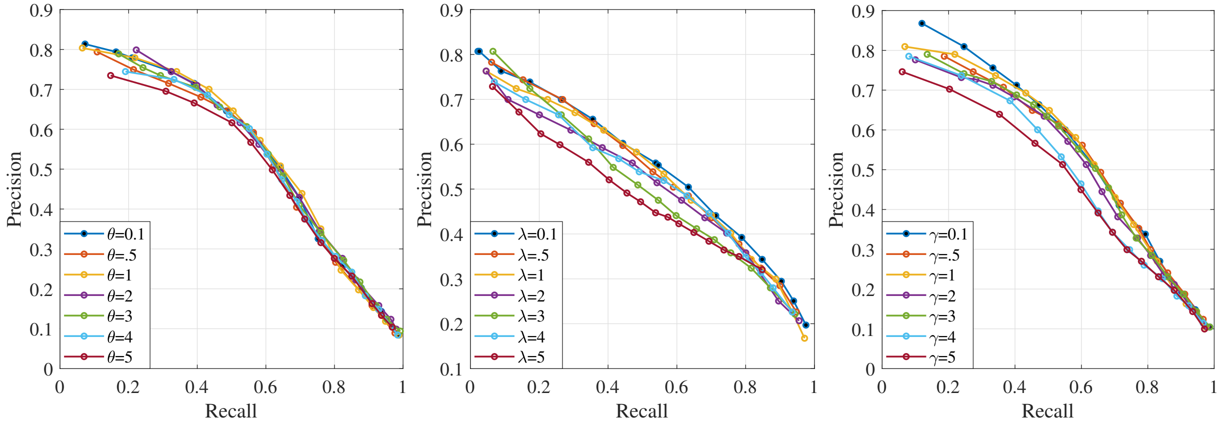
We can observe from Table 5.1 that for an given value of m, the best MAP is achieved when e is equal to or slightly greater than m. For example, when $m = 5$, we can observe that the best MAP is achieved when e is either 5 or 6. It is interesting to note that as e becomes greater and greater than m, the MAP starts reducing. The reason for this reduction is that as the error correcting capability e increases, more and more impostors will fall within the Hamming sphere (error correcting capability), leading to more false positives and lower precision.

Retrieval Performance: MAP results using different number of attributes in the query for DNDCMH, ADCMH and other baselines for the LFW and FaceTracer datasets is given in Table 5.2. For this experiment, we have used a margin of $m = 6$ for margin-based



(a) P-R curve for θ (b) P-R curve for λ (c) P-R curve for γ

Figure 5.8: Influence of Hyper-parameters on P-R curves for CelebA dataset.



(a) P-R curve for θ (b) P-R curve for λ (c) P-R curve for γ

Figure 5.9: Influence of Hyper-parameters on P-R curves for LFW dataset.

DLL. For the best results, we have trained our NECD with BCH(63,30) code, which implies that the hash code length is equal to 63 bits and the error correcting capability e of the NECD is equal to 6, which implies $e = m$. We can clearly see that our DNDCMH method clearly outperforms all the other baseline methods including the ADCMH. An interesting observation is that our method ADCMH with no NECD also outperforms DCMH, which shows that the distance-based logistic loss used in our objective function in (5.1) is better than the negative log-likelihood loss used in DCMH. Also, the addition of NECD to ADCMH, which is our proposed DNDCMH improves the retrieval performance and outperforms the other state-of-the-art deep cross-modal hashing methods PRDH and THN.

Table 5.2: MAP comparison for DNDCMH with other baselines for LFW and CelebA datasets using diff. number of attributes in the query. The best MAP is shown in boldface.

	CelebA			LFW		
Method	Single	Double	Triple	Single	Double	Triple
TagProp	52.610	48.494	41.509	51.776	46.556	39.790
RankBoost	55.116	51.354	50.672	53.320	50.198	48.259
MARR	61.334	57.890	56.098	59.343	54.266	55.334
DCMH	67.210	62.664	61.170	62.320	61.200	60.320
PRDH	72.100	68.550	66.110	66.764	65.776	64.37
THN	74.982	71.498	70.498	69.907	69.897	67.297
ADCMH	68.437	64.173	63.993	63.732	63.171	62.689
DNDCMH	**79.125**	**73.167**	**72.913**	**74.873**	**71.242**	**70.643**

Ranking Performance: Comparison of the NDCG scores, as a function of the ranking truncation level K, using different number of attribute queries are given in Fig. 5.6 and Fig. 5.7 for the LFW and FaceTracer dataset, respectively using hash code length of 63 and BCH (63,30) with $e = m = 6$ for DNDCMH and ADCMH. It is clear from the figures that our approach (DNDCMH) significantly outperforms all the baseline methods for all three types of queries, at all values of K. For example, for LFW dataset, at a truncation level of 20 (NDCG@20), for single, double and triple attribute queries, DNDCMH is respectively, 2.1%, 2.1% and 2.0% better than THN, the best deep cross-modal hashing method, and also DNDCMH is respectively, 11.2%, 7.3% and 8.0% better than MARR, the best shallow method for attribute-based image retrieval. The ranking performance using intermediate hash codes generated by only ADCMH with no NECD also outperforms the shallow methods MARR, RankBoost, and Tagprob, and also outperforms DCMH for double and triple attribute queries and is very close (may be slightly better) to DCMH performance for single-attribute queries. The better performance of DNDCMH when compared to other deep cross modal hashing method demonstrates the effectiveness of NECD in improving the performance for cross-modal retrieval. We can observe that the NDCG values for the FaceTracer dataset for all methods are relatively lower when compared to the LFW dataset. This is due to the difference in the distributions of the two datasets.

Parameter Sensitivity: We explore the influence of the hyper-parameters θ, λ, and

Table 5.3: MAP comparison for DNDCMH with ADCMH using different # of bits.

Task	*Method*	CelebA			LFW		
		31 bits	63 bits	127 bits	31 bits	63 bits	127 bits
Single Attribute	ADCMH	68.158	68.437	68.513	63.325	63.732	63.917
	DNDCMH	78.917	79.125	79.565	74.132	74.873	75.032
Double Attribute	ADCMH	64.121	64.173	64.395	63.121	63.171	63.532
	DNDCMH	74.152	73.167	75.921	71.112	71.242	72.056
Triple Attribute	ADCMH	34.577	63.993	64.634	62.664	62.689	62.754
	DNDCMH	72.152	72.913	73.271	70.216	70.643	70.855

γ. Fig. 5.8 and Fig. 5.9 show the precision-recall results on the LFW and the FaceTracer datasets, respectively with different values of θ, λ, and γ, where the code length is 63 bits and $e = m = 6$. We can see that DNDCMH is not sensitive to θ,λ, and γ with $0.1 < \theta < 5$,$0.1 < \lambda < 5$, and $0.1 < \gamma < 5$.

Effectiveness of the NECD for Improving the ADCMH Retrieval Performance: To show the effectiveness of NECD combined with the ADCMH network, we conducted experiments using two different models: a) ADCMH, which indicates the case where we train the model only using Stage 1(a) optimization without including NECD in the training, specifically this case considers cross-modal hashing based on entropy, quantization and distance-based logistic loss; b) DNDCMH indicates our overall model where we include NECD and use iterative alternate optimization to correct the generated codes by ADCMH, qualitative results shown in Fig.5.5 indicate that the ADCMH retrieval performance is improved by integrating together NECD and ADCMH.

In addition to the qualitative results, we have also compared our DNDCMH with ADCMH using MAP by varying the hash code length. Table 5.3 provides the MAP comparison for DNDCMH and ADCMH for different hash code lengths (31, 63, and 127). For hash code length of 31, 63, and 127 we have used $e = m = 3$, $e = m = 6$, and $e = m = 11$, respectively. For ADCMH, we have used the same margin value as DNDCMH. The results in the Table shows DNDCMH gives much better results than ADCMH, which implies that additional optimization using NECD improves retrieval performance for ADCMH. Additionally, the retrieval performance does not change a lot with increase in the hash code length. Consequently, even with low storage capacity of 31 bits, high retrieval performance is achieved.

Effect of Number of Attributes: From the experimental results, we can see that the retrieval or the ranking performance for DNDCMH and also ADCMH decreases with the increase in the number of facial attributes as query. This is evident from the quantitative results in Fig. 5.5 and also quantitative results in Fig. 5.6, and Fig. 5.7. The reason for this decrease is that as we increase the number of facial attributes in a query, the number of constraints to map the facial image modality into the same Hamming space as the attribute modality, also gets inflated, which leads to lower retrieval performance when compared to only a small number of facial attribute in the query.

5.4 Summary

In this chapter, we proposed a novel iterative two-step deep cross-modal hashing method that takes facial attributes as query and returns a list of images based on a Hamming distance similarity. In this framework, we leveraged a neural network based decoder to correct the codes generated by the facial attribute-based deep cross-modal hashing to improve the retrieval performance. The experimental results show that the neural network decoder significantly improves the retrieval performance of the attribute-based deep cross-modal hashing network. Moreover, the results indicate that the proposed framework outperforms most of the other face image retrieval approaches.

Chapter 6

Attribute-Guided Coupled GAN for Cross-Resolution Face Recognition

In this chapter, we propose a novel attribute-guided cross-resolution (low-resolution to high-resolution) face recognition framework that leverages a coupled GAN structure with adversarial training to find the hidden relationship between the low-resolution and high-resolution images in a latent common embedding subspace.

6.1 Introduction

Facial biometrics is used in a variety of modern recognition and surveillance applications ranging from stand-alone camera applications in banks and supermarkets to multiple networked closed-circuit televisions in law enforcement applications or even in cloud-based authentication applications. [49, 51–53, 57, 65]. The large distance between surveillance cameras and the subjects leads to low-resolution (LR) face regions in the captured images. Usually, the discriminant properties of the face are degraded in the LR images, which leads to a significant drop in the accuracy of traditional face recognition algorithms developed for high-resolution (HR) images. An efficient face-recognition algorithm should perform well even for LR faces without significantly reducing recognition accuracy.

In comparison to HR face images, LR faces have their own unique visual properties. Although many visual features are missing in LR face images, humans are still able to notice similarities between the LR and HR face images of a given subject. This implies that the neural systems of the human brain is able to recover missing visual properties of LR faces if the human brain is familiar with the high-resolution image of that subject or a given identity [131]. Inspired by this fact, several LR face recognition models have been introduced that can be generally divided into two categories: the hallucination category and the embedding category. The models in the hallucination category reconstruct HR faces from LR faces before recognition [132–138]. The hallucination category of super-resolution is also used for other applications [78].

Methods based on hallucination usually achieve promising results in recognizing the reconstructed HR face images. However, the super-resolution operation in hallucination models usually requires significant additional computation that often translates to a reduction in the recognition speed. In contrast to methods based on the hallucination, methods in the embedding category extract features from LR faces by leveraging various external face contexts. Ren et al. [139] introduce a coupled kernel embedding to implicitly map face images with different resolutions into an infinite space. The recognition task is then performed in this new space to minimize the dissimilarities obtained by their kernel Gram matrices in the low and high-resolution spaces, respectively. Intuitively, the main step in the embedding method is to transfer knowledge from HR to LR face images. However, in these methods, one must be careful to transfer only the desired knowledge instead of transferring all knowledge from a HR domain to a LR domain.

In addition to knowledge sharing between the high and low-resolution images, soft biometric traits such as facial attributes can also be used as complimentary information to improve the cross-resolution face recognition model. Facial attributes have been previously used jointly with face biometrics in different face recognition applications [140].

In this chapter, we present an embedding model for cross-resolution face recognition based on novel attribute-guided deep coupled learning framework using GAN to find the hidden relationship between the features of high-resolution and low-resolution images in a latent common embedding subspace. The framework also utilizes CNN weight sharing followed by dedicated weights for learning the representative features for each specific face attribute. Specifically, our coupled framework exploits the facial attribute to further maximize the correlation between the low-resolution and high-resolution domains, which leads to a more discriminative embedding subspace to enhance the performance of the main task, which is the cross-resolution face recognition. Additionally, in our approach, we also predict the attributes for low-resolution images along with cross-resolution face recognition in a multi-tasking paradigm. Multi-task learning attempts to solve correlated tasks simultaneously by leveraging the knowledge sharing between the two tasks [58, 99, 141]. To summarize, our main contributions of this chapter are:

- A novel attribute guided cross-resolution (low-resolution to high-resolution) face recognition model using coupled GAN and multiple loss functions.
- A mutli-task learning framework to predict facial attributes for low-resolution facial images.
- Extensive experiments using four different datasets and a comparison of the proposed method with state-of-the-art methods.

6.2 Generative Adversarial Network

GANs have been widely used in different computer vision application such as style transfer, sketch to photo synthesis, and also in military applications [69, 70, 77, 142–144]. GANs consist of two competing networks, namely a generator G and discriminator D. The goal of GAN is to train the generator G to produce samples from training noise distribution $p_z(z)$ such that the discriminator D cannot distinguish the synthesized samples from actual data y with distribution p_{data}. Generator $G(z;\theta_g)$ is a differentiable function which maps the noise variable z to a data space using the parameters θ_g. On the other hand, discriminator $D(.;\theta_d)$ is also a differentiable function, which tries to discriminate using a binary classification between the real data y and $G(z)$. Specifically, the generator and discriminator compete with each other in a two-player minimax game to minimize the Jenson-Shannon divergence [145]. The loss function $L(D, G)$ for GAN is given as:

$$L(D,G) = E_{y\sim P_{data}(y)}[\log D(y)] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))] \tag{6.1}$$

The objective (two player minimax game) for GAN is given by:

$$\min_G \max_D L(D,G) = \min_G \max_D [E_{y \sim P_{data}(y)}[\log D(y)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))]] \tag{6.2}$$

Conditional GAN is another variant of GAN where both the generator and the discriminator are conditioned on an additional variable x. This additional variable could be any kind of auxilary information such as discrete labels [146], text [147], or images [148]. The loss function for conditional GAN is given as:

$$L_c(D,G) = E_{y \sim P_{data}(y)}[\log D(y|x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z|x)))]. \tag{6.3}$$

The objective for the conditional GAN is the same two player minimax game as in (6.2) with loss function as $L_c(D,G)$. Hereafter, we will denote the objective for conditional GAN as $O_{cGAN}(D,G,y,x)$, which is given by:

$$O_{cGAN}(D,G,y,x) = \min_G \max_D [E_{y \sim P_{data}(y)}[\log D(y|x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z|x)))]]. \tag{6.4}$$

6.3 Proposed Method

In this section, we describe the proposed method for cross-resolution face recognition. In contrast to the hallucination approach, we do not up-sample each low-resolution image to the high-resolution domain before matching. Instead, we seek to project the high and low-resolution images to a common latent low-dimensional embedding subspace using generative modeling. Inspired by the success of GANs [145], we explore adversarial networks in a multi-tasking paradigm to project low and high-resolution images to a common subspace for recognition, and also predict facial attributes from low recognition images.

As shown in Fig. 6.1, the proposed method consists of a coupled framework made of two sub-networks, where each sub-network is a GAN architecture made of a generator and a discriminator. The generators are coupled together using a contrastive loss function. Each generator is also responsible to predict attributes in a multi-tasking paradigm. In addition to the adversarial loss, and contrastive loss, we propose to guide the sub-networks using a perceptual loss based on the VGG 16 architecture and also an L_2 reconstruction

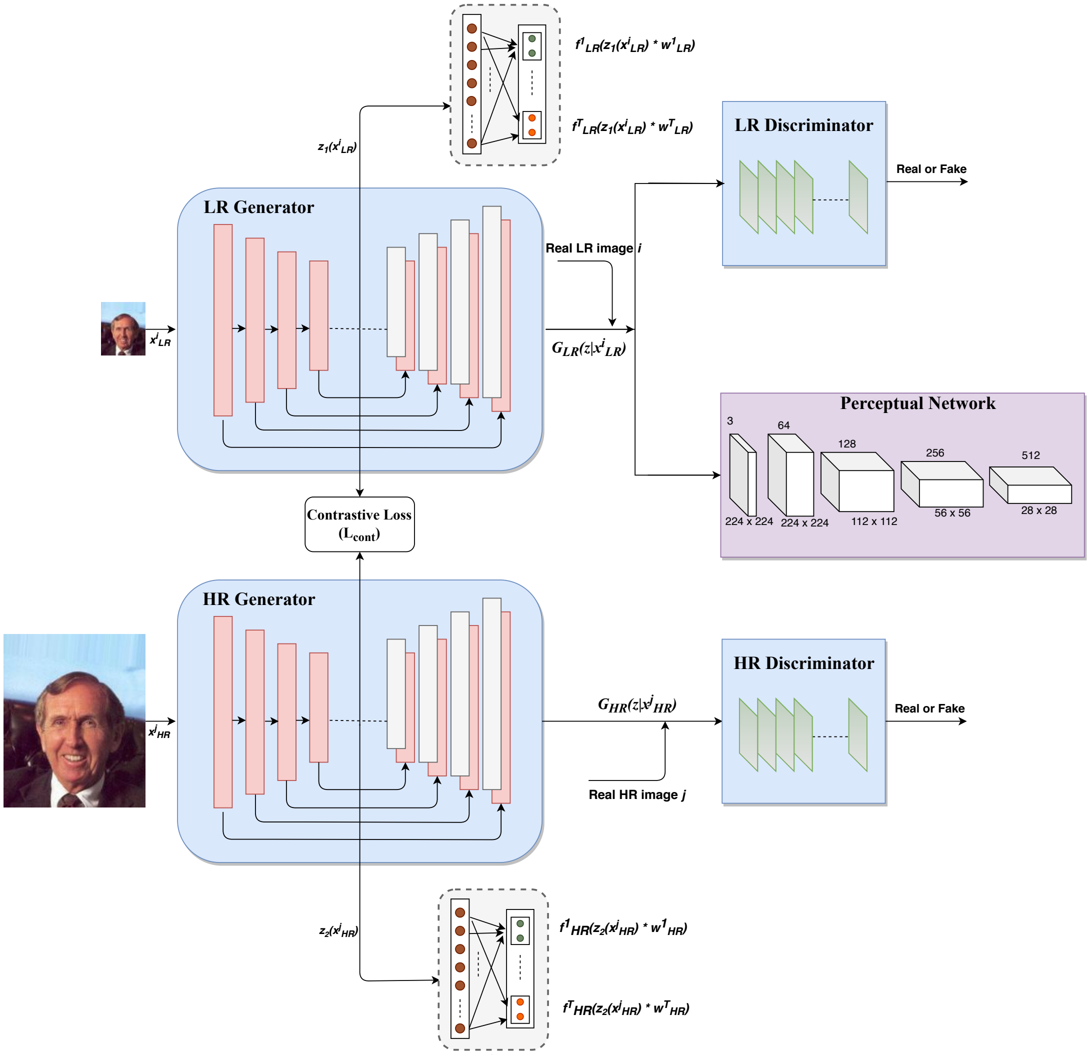


Figure 6.1: Block diagram of the proposed framework.

error. This is because the perceptual loss in optimization helps to achieve a realistic image reconstruction [149].

6.3.1 Deep Coupled Framework

The objective of our method is the recognition of low-resolution face images with respect to a gallery of high-resolution images, which have not been seen during the training. The matching of the low-resolution and the high-resolution images is performed in a common embedding subspace. For this reason, we use a coupled framework which contains two sub-networks: low-resolution (LR) network and high-resolution (HR) network. The LR network consists of a GAN (generator + discriminator), attribute predictor, and a perceptual network based on VGG-16, while the HR network consists of a GAN (generator + discriminator) and an attribute predictor.

For the generators, we have used a U-Net network [150] to better capture the low-level features and overcome the vanishing gradient problem due to deep network. Motivated by [148], we have used patch-based discriminators, which are trained iteratively along with the respective generators. Patch-based discriminator ensures preserving of high-frequency details which are usually lost when only L_1 loss is used. The final objective of our proposed method is to find the global deep latent features in a common embedding subspace representing the relationship between the low-resolution and their corresponding high-resolution face images. To find this common subspace between the two domains, we couple the two generators via a contrastive loss function L_{cont} [151].

This loss function (L_{cont}) is minimized so as to drive the genuine pairs (i.e., a LR image with its own corresponding HR image) towards each other in a common embedding subspace, and at the same time, push the impostor pairs (i.e., a LR image of a subject with another subject's HR image) away from each other. Let x^i_{LR} denote the input LR face image, and x^j_{HR} denote the input HR image. $c(i, j)$ is a binary label, which is equal to 0 if x^i_{LR} and x^j_{HR} belong to the same class (i.e., genuine pair), and equal to 1 if x^i_{LR} and x^j_{HR} belong to the different class (i.e., impostor pair). Let $z_1(.)$ and $z_2(.)$ denote the deep CNN-based embedding functions to transform x^i_{LR} and x^j_{HR}, respectively into a common latent embedding subspace. Then, contrastive loss function (L_{cont}) if $c(i, j) = 0$ (i.e., genuine pair) is given as:

$$L_{cont}(z_1(x^i_{LR}), z_2(x^j_{HR}), c(i,j)) = \frac{1}{2} \left\| z_1(x^i_{LR}) - z_2(x^j_{HR}) \right\|_2^2 . \tag{6.5}$$

Similarly if $c(i, j) = 1$ (i.e., impostor pair), then contrastive loss function (L_{cont}) is :

$$L_{cont}(z_1(x^i_{LR}), z_2(x^j_{HR}), c(i,j)) = \frac{1}{2} \max\Big(0, m - \left\| z_1(x^i_{LR}) - z_2(x^j_{HR}) \right\|_2^2 \Big), \tag{6.6}$$

where m is the contrastive margin and is used to "tighten" the constraint. Therefore, the total loss function for coupling the sub-networks is denoted by L_{cpl} and is given as:

$$L_{cpl} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} L_{cont}(z_1(x^i_{LR}), z_2(x^j_{HR}), c(i,j)), \tag{6.7}$$

where N is the number of training samples. The main motivation for using the coupling loss is that it has the capacity to find the discriminative embedding subspace because it uses the class labels implicitly, which may not be the case with some other metric such as Euclidean distance. This discriminative embedding subspace would be useful for matching of the LR images with the HR images and also for attribute prediction task.

6.3.2 Attribute Prediction Task

In addition to cross-resolution face recognition, another important objective of our proposed method is prediction of attributes using a LR or HR face image. However, separating these two objectives by learning multiple CNNs individually is not optimal since different objectives may share common features and have hidden relationship, which can be leveraged to jointly optimize the objectives. This notion of joint optimization has been used in [152], where they train a CNN for face recognition, and utilize the features for attribute prediction. Therefore, for this task, we use the respective feature set (i.e., $z_1(x^i_{LR})$ for LR, or $z_1(x^j_{HR})$ for HR) from the common embedding subspace to also predict the attributes for a given image. Also, our network shares a large portion of its parameters among different attribute prediction tasks in order to enhance the performance of the recognition task in a mutli-task paradigm.

For the attribute prediction task, a LR or HR image is given as input to the network to predict a set of attributes. Consider that the input is a LR image denoted by x^i_{LR}, where the class label for the image is given by $\ell^i \in L$ for $i = 1, \cdots, N$ where N is the number of training samples. Let's consider T to be the number of different facial attributes and $a^{i,t}$ denotes the ground truth attribute label for training sample i and attribute t for $t = 1 \cdots T$. In this case, using the feature set from the common embedding subspace, the attribute prediction loss function is given as:

$$L_{aLR} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} l(f^t_{LR}(z_1(x^i_{LR}) \times w^t_{LR}), a^{i,t}), \tag{6.8}$$

where $f^t_{LR}(.)$ is a binary classifier for the attribute t operated on the bottle-neck of LR generator as shown in Fig. 6.1. The classifier is learned by using a loss function l (e.g., cross entropy) and w^t_{LR} represents the weight parameters for the classifier and these parameters are learned separately for each facial attribute task.

Similarly, we can consider the other sub-network (HR network) and perform the same procedure with a HR image. The HR network also predicts the set of facial of attributes using the features $z_2(x^j_{HR})$ from the common embedding subspace for a given HR image. Therefore, the loss function for facial attribute prediction for HR image is given as:

$$L_{aHR} = \frac{1}{N}\sum_{j=1}^{N}\sum_{t=1}^{T} l(f^t_{HR}(z_2(x^j_{HR}) \times w^t_{HR}), a^{j,t}), \tag{6.9}$$

where the notations are similar to (6.8) but correspond to the HR network. The total attribute prediction loss given as:

$$L_a = L_{aLR} + L_{aHR}. \tag{6.10}$$

6.3.3 Generative Adversarial Loss

Let G_{LR} and G_{HR} denote the generators that synthesize the corresponding LR and HR images from the input LR and HR image, respectively. Let D_{LR} and D_{HR} denote the discriminators for LR and HR GANs respectively. We have utilized the GAN loss function [145] to train the generators and the corresponding discriminators in order to ensure that the discriminators cannot distinguish the synthesized images by the generators from the corresponding ground truth images. Also, it can be observed from Fig. 6.1, that the generators G_{LR} and G_{HR} try to generate a LR and HR image with the network conditioned on the input LR and HR image, respectively. The total loss for the coupled GAN is given by:

$$L_{GAN} = L_{LR} + L_{HR}, \tag{6.11}$$

where L_{LR} and L_{HR} denote the GAN loss functions for the LR and the HR network, respectively and are given as:

$$L_{LR} = O_{cGAN}(D_{LR}, G_{LR}, y^i, x^i_{LR}) \tag{6.12}$$

$$L_{HR} = O_{cGAN}(D_{HR}, G_{HR}, y^j, x^j_{HR}), \tag{6.13}$$

where function O_{cGAN} is given by (6.4). x^i_{LR} (x^j_{HR}) is the LR (HR) image used as a condition for the LR (HR) GAN and y^i (y^j) denotes the real LR (HR) data. Note that the real LR (HR) data y^i (y^j) and the network condition given by x^i_{LR} (x^j_{HR}) are the same.

6.3.4 Perceptual Loss

Perceptual loss function was introduced in [149] for style transfer and super-resolution. In [149], instead of relying only on L_1 or L_2 reconstruction error, the network parameters are learned using errors between high-level image feature representations extracted from a pre-trained convolutional neural network. Similarly, in our proposed approach, perceptual loss is added only to the LR network using a pre-trained VGG-16 [22] network to extract high-level features (ReLU3-3 layer) and the L_1 distance between these features of real and synthesized images is used to guide the generator G_{LR}. The perceptual loss for features for only the LR network is:

$$L_{P_{LR}} = \frac{1}{C_p W_p H_p} \sum_{c=1}^{C_p} \sum_{w=1}^{W_p} \sum_{h=1}^{H_p} \left\| V(G_{LR}(z|x^i_{LR}))^{c,w,h} - V(y^i)^{c,w,h} \right\|, \tag{6.14}$$

where y^i is the ground truth LR image, $G_{LR}(z|x^i_{LR})$ is the output of the LR generator. $V(.)$ represents a particular layer of the VGG-16 network, where the layer dimensions are given by C_p, W_p, and H_p. We have applied the perceptual loss only for the LR network to generate more sharper LR images helpful for recognition.

Similarly, we utilized perceptual loss for attribute prediction as well to measure the difference between the facial attributes of the synthesized and the real image. We applied the perceptual loss for attributes for both the LR and the HR network. To extract the attributes from a given HR image, we fine-tune the pre-trained VGG-Face [87] on 12 annotated facial attributes, which are shown in Table 6.2. After this, we utilize this attribute predictor to measure the attribute perceptual loss for both the LR and HR networks and the respective losses are given below:

$$L_{pa_{LR}} = \left\| A(G_{LR}(z|x^i_{LR})) - A(y^i) \right\|_2^2, \tag{6.15}$$

$$L_{pa_{HR}} = \left\| A(G_{HR}(z|x^j_{HR})) - A(y^j) \right\|_2^2, \tag{6.16}$$

where A(.) is the fine-tuned VGG-Face attribute predictor network. The total attribute perceptual loss is the sum of the perceptual attribute loss for the LR network ($L_{pa_{LR}}$) and the HR network ($L_{pa_{HR}}$):

$$L_{pa} = L_{pa_{LR}} + L_{pa_{HR}}. \tag{6.17}$$

6.3.5 L_2 Reconstruction Loss

L_2 reconstruction loss measures the reconstruction error in terms of Euclidean distance between the synthesized image and the corresponding real image and is defined for the LR and the HR network as follows:

$$L_{2_{LR}} = \left\| G_{LR}(z|x^i_{LR}) - y^i \right\|_2^2 \tag{6.18}$$

$$L_{2_{HR}} = \left\| G_{HR}(z|x^j_{HR}) - y^j \right\|_2^2. \tag{6.19}$$

The total L_2 reconstruction loss function is given by:

$$L_2 = L_{2_{LR}} + L_{2_{HR}}. \tag{6.20}$$

6.3.6 Overall Objective Function

The overall objective function for learning the network parameters in the proposed method is given as the sum of all the above defined loss functions:

$$L_{tot} = L_{cpl} + \lambda_1 L_a + \lambda_2 L_{GAN} + \lambda_3 L_{P_{LR}} + \lambda_4 L_{pa} + \lambda_5 L_2, \tag{6.21}$$

where L_{cpl} is the coupling loss function, L_a is the total attribute prediction loss function, L_{GAN} is the total generative adversarial loss function, $L_{P_{LR}}$ is the perceptual loss for the LR network, L_{pa} is the total perceptual attribute loss function and L_2 is the total reconstruction error. $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ are the adjustable hyper-parameters to weigh the different loss terms.

6.4 Experiments and Results

In this section, we demonstrate the effectiveness of the proposed approach by conducting various experiments on four datasets: Labeled Faces in the Wild-a (LFWA) [153], Celeb-

Faces Attributes Dataset (CelebA) [154], Surveillance Camera Face (SCFace) [155], and UnConstrained College Students (UCCS) Dataset [156]. We have compared our proposed method with six state-of-the-art methods on different datasets: VLRR [157], DCA [158], LR-FRW [159], D-align [160], SHSR [135] and SKD [131]. In addition, we conduct an ablation study to demonstrate the effectiveness of each loss function of our network.

6.4.1 Datasets

CelebA consists of 202,599 images with training, validation and test splits of approximately 162,000, 20,000 and 20,000 images, respectively. The total dataset corresponds to about 10,000 identities (20 images per identity) with no identity overlap. Images are annotated with 40 facial attributes such as, "wavy hair", "chubby", "bald", "male", etc. However, we only use 12 attributes (shown in Table 6.2) for our proposed method. We use the pre-cropped version of the dataset, where the face images aligned using the hand-labeled key points. The image size in regular HR resolution is equal to 178 × 212. We downsample the images to low-resolutions of 88 × 108, 68 × 84, 48 × 58.

LFWA has a total of 13,232 images of 5,749 identities with pre-defined train and test splits dividing the entire dataset into approximately two equal partitions. Each image is annotated with the same 40 attributes used in CelebA dataset. The images are normalized to 224 × 224 for HR image and downsampled to 96 × 96, 64 × 64, 32 × 32.

The SCface Dataset consists of 130 subjects, each having one HR frontal face image and multiple HR images, captured from three distances (4.2m, 2.6m and 1.0m, respectively) using different quality surveillance cameras. For fair comparison with previous methods [158], 50 subjects are randomly selected for training and the rest 80 subjects for testing. As in [158], we fix the HR image at 128 × 128 and downsample to 64 × 64, 32 × 32 and 16 × 16 for LR images.

The UCCS dataset is a very challenging dataset taken under unconstrained conditions. Following the experimental setting in [157], we perform evaluations on a 180-subject subset, where each subject has 25 or more images. We get a total of 5,220 images and and use 4,200 images for training, and the remaining 1,020 images for testing. For fair comparison, we normalize the cropped face regions to 80×80 as HR, and downsample them for LR images of 16×16. For the datasets SCFace and UCCS, which are not annotated with attributes, we use the state-of-the-art mixed-objective optimization network (MOON) [161] for generating the ground truth attributes.

6.4.2 Training Details

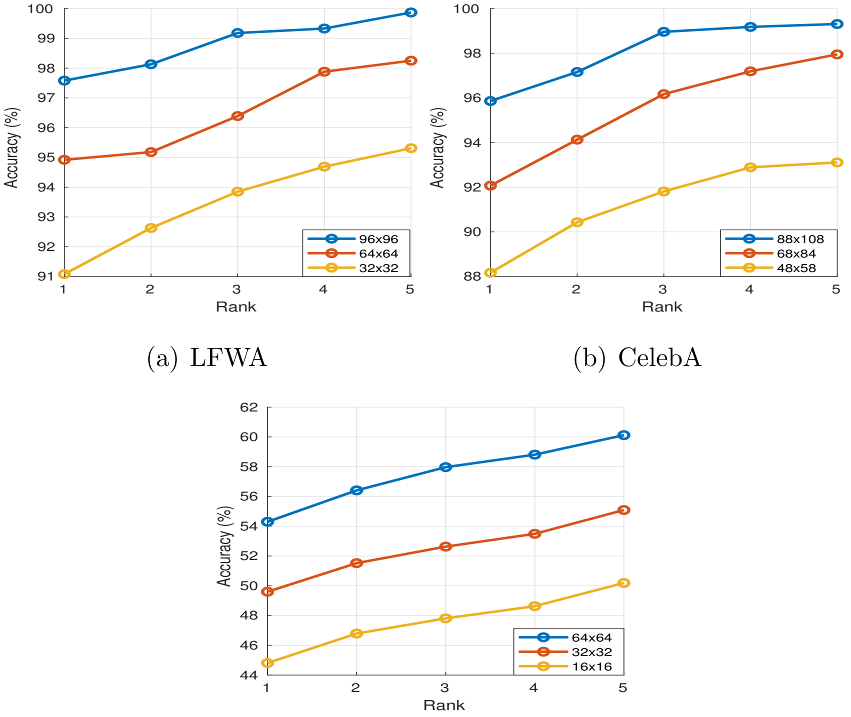
As mentioned, we have implemented the U-Net network as generators and patch based discriminators for the LR and HR networks. The entire architecture has been been implemented in Pytorch. For convergence, all the hyper-parameters are set to 1 except λ_3, and λ_4, which are set to 0.5. We have used a batch size of 6 for Adam optimizer [101] with first-order momentum of 0.5, and learning at a rate of 0.0004. We have used ReLU activation for the generator and Leaky ReLU with a slope of 0.25 for the discriminator. For fine-tuning the attribute predictor network VGG-Face for attribute perceptual loss, we have chosen 12 attributes (shown in Table 6.2 from the LFWA dataset).

The complex loss in (6.21) makes it difficult to train the whole network directly as the gradient diffusion caused by different tasks will lead to slow network convergence. To address this issue, we have employed a stage-wise learning strategy, where the information in the training data is presented to the network gradually. Specifically, we first optimize each task greedily by not updating the other task simultaneously. After the 'initialization' for each task, we fine-tune the whole network all together by optimizing all the tasks jointly.

For training, we require genuine and impostor pairs. The genuine/impostor pairs are constructed using LR and HR images of same/different subject. We balance the training set by using same number of genuine and impostor pairs.

6.4.3 Testing of the Proposed Method

The main objective of our proposed method is to match a test LR image against a gallery of HR images using the corresponding feature set from the common latent embedding subspace. During testing, a given probe LR image x^p_{LR} is passed through the LR network, and $z_1(x^p_{LR})$ is generated from the common embedding subspace. Similarly, the HR images



(a) LFWA

(b) CelebA

(c) SCFace

Figure 6.2: CMC curves for rank-n recognition accuracy for different low-resolution images for different datasets.

from the gallery are passed through the HR network and $z_2(x^j_{HR})$ is measured for each image x^j_{HR}. Eventually, the face recognition is performed by calculating the minimum Euclidean distance between $z_1(x^p_{LR})$ and $z_2(x^j_{HR})$ for all the gallery HR images:

$$\hat{j} = \arg\min_{j} \left\| z_1(x^p_{LR}) - z_2(x^j_{HR}) \right\|_2^2 . \tag{6.22}$$

Therefore, $x^{\hat{j}}_{HR}$ is the matching HR image from the gallery for the given probe LR image x^p_{LR}. The ratio of the number of correctly classified probes to the total number of probes is computed as the identification rate.

Additionally, the LR network can also be used for facial attribute prediction of a given LR probe image by passing the feature set $z_1(x^p_{LR})$ through the attribute predictor of the LR network. The predicted facial attribute can be used to narrow down the search for identification in a large gallery of HR images.

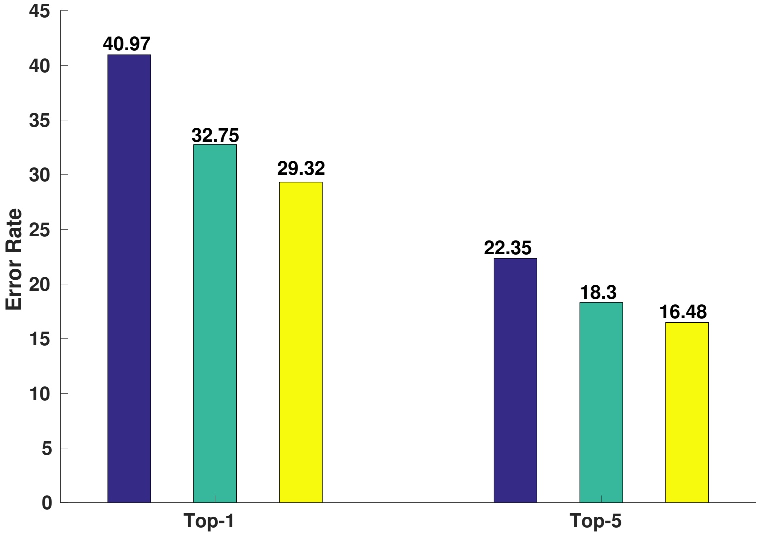


Figure 6.3: Top-1 and Top-5 Error rate comparison for VLLR (blue), SKD (Green), and our method (Yellow) using the UCCS dataset.

Table 6.1: Rank-1 recognition accuracy (%) on SCFace.

Model	Dist-1	Dist-2	Dist-3	Average
SHSR	14.70	15.70	19.10	16.50
DCA	12.19	18.44	25.53	18.72
LRFRW	20.40	20.80	31.71	18.72
D-Align	34.37	39.38	49.37	24.30
SKD	43.50	48.00	53.50	48.33
Our Method	44.81 ± 0.36	49.60 ± 0.41	54.30 ± 0.23	49.57 ± 0.39

6.4.4 Performance Evaluation

We have evaluated the proposed method and compared with other state-of-the-art methods on four different datasets using different low-resolution images. Fig. 6.2 provides the recognition accuracy of our proposed method from rank-1 to rank-5 for different resolution images using the LFWA, CelebA and SCFace dataset. We can clearly see that the proposed method gives very good performance for the LFWA and CelebA dataset. However, the SCFace has more challenging face variations than the LFWA and CelebA and the SCFace images have been taken in a typical commercial surveillance environment, which leads to lower recognition performance for the SCFace dataset when compared to LFWA and CelebA as seen in Fig. 6.2.

Table 6.1 tabulates the comparison of Rank-1 recognition rates with different state-of-

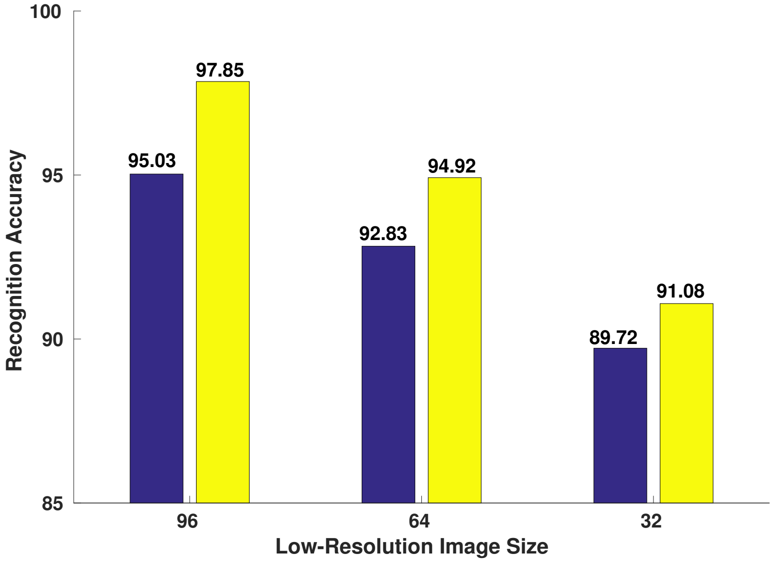


Figure 6.4: Recognition accuracy (%) comparison for SKD (Blue), and our method (Yellow) using the LFWA dataset for different size of LR image.

Table 6.2: Attribute prediction accuracy (%) comparison using CelebA dataset.

	Double Chin	Chubby	Eye Glasses	Male	Pale Skin	Moustache	Mouth Slightly Open	Young	Smiling	Goatee	Bald	Blond Hair
HR Input (Net A)	90.8	90.1	96.5	97.5	89.2	92.8	90.1	85.3	91.3	93.8	97.4	92.7
LR Input (Net A)	51.3	50.1	56.5	55.3	46.4	53.1	51.6	49.2	55.4	56.9	62.8	59.7
LR Input (fine-tuned Net A)	71.6	69.3	74.6	72.9	68.6	71.4	70.3	69.5	75.8	76.0	88.5	81.4
LR Input (Proposed Method)	83.2	82.6	89.3	91	83.3	88.1	86.6	78.9	87.2	88.9	93.6	88.4

the-art methods for the SCFace dataset using different distances from the camera, which corresponds to different resolution images. We can observe that our proposed method outperforms the state-of-the-art embedded method of cross-resolution face recognition model SKD [131] by 1.31%,1.60%,0.80%, and 1.22% on Dist-1, Dist-2, Dist-3, and average, respectively. We can also observe that our model even outperforms the state-of-the-art hallucination model SHSR [135] by approximately 33% an average for all the three distances. We have also compared proposed method with VLRR [157], and SKD [131] using the UCCS dataset. Top-1 and Top-5 error rate comparison has been shown in Fig. 6.3. UCCS is also a very challenging dataset, where the faces have been captured in completely unconstrained conditions. Due to this reason, the error rates for this dataset are very high. However, our proposed method performs better than the other two compared method by giving a lower error rate of at least 3.4% and 1.8% for Top-1 and Top-5 recognition. We can also notice from Fig. 6.4, that our proposed method outperforms SKD even for the LFWA dataset for

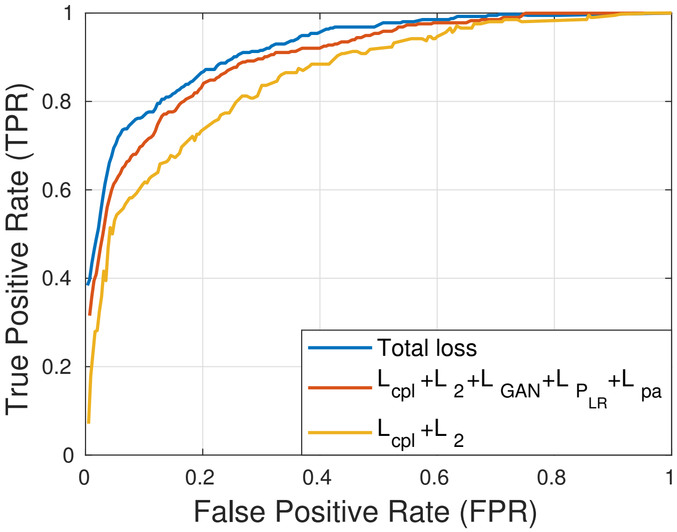


Figure 6.5: ROC curves corresponding to the ablation study.

different resolutions.

From performance evaluation, we observe that our proposed coupled framework with the contrastive loss function and leveraging facial attributes to transform different domains (LR and HR) into a common discriminative embedding subspace is superior than the other embedding techniques such as SKD and D-Align. It also shows the efficacy of exploiting multiple loss functions for cross-resolution face recognition. The relative importance of the loss functions has been covered in detail in ablation study (Sec. 6.4.6).

6.4.5 Attribute Prediction for Low-Resolution

One of the advantages of the proposed method is that it can be used for attribute prediction for LR face images. To illustrate the efficacy of our proposed approach for attribute prediction of LR face images, we have performed attribute prediction for 4 different scenarios: 1) Attribute prediction for HR images with the VGG-Face based attribute predictor, which is represented as Net A in Sec. 6.3.4. 2) Attribute prediction for LR images using attribute predictor Net A. 3) In this scenario, we first fine-tune the attribute predictor A with annotated LR images and then use it for attribute prediction of LR test images. This will be called “fine-tuned Net A” 4) In this final case, we test our attribute predictor from LR network for attribute prediction of the LR test images (see Fig. 6.1). We have performed this experiment for the Celeb-A dataset using 68 × 88 as LR images.

The attribute prediction results for the above 4 scenarios for 12 attributes using the Celeb-A have been tabulated in Table 6.2. We can notice from Table 6.2 that our approach shows the best performance in predicting the attributes of LR images for both datasets. Fine-tuning the Net A with LR images helps in improving its performance, however it does not perform as well as our method. Additionally, the performance of our LR network attribute predictor is comparable to the Net A performance for HR images.

6.4.6 Ablation Study

The objective function defined in (6.21) contains multiple loss functions: coupling loss (L_{cpl}), attribute prediction loss (L_a), perceptual loss ($L_{P_{LR}}$, L_{pa}), L_2 reconstruction loss (L_2), and GAN loss (L_{GAN}). In this section, we study the relative importance of different loss functions and the benefit of using them in our proposed method. For this experiment, we use different variations of our proposed approach and perform the evaluation using the LFWA dataset (64 × 64 LR images). The variations are: 1) cross-resolution face verification using the coupled framework with only coupling loss and L_2 reconstruction loss ($L_{cpl}+L_2$); 2) cross-resolution face verification using the coupled framework with coupling loss, L_2 reconstruction loss, GAN loss and perceptual loss ($L_{cpl} + L_2 + L_{GAN} + L_{P_{LR}} + L_{pa}$); 3) cross-resolution face verification using our framework with all the loss functions ($L_{cpl}+L_2+L_{GAN}+L_{P_{LR}}+L_{pa}+L_a$).

We use the above three variations of our framework and plot the ROC curve for the task of cross-resolution face verification using the features from the common embedding subspace. We can see from Fig. 6.5 that the generative adversarial loss and the perceptual loss (red curve) help in improving the cross-resolution verification performance, and adding the attribute prediction loss (blue curve) helps in more performance improvement. The reason for this improvement is that using facial attribute loss along with the contrastive loss leads to a more discriminative embedding subspace leading to a better face recognition performance. This also shows that multitask learning of attribute prediction and face recognition is useful and helps in cross-resolution face recognition task.

6.5 Summary

We have proposed a novel framework, which adopts a coupled GAN and exploits facial attributes for cross-resolution face recognition. The coupled GAN includes two sub-networks which project the low and high-resolution images into a common embedding subspace, where the goal of each sub-network is to maximize the pair-wise correlation between low and high-resolution images during the projection process. Moreover, we leverage facial attributes to further maximize the pair-wise correlation by implicitly matching facial attributes of the low and high-resolution images during the training. We comprehensively evaluated our model on four standard datasets and the results indicate that our model significantly outperforms other state-of-the-art models for cross resolution face recognition. Additionally, the enhancement obtained by different losses in the proposed method has been considered in an ablation study.

Chapter 7

Profile to Frontal Face Recognition in the Wild Using Coupled Conditional GAN

With the advent of deep learning, face recognition has achieved exceptional success. However, many of these deep face recognition models perform much better in handling frontal faces compared to profile faces. The major reason for poor performance in handling of profile faces is that it is inherently difficult to learn pose-invariant deep representations that are useful for profile face recognition. In this chapter, we hypothesize that the profile face domain possesses a latent connection with the frontal face domain in a latent feature subspace. We look to exploit this latent connection by projecting the profile faces and frontal faces into a common latent subspace and perform verification or retrieval in the latent domain. We leverage a cpGAN structure similar to Chapter 6 to find the hidden relationship between the profile and frontal images in a latent common embedding subspace. Specifically, the cpGAN framework consists of two conditional GAN-based sub-networks, one dedicated to the frontal domain and the other dedicated to the profile domain. Each sub-network tends to find a projection that maximizes the pair-wise correlation between the two feature domains in a common embedding feature subspace. The result is a system that is capable of effectively matching profile probe images against a gallery of frontal images

7.1 Introduction

In recent past, deep-learning based face recognition models have achieved exceptional success [162], however, many of these models perform relatively poorly in handling profile faces compared to frontal faces [163, 164]. When faces are captured in an unconstrained environment, in the wild, they are often in a profile orientation. Thus there is an equivalency between the challenging problems of unconstrained face recognition and profile face recognition. Pose, expression, and lighting variations are considered to be major obstacles in attaining high unconstrained face recognition performance. Existing methods focus on the pose variation problem by training separate models for learning pose-invariant features [162, 165], elaborate dense 3D facial landmark detection and warping [166], and synthesizing a frontal, neutral expression face from a single image [167–171].

Pose-Invariant Feature Representation: Face frontalization may be considered as an image-level pose invariant representation. However, feature-level pose invariant representations have also been a mainstay for face recognition. Canonical Correlation Analysis (CCA) was used in earlier works to analyze the commonality among pose-variant samples. Recently, with the advent of deep-learning, deep-learning-based methods have become popular for pose invariant feature representation. Cao *et al.* [162] exploit the inherent mapping between profile and frontal faces and transform a deep profile face representation to a canonical pose by adaptively adding residuals. Additionally, deep-learning methods consider several aspects, such as multiview perception layers [172], to learn a model separating identity from viewpoints. In [172], given a single 2D face image, a deep neural net, named Multi-View Perceptron (MVP) can untangle the identity and view features, and infer a full spectrum of multi-view images. MVP can also predict images under viewpoints that are unobserved in the training data. To allow a single network structure for multiple pose inputs, feature pooling across different poses is proposed in [173]. There have also been methods related to pose-invariant feature disentanglement [174] or identity preservation [175, 176] that aim to factorize out the non-identity part with a meticulously designed network. In [176], a new learning-based face representation, the Face Identity-Preserving (FIP) features, has been proposed. The FIP features are learned by using a deep neural network that combines the feature extraction layers and the reconstruction layer. The former layer generates FIP features from a face image, while the latter layer transforms the FIP features into an image in the canonical view.

Face Frontalization: Using a single image with large pose variation, it is very challenging to synthesize face with a frontal view with a neutral expression face due to two major reasons: a) recovering the 3D information from 2D projections is obscure and uncertain and b) presence of self-occlusion. Seminal works date back to the 3D Morphable Model (3DMM) [177], which models both the shape and appearance as PCA spaces. Hassner *et al.* [178] adopt a 3D shape model combined with input images to register and produce the frontalized face. Based on 3DMM, Zhu *et al.* [179] provide a high-fidelity pose and expression normalization method. However, 3D-based methods often do not provide reasonable results and suffer from a significant performance drop with large pose variations due to artifacts and severe texture losses. Some deep-learning-based methods have shown promising performance in terms of face frontalization [167,169–171,180–182]. In [181], a recurrent transform unit is proposed to incrementally rotate faces in fixed yaw angles and synthesize discrete 3D views. FF-GAN [169] solves the problem of large-pose face frontalization in the wild by incorporating a 3D face model into a GAN. Considering photo realistic and identity-preserving frontal view synthesis, a domain adaptation strategy for pose invariant face recognition is discussed in [182]. Tran *et al.* [167] propose a GAN framework to rotate a face and disentangle the identity representation by using a given pose code. In [171], a face normalization model (FNM) uses a GAN network with three distinct losses for generating canonical-view and expression-free frontal images. However, there are three major difficulties related to face frontalization or normalization in unconstrained environments:

- Complicated face-variations besides pose: In comparison to a controlled environment, there are more complex face variations, e.g., lighting, head pose, expression, in real-world scenarios. It is a difficult task to directly warp the input face to a normalized view [171].

- Unpaired data: Undoubtedly, obtaining a strictly normalized face is expensive and time-consuming, but getting an effective pair of images consisting of a target normalized face (i.e., frontal-facing, neutral expression) and an input face is difficult due to highly imbalanced datasets [171].

- Presence of artifacts: Synthesized 'frontal' faces contain artifacts caused by occlusions and non-rigid expressions.

In this chapter, we hypothesize that the profile face domain shares a latent connection with the frontal face domain in a latent deep feature subspace. We aim to exploit this connection by projecting the profile faces and frontal faces into a common latent subspace and perform verification or retrieval in this latent domain. We propose an embedding model for profile to frontal face verification based on a deep coupled learning framework which uses a GAN to find the hidden relationship between the profile face features and frontal face features in a latent common embedding subspace.

Our work is conceptually related to the embedding category of super-resolution [159, 183–185] in that our approach also performs verification of profile and frontal faces in the latent space but not in the original image space. From our experiments, we observe that transforming profile and frontal face features into a latent embedding subspace could yield higher performance than image-level face frontalization, which is susceptible to the negative influence of artifacts as a result of image synthesis. To our best knowledge, this study is the first attempt to perform profile-to-frontal face verification in a latent embedding subspace using generative modeling.

This chapter makes the following contributions:

- The chapter develops of a novel profile to frontal face recognition model using a cpGAN framework with multiple loss functions.

- The chapter includes comprehensive experiments using different datasets and a comparison of the proposed method with the state-of-the-art methods, indicating the efficacy of the proposed GAN framework.

- The proposed framework can potentially be used to improve the performance of traditional face recognition methods by integrating it as a preprocessing procedure for a face-frontalization schema.

- The chapter includes experiments to evaluate the frontalization performance of the cpGAN by using a face matcher (verifier) to compare off-pose faces with a gallery of frontal faces and also compare the frontalized images with the gallery to see if frontalizing the face would increase the face matcher performance.

- The chapter implements a coupled CNN (cpCNN) and includes experiments to evaluate the benefits of using the GAN by comparing the performance of a cpCNN with our proposed approach (cpGAN).

- The chapter implements an ADDA framework for profile to frontal face recognition and includes experiments to compare the performance of our proposed cpGAN with an ADDA network.

- The chapter includes generated qualitative results for the VGGFace2 dataset to test the robustness and reconstruction ability of our proposed coupled GAN framework.

7.2 Proposed Method

Here, we describe our method for profile to frontal face recognition. In contrast to the face normalization methods, we do not perform pose normalization (i.e., frontalization) on each profile image before matching. Instead, we seek to project the profile and frontal face images to a common latent low-dimensional embedding subspace using generative modeling. Inspired by the success of GANs [145], we explore adversarial networks to project profile and frontal images to a common subspace for recognition.

The framework of proposed profile to frontal coupled generative adversarial network (PF-cpGAN; shown in Fig. 7.1) consists of two modules, where each module contains a GAN architecture comprised of a generator and a discriminator. The generators that we have used in both modules are U-net auto-encoders that are coupled together using a contrastive loss function. In addition to adversarial and contrastive loss, we propose to guide the generators using the perceptual loss [149] based on the VGG 16 architecture, as well as an L_2 reconstruction error. The perceptual loss helps to generate a sharp and realistic reconstruction of the images.

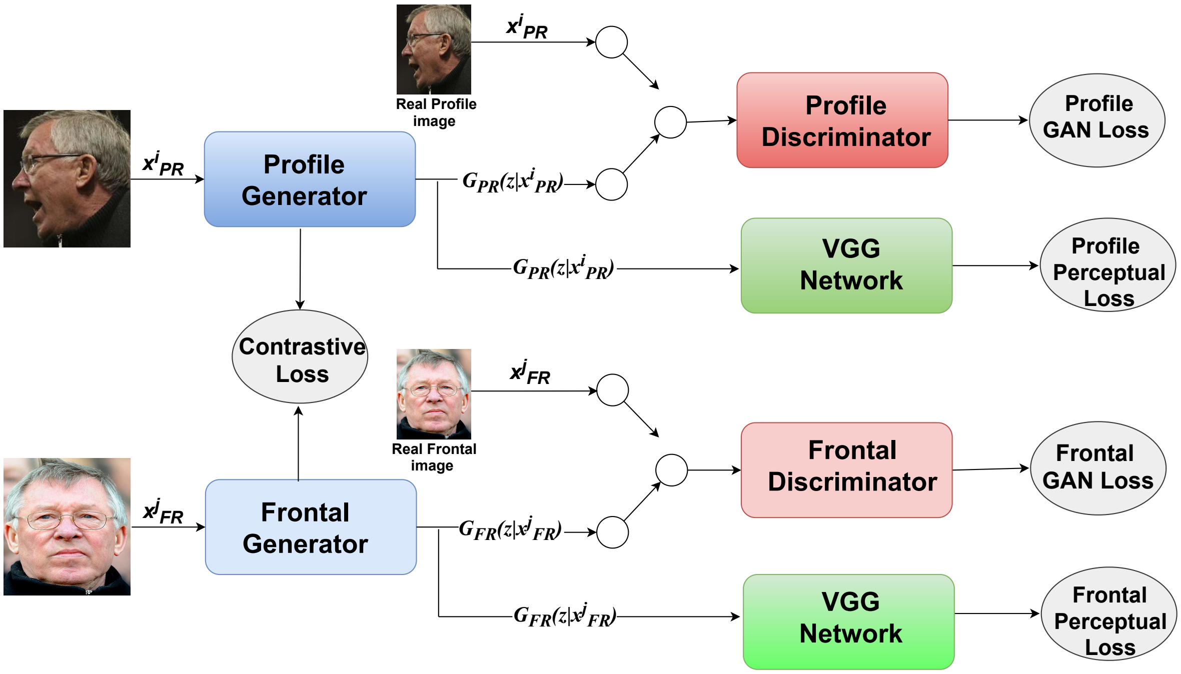


Figure 7.1: Block diagram of PF-cpGAN.

7.2.1 Profile to Frontal Coupled GAN

The main objective of PF-cpGAN is the recognition of profile face images with respect to a gallery of frontal face images, which have not been seen during the training. The matching of the profile and the frontal face images is performed in a common embedding subspace. PF-cpGAN consists of two modules: a profile GAN module and a frontal GAN module, both consisting of a GAN (generator + discriminator), and a perceptual network based on VGG-16.

For the generators, we use a U-Net [150] auto-encoder architecture (shown in Fig. 7.2(a)). The primary reason for using U-Net is that the encoder-decoder structure tends to extract global features and generate images by leveraging this overall information, which is very useful for global shape transformation tasks such as profile to frontal image conversion. Moreover, for many image translation problems, there is a significant amount of low-level information that needs to be shared between the input and output, and it is desirable to pass this information directly across all the layers including the bottleneck. Therefore, the use of skip-connections, as in U-net, provides a means for the encoder-decoder structure to circumvent the bottleneck and pass the information over to other layers.

For discriminators, we have used patch-based discriminators [148](shown in Fig. 7.2(b)), which are trained iteratively along with the respective generators. L_1 loss performs very well when trying to preserve the low-frequency details but fails to preserve the high-frequency details. However, using a patch-based discriminator that penalizes structure at the scale of the patches ensures the preservation of high-frequency details, which are usually eliminated when only L_1 loss is used.

The final objective of PF-cpGAN is to find the hidden relationship between the profile face features and frontal face features in a latent common embedding subspace. To find this common subspace between the two domains, we couple the two generators via a contrastive loss function, L_{cont}.

This loss function (L_{cont}) is a distance-based loss function, which tries to ensure that semantically similar examples (genuine pairs, i.e., a profile image of a subject with its corresponding frontal image) are embedded closely in the common embedding subspace, and, simultaneously, semantic dissimilar examples (impostor pairs, i.e., a profile image of a subject and a frontal image of a different subject) are pushed away from each other in the common embedding subspace. The contrastive loss function is defined as:

$$L_{cont}(z_1(x^i_{PR}), z_2(x^j_{FR}), Y) = (1-Y)\frac{1}{2}(D_z)^2 + (Y)\frac{1}{2}(\max(0, m - D_z))^2, \tag{7.1}$$

where x^i_{PR} and x^j_{FR} denote the *i-th* profile and *j-th* frontal face image, respectively. The variable Y is a binary label, which is equal to 0 if x^i_{PR} and x^j_{FR} belong to the same class (i.e., genuine pair), and equal to 1 if x^i_{PR} and x^j_{FR} belong to a different class (i.e., impostor pair). $z_1(.)$ and $z_2(.)$ denote only the encoding functions of the U-Net auto-encoder to transform x^i_{PR} and x^j_{FR}, respectively into a common latent embedding subspace. The value m is the contrastive margin and is used to "tighten" the constraint. D_z denotes the Euclidean distance between the outputs of the functions $z_1(x^i_{PR})$ and $z_2(x^j_{FR})$ given by:

$$D_z = \left\| z_1(x^i_{PR}) - z_2(x^j_{FR}) \right\|_2 . \tag{7.2}$$

Therefore, if $Y = 0$ (i.e., genuine pair), then the contrastive loss function (L_{cont}) is given as:

$$L_{cont}(z_1(x^i_{PR}), z_2(x^j_{FR}), Y) = \frac{1}{2}\left\|z_1(x^i_{PR}) - z_2(x^j_{FR})\right\|^2_2, \tag{7.3}$$

and if $Y = 1$ (i.e., impostor pair), then contrastive loss function (L_{cont}) is:

$$L_{cont}(z_1(x^i_{PR}),z_2(x^j_{FR}), Y) = \frac{1}{2}\max\left(0, m - \left\|z_1(x^i_{PR}) - z_2(x^j_{FR})\right\|_2\right)^2. \tag{7.4}$$

Thus, the total loss for coupling the profile generator and the frontal generator is denoted by L_{cpl} and is given as:

$$L_{cpl} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} L_{cont}(z_1(x^i_{PR}), z_2(x^j_{FR}), Y), \tag{7.5}$$

where N is the number of training samples. The contrastive loss in the above equation can also be replaced by some other distance-based metric, such as the Euclidean distance. However, the main aim of using the contrastive loss is to be able to use the class labels implicitly and find the discriminative embedding subspace, which may not be the case with some other metric such as the Euclidean distance. This discriminative embedding subspace would be useful for matching of a profile image against a frontal image.

7.2.2 Generative Adversarial Loss

Let the profile and frontal generators that reconstruct the corresponding profile and frontal image from the input profile and frontal image, be denoted as G_{PR} and G_{FR}, respectively. The patch-based discriminators used for the profile and frontal GANs are denoted as D_{PR} and D_{FR}, respectively. For the proposed method, we have used the conditional GAN, where the generator networks G_{PR} and G_{FR} are conditioned on input profile and frontal face images, respectively. We have used the conditional GAN loss function [146] to train the generators and the corresponding discriminators in order to ensure that the discriminators cannot distinguish the images reconstructed by the generators from the corresponding ground truth images. Let L_{PR} and L_{FR} denote the conditional GAN loss functions for the profile and the frontal GANs, respectively, where L_{PR} and L_{FR} are given as:

$$L_{PR} = F_{cGAN}(D_{PR}, G_{PR}, y^i_{PR}, x^i_{PR}), \tag{7.6}$$

$$L_{FR} = F_{cGAN}(D_{FR}, G_{FR}, y^j_{FR}, x^j_{FR}), \tag{7.7}$$

where function F_{cGAN} is the conditional GAN objective defined in (6.4). The term x^i_{PR} denotes the profile image used as a condition for the profile GAN, and y^i_{PR} denotes the real profile image. Note that the real profile image y^i_{PR} and the network condition given by x^i_{PR} are the same. Similarly, x^j_{FR} denotes the frontal image used as a condition for the frontal GAN and y^j_{FR} denotes the real frontal image. Again, the real frontal image y^j_{FR} and the network condition given by x^j_{FR} are the same. The total loss for the coupled conditional GAN is given by:

$$L_{GAN} = L_{PR} + L_{FR}. \tag{7.8}$$

7.2.3 L_2 Reconstruction Loss

We also consider the L_2 reconstruction loss for both the profile GAN and frontal GAN. The L_2 reconstruction loss measures the reconstruction error in terms of the Euclidean distance between the reconstructed image and the corresponding real image. Let $L_{2_{PR}}$ denote the reconstruction loss for the profile GAN and be defined as:

$$L_{2_{PR}} = \left\| G_{PR}(z|x^i_{PR}) - y^i_{PR} \right\|^2_2, \tag{7.9}$$

where y^i_{PR} is the ground truth profile image $G_{PR}(z|x^i_{PR})$ is the output of the profile generator.

Similarly, Let $L_{2_{FR}}$ denote the reconstruction loss for the frontal GAN:

$$L_{2_{FR}} = \left\| G_{FR}(z|x^j_{FR}) - y^j_{FR} \right\|^2_2, \tag{7.10}$$

where y^j_{FR} is the ground truth frontal image, $G_{FR}(z|x^j_{FR})$ is the output of the frontal generator.

The total L_2 reconstruction loss function is given by:

$$L_2 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (L_{2_{PR}} + L_{2_{FR}}). \tag{7.11}$$

7.2.4 Perceptual Loss

In addition to the GAN loss and the reconstruction loss that are used to guide the generators, we have also used the perceptual loss, which was introduced in [149] for style transfer and super-resolution. The perceptual loss function is used to compare high level differences, like content and style discrepancies, between images. The perceptual loss function involves comparing two images based on high-level representations from a pre-trained CNN, such as VGG-16 [22]. The perceptual loss function is a good alternative to solely using L_1 or L_2 reconstruction error, as it gives better and sharper high-quality reconstruction images [149].

In our proposed approach, perceptual loss is added to both the profile and the frontal module using a pre-trained VGG-16 network [22] . We extract the high-level features (ReLU3-3 layer) of the VGG-16 for both the real input image and the reconstructed output of the U-Net generator. The L_1 distance between these features of real and reconstructed images is used to guide the generators G_{PR} and G_{FR}. The perceptual loss for the profile network is defined as:

$$L_{P_{PR}} = \frac{1}{C_p W_p H_p} \sum_{c=1}^{C_p} \sum_{w=1}^{W_p} \sum_{h=1}^{H_p} \left\| V(G_{PR}(z|x^i_{PR}))^{c,w,h} - V(y^i_{PR})^{c,w,h} \right\|, \tag{7.12}$$

where $V(.)$ denotes a particular layer of the VGG-16, and the layer dimensions are given by C_p, W_p, and H_p.

Likewise the perceptual loss for the frontal network is:

$$L_{P_{FR}} = \frac{1}{C_p W_p H_p} \sum_{c=1}^{C_p} \sum_{w=1}^{W_p} \sum_{h=1}^{H_p} \left\| V(G_{FR}(z|x^j_{FR}))^{c,w,h} - V(y^j_{FR})^{c,w,h} \right\|. \tag{7.13}$$

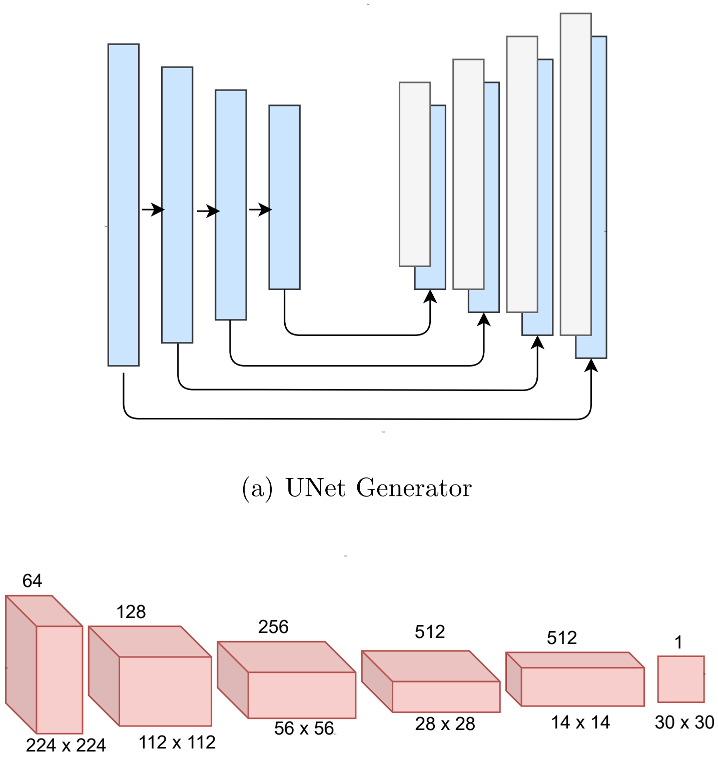
The total perceptual loss function is given by:

$$L_P = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (L_{P_{PR}} + L_{P_{FR}}). \tag{7.14}$$

7.2.5 Overall Objective Function

The overall objective function for learning the network parameters in the proposed method is given as the sum of all the loss functions defined above:

$$L_{tot} = L_{cpl} + \lambda_1 L_{GAN} + \lambda_2 L_P + \lambda_3 L_2, \tag{7.15}$$



(a) UNet Generator

(b) Patch-Based Discriminator

Figure 7.2: GAN Architectures

where L_{cpl} is the coupling loss given by (9), L_{GAN} is the total generative adversarial loss given by (12), L_P is the total perceptual loss given by (18), and L_2 is the total reconstruction error given by (15). Variables λ_1, λ_2, and λ_3 are the hyper-parameters to weigh the different loss terms.

7.3 Experiments

We initially describe our training setup and the datasets that we have used in our experiments. We show the efficiency of our method for the task of frontal to profile face verification by comparing its performance with state-of the-art face verification methods across pose-variation. We also explore the effect of face yaw in our algorithm. Additionally, we have implemented a cpCNN and an ADDA for profile to frontal face recognition. We have evaluated the performance of cpCNN and ADDA and compared it with the proposed

PF-cpGAN. We have also evaluated our PF-cpGAN for reconstruction of frontal images from input profile images. Finally, we conduct an ablation study to investigate the effect of each term in our total training loss function in (7.16).

7.3.1 Experimental Details

Datasets: The Celebrities in Frontal-Profile (CFP) dataset [163] is a mixture of constrained (i.e., carefully collected under different pose, illumination and expression conditions) and unconstrained (i.e., collected images from the Internet) settings. CFP includes 500 celebrities, averaging ten frontal and four profile face images per each celebrity. Following the standard 10-fold protocol [163], we divide the dataset into 10 folds, each of which consists of 350 same and 350 different pairs generated from 50 subjects (i.e., 7 same and 7 different pairs for each subject).

The CMU Multi-PIE database [75] contains 750,000 images of 337 subjects. Subjects were imaged from 15 viewing angles and 19 illumination conditions while exhibiting a range of facial expressions. It is the largest database for graded evaluation with respect to pose, illumination, and expression variations. There are four sessions in this database. For fair comparison, the database setting was made consistent with CAPG-GAN [186], where 250 subjects from Multi-PIE have been used. Consistent with CAPG-GAN, face images with neutral expression under 20 illuminations and 13 poses within ± 90°are used. We follow the setting-1 testing protocol provided in CAPG-GAN.

In setting-1, only images from session 1, which contains faces of 250 subjects were used. First 150 identities were used in the training set and remaining 100 identities were used for testing. The training set consists of all the images (13 poses and 20 illumination levels) of 150 identities, i.e., $150 \times 13 \times 20 = 39,000$ images in total. For testing, one gallery image with frontal view and normal illumination is used for each of the remaining 100 subjects. The numbers of the probe and gallery sets are 24,000 and 100 respectively

The IARPA Janus Benchmark A (IJB-A) [187] is a challenging dataset collected under complete unconstrained conditions covering full pose variation (yaw angles $-90°$to $+90°$). IJB-A contains 500 subjects with 5,712 images and 20,414 frames extracted from videos.

Following the standard protocol in [187], we evaluate our method on both verification and identification. The IARPA Janus Benchmark C (IJB-C) dataset [188] builds on IJB-A, and IJB-B [189] datasets and has a total of 31,334 images for a total number of 3,531 subjects. We have also evaluated our method on IJB-A and IJB-C datasets.

VGGFace2 is a large-scale face recognition dataset, where the images are downloaded from Google Image Search and have large variations in pose, age, illumination, and ethnicity. The dataset contains about 3.3 million images corresponding to more than 9000 identities with an average of 364 images per subject.

Implementation Details: We have implemented a U-Net autoencoder with a ResNet-18 [23] architecture pre-trained on ImageNet. We have added an additional fully-connected layer after the average pooling layer for the ResNet-18 for our U-Net encoder. The U-Net decoder has the same number of layers as the encoder. The entire framework has been implemented in Pytorch. For convergence, λ_1 is set to 1, and λ_2, and λ_3 are both set to 0.25. We used a batch size of 128 and an Adam optimizer [101] with first-order momentum of 0.5, and learning rate of 0.0004. We have used the ReLU activation function for the generator and Leaky ReLU with a slope of 0.3 for the discriminator.

For training, genuine and impostor pairs were required. The genuine/impostor pairs are created by frontal and profile images of the same/different subject. During the experiments, we ensure that the training set are balanced by using the same number of genuine and impostor pairs.

7.3.2 Evaluation on CFP with Frontal-Profile Setting

We first perform evaluation on the CFP dataset [163], a challenging dataset created to examine the problem of frontal to profile face verification in the wild. The same 10-fold protocol is applied on both the Frontal-Profile and Frontal-Frontal settings. For fair comparison and as given in [163], we consider different types of feature extraction techniques like HoG [190], LBP [191], and Fisher Vector [192] along with metric learning techniques like Sub-SML [193], and the Diagonal metric learning (DML) as reported in [192]. We also compare against deep-learning techniques, including Deep Features [194], and PR-REM [162].

Table 7.1: Performance comparison on CFP dataset. Mean Accuracy and equal error rate (EER) with standard deviation over 10 folds.

	Frontal-Profile		Frontal-Frontal	
Algorithm	Accuracy	EER	Accuracy	EER
HoG+Sub-SML [163]	77.31 ± 1.61	22.20 ± 1.18	88.34 ± 1.31	11.45 ± 1.35
LBP+Sub-SML [163]	70.02 ± 2.14	29.60 ± 2.11	83.54 ± 2.40	16.00 ± 1.74
FV+Sub-SML [163]	80.63 ± 2.12	19.28 ± 1.60	91.30 ± 0.85	8.85 ± 0.74
FV+DML [163]	58.47 ± 3.51	38.54 ± 1.59	91.18 ± 1.34	8.62 ± 1.19
Deep Features [194]	84.91 ± 1.82	14.97 ± 1.98	96.40 ± 0.69	3.48 ± 0.67
PR-REM [162]	93.25 ± 2.23	7.92 ± 0.98	98.10 ± 2.19	1.10 ± 0.22
PF-cpGAN	93.78 ± 2.46	7.21 ± 0.65	98.88 ± 1.56	0.93 ± 0.14

The results are summarized in Table 7.1.

We can observe from Table 7.1 that our proposed framework, PF-cpGAN, gives much better performance than the methods that use standard hand-crafted features of HoG, LBP, or FV, providing minimum of 13% improvement in accuracy with a 12% decrease in EER for the profile-frontal setting. PF-cpGAN also improves on the performance of the Deep Features by approximately 9% with a 7.5% decrease in EER for the profile-frontal setting. Finally, PF-cpGAN performs on-par with the best deep-learning method of PR-REM, and, in-fact, does slightly better than PR-REM by $\approx 0.5\%$ improvement in accuracy with a 0.7% decrease in EER for the profile-frontal setting. This performance improvement clearly shows that usage of a GAN framework for projecting the profile and frontal images in the latent embedding subspace and maintaining the semantic similarity in the latent space is better than some other deep-learning techniques such as Deep Features or PR-REM.

7.3.3 Evaluation on IJB-A and IJB-C

Here, we focus on unconstrained face recognition on the IJB-A dataset to quantify the superiority of our PF-cpGAN for profile to frontal face recognition. Some of the baselines for comparison on IJB-A are DR-GAN [167], FNM [171], PR-REM [162], and FF-GAN [169]. We have also compared them with other methods as listed in [171] and shown in Table 7.2. As shown in Table 7.2, we perform better than the state-of-the-art methods for both verification and identification. Specifically, for verification, we improve the GAR by at least

Table 7.2: Performance comparison on IJB-A benchmark. Results reported are the 'average±standard deviation' over the 10 folds specified in the IJB-A protocol. Symbol '-' indicates that the metric is not available for that protocol.

	Verification		Identification	
Method	GAR@ FAR= 0.01%	GAR@ FAR= 0.001%	@ Rank-1	@ Rank-5
OPENBR [195]	23.6 ± 0.9	10.4 ± 1.4	24.6 ± 1.1	37.5 ± 0.8
GOTS [195]	40.6 ± 1.4	19.8 ± 0.8	43.3 ± 2.1	59.5 ± 2.0
PAM [165]	73.3 ± 1.8	55.2 ± 3.2	77.1 ± 1.6	88.7 ± 0.9
DCNN [196]	78.7 ± 4.3	-	85.2 ± 1.8	93.7 ± 1.0
DR-GAN [167]	77.4 ± 2.7	53.9 ± 4.3	85.5 ± 1.5	94.7 ± 1.1
FF-GAN [197]	85.2 ± 1.0	66.3 ± 3.3	90.2 ± 0.6	95.4 ± 0.5
FNM [171]	93.4 ± 0.9	83.8 ± 2.6	96.0 ± 0.5	98.6 ± 0.3
PR-REM [162]	94.4 ± 0.9	86.8 ± 1.5	94.6 ± 1.1	96.8 ± 1.0
PF-cpGAN	95.8 ± 0.82	91.2 ± 1.3	97.6 ± 1.0	98.8 ± 0.4

1.4% compared to other methods. For instance, at the FAR of 0.01%, the best previously-used method is PR-REM, with an average GAR of 94.4%. PF-cpGAN improves upon PR-REM and gives an average GAR of 95.8% at the same FAR. We also show improvement in identification. Specifically, the rank-1 recognition rate shows an improvement of around 1.6% in comparison to the best state-of-the-art method, FNM [171].

Table 7.3: Performance comparison on IJB-C benchmark. Results reported are the 'average±standard deviation' over the 10 folds specified in the IJB-C protocol. Symbol '-' indicates that the metric is not available for that protocol.

	Verification		Identification	
Method	GAR@ FAR= 0.01%	GAR@ FAR= 0.001%	@ Rank-1	@ Rank-5
GOTS [188]	62.1 ± 1.1	36.3 ± 1.2	38.5 ± 1.6	53.8 ± 1.8
FaceNet [85]	82.3 ± 1.18	66.3 ± 1.3	70.4 ± 1.2	78.8 ± 2.3
VGG-CNN [87]	87.2 ± 1.09	74.3 ± 0.9	79.6 ± 1.04	87.8 ± 1.3
FNM [171]	91.2 ± 0.8	80.4 ± 1.8	84.6 ± 0.6	93.7 ± 0.9
PR-REM [162]	92.1 ± 0.8	83.4 ± 1.5	83.1 ± 0.4	92.6 ± 1.1
PF-cpGAN	93.8 ± 0.67	86.1 ± 0.7	88.3 ± 1.2	94.8 ± 0.6

We have also performed the task of verification and identification using the IJB-C dataset according to the verification and the identification protocol given in that dataset. The results are provided in Table 7.3, showing that the proposed PF-cpGAN improves on the existing state-of-the-art methods for both verification and identification. For instance, at the FAR of 0.01%, the best previously-used method is PR-REM, with an average GAR of 92.1%. PF-cpGAN improves upon PR-REM and gives an average GAR of 93.8% at the same FAR. We also observe that, for identification, specifically, rank-1 recognition, PF-cpGAN shows an improvement over the previous best state-of-the-art method FNM [171] by about 1.1%.

7.3.4 A Further Analysis on Influences of Face Yaw

In addition to complete profile to frontal face recognition, we also perform a more in-depth analysis on the influence of face yaw angle on the performance of face recognition to better understand the effectiveness of the PF-cpGAN for profile to frontal face recognition. We perform this experiment for the CMU Multi-PIE dataset [75] under setting-1 for fair comparison with other state-of-the-art methods. As shown in Table 7.4, we achieve comparable performance with other state-of-the-art methods for different yaw angles. Under extreme pose, PF-cpGAN achieves significant improvements (i.e., approx. 77% to 88% under ±90°).

Table 7.4: Rank-1 recognition rates (%) across poses and illuminations under Multi-PIE Setting-1.

Method	±90°	±75°	±60°	±45°	±30°	±15°
HPN [198]	29.82	47.57	61.24	72.77	78.26	84.23
c-CNN [199]	47.26	60.7	74.4	89.0	94.1	97.0
TP-GAN [200]	64.0	84.1	92.9	98.6	99.99	99.8
PIM [182]	75.0	91.2	97.7	98.3	99.4	99.8
CAPG-GAN [186]	77.1	87.4	93.7	98.3	99.4	99.99
FNM+VGG-Face [171]	41.1	67.3	83.6	93.6	97.2	99.0
FNM+Light CNN [171]	55.8	81.3	93.7	98.2	99.5	99.9
PF-cpGAN	88.1	94.2	97.6	98.9	99.9	99.9

7.3.5 Reconstruction of Frontal and Profile Images

As noted in Sec. 1, the PF-cpGAN framework can also be used for reconstruction of frontal images by using profile images as input and vice versa. The results of reconstructing frontal images using the profile images as input are given in Fig. 7.3, and the results of reconstructing profile images using the frontal images as input is given in Fig. 7.4. The

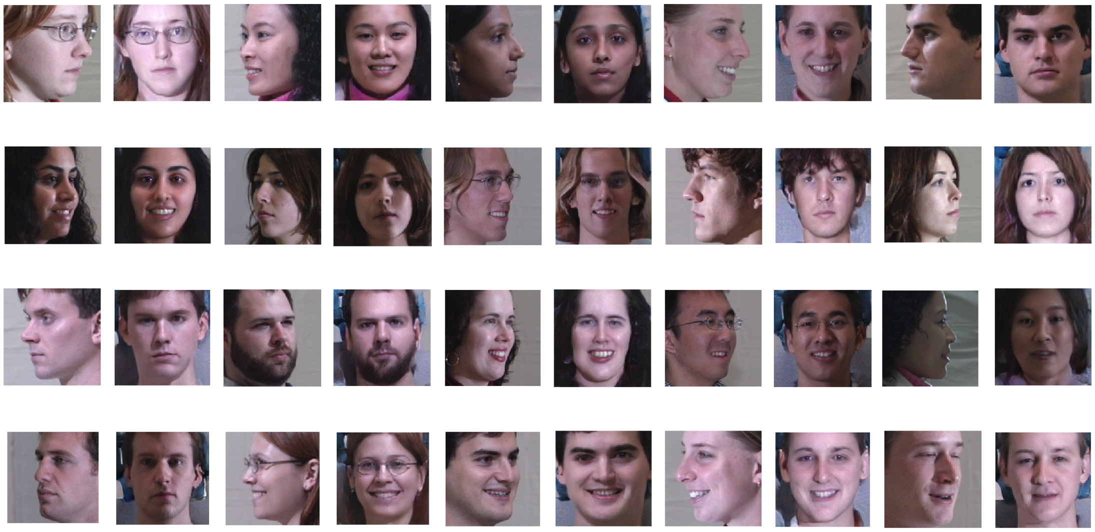


Figure 7.3: Reconstruction of frontal images at the output of the frontal U-Net generator with profile images as input to the profile U-Net generator. Every odd number column represent the input profile image and every even number column represents the output frontal image. The input images belong to the CMU-MultiPIE dataset.

reconstruction procedure for frontal images is given as follows: The profile image is given as input to the profile U-Net generator and the feature vector generated at the bottleneck of the profile generator (i.e., at the output of the encoder of the profile U-Net generator) is passed through the decoder section of the frontal U-Net generator to reconstruct the frontal image. Similarly the reconstruction procedure for profile images is given as follows: The frontal image is given as input to the frontal U-Net generator, and the feature vector generated at the bottleneck of the frontal generator (i.e., at the output of the encoder of the frontal U-Net generator) is passed through the decoder section of the profile U-Net generator to reconstruct the profile image. As we can see from Fig. 7.3 and Fig. 7.4, the PF-cpGAN can preserve the identity and generate high-fidelity faces from an unconstrained dataset such as CMU-MultiPIE. These results show the robustness and effectiveness of PF-cpGAN for multiple use of profile to frontal matching in the latent common embedding subspace, as well as in the reconstruction of facial images.

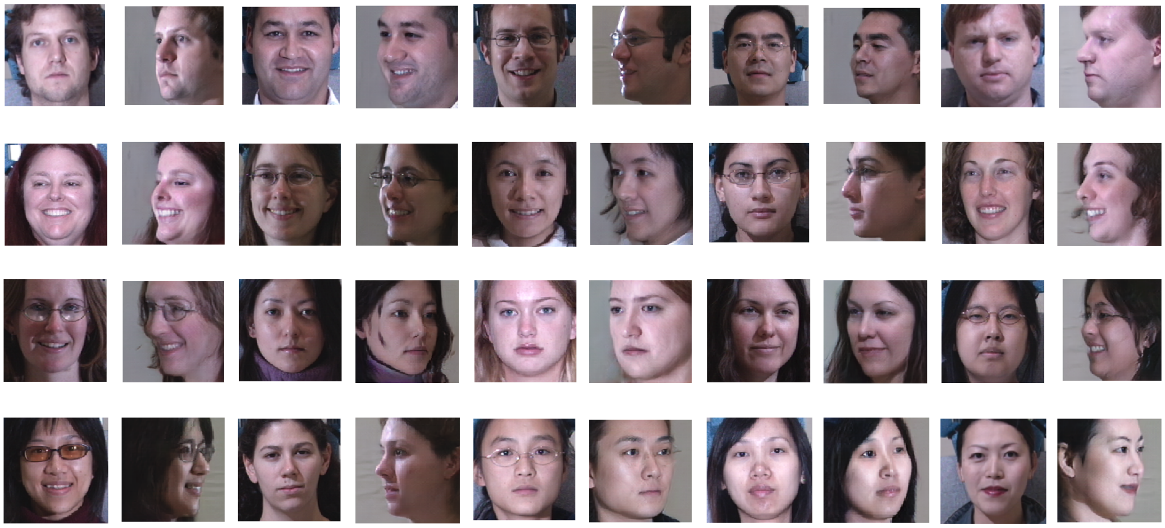
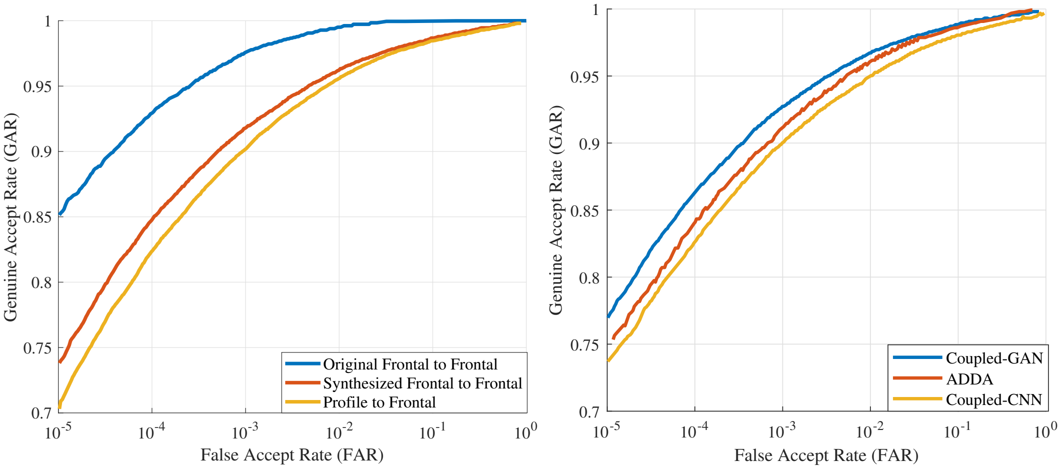


Figure 7.4: Reconstruction of profile images at the output of the profile U-Net generator with frontal images as input to the frontal U-Net generator. Every odd number column represents the input frontal image, and every even number column represents the output profile image. The input images belong to the CMU-MultiPIE dataset.

7.3.6 Evaluation of the Frontalization by CpGAN as a Pre-processing for Face Matching

As mentioned earlier, our coupled GAN framework can also be used for frontalization, which can be an important pre-processing step for other face-recognition tasks. Here, we conducted experiments to indicate the effectiveness of the frontalization performed using our cpGAN for the face verification task. In this set of experiments, we have used an Inception [201] based FaceNet [85] model for the face verification task, which is specifically the NN2 model from [85]. We have performed this set of experiments on the VGGFace2 dataset [202].

The VGGFace2 dataset provides annotation to enable evaluation of face matching across different poses [202]. In the dataset, six pose templates corresponding to three poses (i.e., two templates for a single pose) have been provided for about 300 identities. A template corresponds to five faces from the same subject with a consistent pose. This pose can be frontal, three-quarter or profile view. Consequently, for the 300 identities, there are a total of 1.8K templates with 9K images in total [202]. For this set of experiments, we have used only the profile and the frontal templates, which corresponds to about 6K images corresponding



(a) Frontalization Performance (b) Different network comparisons

Figure 7.5: Performance comparison for (a) Frontalization using cpGAN as a preprocessing. (b) PF-cpGAN (Coupled-GAN) vs cpCNN (Coupled-CNN) vs PF-ADDA (ADDA). FAR and GAR are given in %.

to 300 identities.

Here, we perform face verification using FaceNet in three different settings. In the first setting, we choose about 2.5K frontal images corresponding to 250 identities. Using these images, we fine-tune the Inception model NN2 from FaceNet for frontal to frontal face verification. Next, using this FaceNet model, we evaluate the frontal to frontal face verification on the remaining 50 identities. This setting will be called *Original Frontal to Frontal*. In the second setting, we choose about 5K images corresponding to 250 identities, which have both profile and frontal images. Using these images, we fine-tune the Inception model NN2 from FaceNet for profile to frontal face verification. Next, using this FaceNet model, we evaluate the profile to frontal face verification on the remaining 50 identities. This setting will be called *Profile to Frontal*. In the third setting, we used our cpGAN to frontalize the profile images from the dataset used in second setting (300 subjects with about 3K profile images) using the method outlined in Sec. 7.3.5. We call this frontalized dataset synthesized frontal dataset. Next, using the fine-tuned FaceNet model from the first setting, we evaluate the frontal to frontal face verification on 50 identities from the synthesized frontal dataset. Specifically, in the third setting we are trying to check how well the proposed cpGAN is able to frontalize the images by running the frontal to frontal face verification model on the synthesized frontal dataset. This setting will be called *Synthesized Frontal to Frontal*. Note that we try to keep the 50 identities used for evaluation consistent across all the three settings.

Using the ROC curve as our performance metric, we have compared the performance of these three settings to evaluate the effectiveness of frontalization performed using our proposed cpGAN. The performance curves are provided in Fig. 7.5(a). As expected the first setting (Original Frontal to Frontal) gives us the best performance and it is the upper bound as we are using the original frontal dataset for training and evaluation in this setting. On comparing the curves for the second (Profile to Frontal) and third settings (Synthesized Frontal to Frontal), it can be observed that the Synthesized Frontal to Frontal outperforms the Profile to Frontal face recognition model. This shows that the preprocessing in the form of frontalization performed using the proposed cpGAN framework improves the performance of a FaceNet model for profile to frontal face verification.

7.3.7 Implementation of Coupled CNN and Domain Adaptation Network for Profile to Frontal Face Matching

Before the advent of GAN, many deep-learning applications used CNNs for classification, regression, or reconstruction. To showcase the advantage of using a GAN model in our proposed approach for profile to frontal face recognition, we have also implemented two other frameworks that will be explained in this section. The performance comparison of the proposed PF-cpGAN with these new frameworks will be discussed in the following section.

Coupled CNN: In the literature, it has been shown that GAN is better than CNN for some deep-learning applications. To confirm this hypothesis for our proposed application, we have implemented a coupled CNN and compared its performance with our proposed coupled GAN architecture. The coupled CNN (cpCNN) architecture is shown in Fig. 7.6. For fair comparison, we have used ResNet18 [23] pre-trained on ImageNet network as our CNN architecture for both Frontal CNN and Profile CNN. Additionally, we have added an extra fully-connected layer after the average pooling layer of ResNet18 for our coupled CNNs.

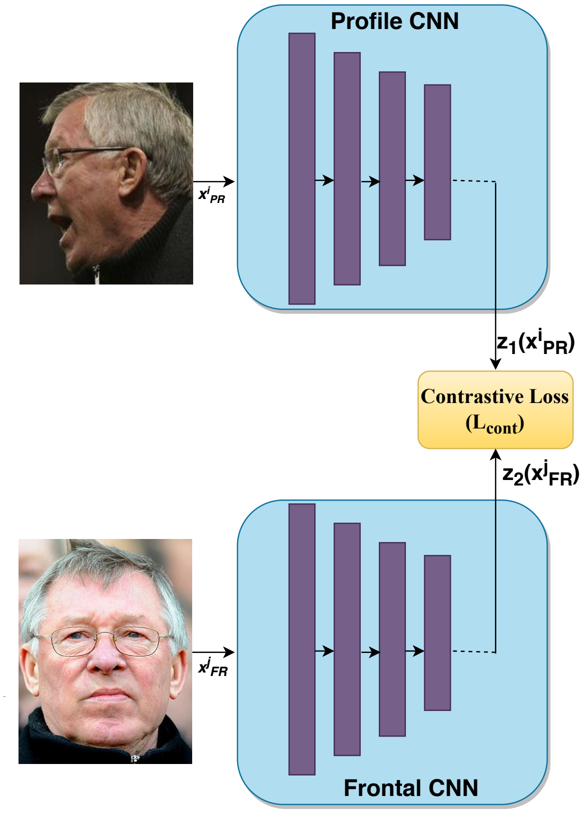


Figure 7.6: Block diagram of Coupled CNN.

The frontal and profile CNNs are coupled together at their output layer using a contrastive loss function (L_{cont}). This loss function (L_{cont}) is a distance-based loss function, which is similar to the contrastive loss function (7.1) that we have used for PF-cpGAN. For ease of understanding, we have used the same naming convention for cpCNN as in PF-cpGAN.

We have used the VGGFace2 dataset for training and testing of the cpGAN. As in Sec. 7.3.6, we choose about 5K images corresponding to 250 identities, which have both profile and frontal images for fine-tuning the cpCNN for profile to frontal face verification. We have tested the cpCNN on the 50 disjoint identities from VGGFace2. The performance comparison is discussed in the following section.

Domain Adaptation Network: A profile to frontal recognition network could very well be implemented using deep-learning based domain adaptation techniques. These domain adaptation techniques attempt to alleviate the negative effects of domain shift (frontal domain to profile domain in our case) by learning deep neural transformations that map

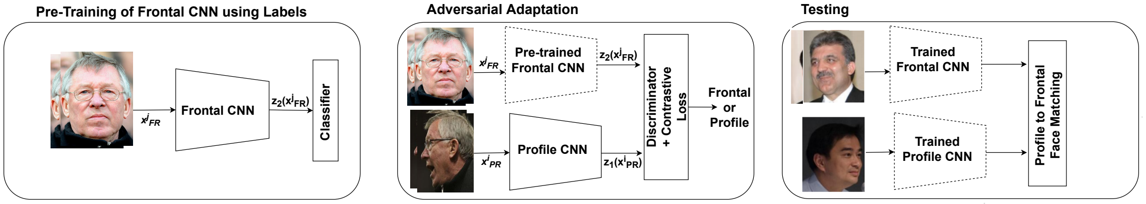


Figure 7.7: Block diagram of Profile to Frontal Adversarial Domain Adaptation (PF-ADDA).

both domains into a common feature space. Recently, adversarial adaptation methods, which are based on reconstructing the target domain from the source representation have become increasingly popular. These adversarial methods seek to reduce an approximate domain discrepancy distance through an adversarial objective function with respect to a domain discriminator [203].

Taking a cue from [203], we have implemented an unsupervised discriminative domain adaption network for profile to frontal face recognition. Hereafter, this network will be known as PF-ADDA. For this adversarial domain adaptation network, we consider the source domain as the frontal images and the target domain as the profile images. The architecture of PF-ADDA is shown in Fig. 7.7. PF-ADDA has been implemented and optimized in two steps, as described in the next two paragraphs: In the first step of pre-training a frontal CNN, a discriminative representation is learned using the labels in the frontal image domain (source domain). This implies we first pre-train a frontal image encoder CNN using labeled frontal image examples. The optimization for this step is given as:

$$\min_{z_2,C} L_{cls}(X_{FR}, Y_{FR}) = -E_{(x_{FR},y_{FR})\sim(X_{FR},Y_{FR})} \sum_{k=1}^{K} 1_{[k=y_{FR}]} \log C(z_2(x^j_{FR})), \tag{7.16}$$

where the classification loss L_{cls} is optimized over z_2, and frontal image classifier, C, by training using the labeled source data, X_{FR}, and Y_{FR}.

In the second step of adversarial adaptation, a separate encoding that maps the profile image data to the same space as the frontal image domain using an asymmetric mapping is learned through a combination of domain-adversarial loss and the contrastive loss. In other words, this implies that we perform adversarial adaptation by learning a profile image encoder CNN such that a discriminator that sees encoded frontal and profile images cannot accurately predict their domain label. In addition to the discriminator loss, the frontal and

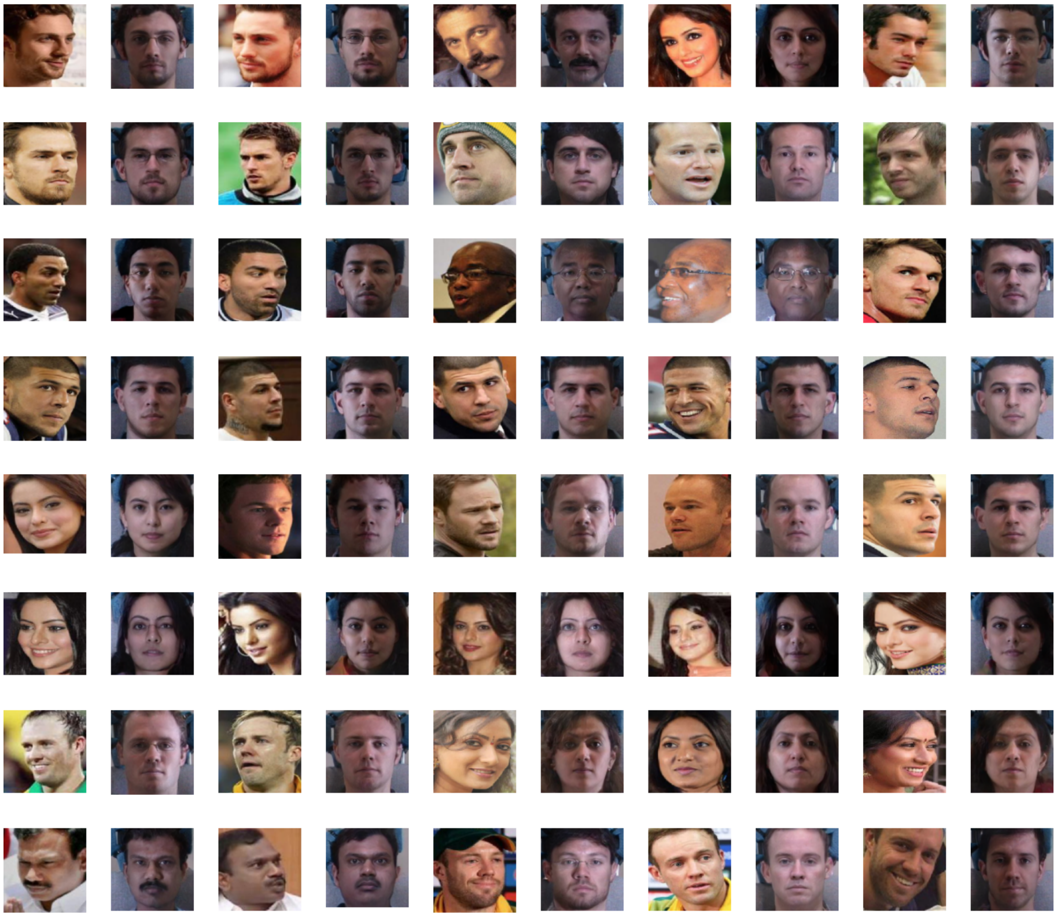


Figure 7.8: Reconstruction of frontal images at the output of the frontal U-Net generator with profile images as input to the profile U-Net generator. Every odd number column represents the input profile image, and every even number column represents the output frontal image. The input images belong to the VGGFace2 dataset.

profile domain CNNs are also coupled through a contrastive loss. The optimizations for this step are given as :

$$\min_{D} L_{adv_D}(X_{FR}, X_{PR}, z_2, z_1) = \\ - E_{x_{FR} \sim X_{FR}}[\log D(z_2(x^j_{FR}))] - E_{x_{PR} \sim X_{PR}}[\log(1 - D(z_1(x^i_{PR})))], \tag{7.17}$$

$$\min_{z_1} L_{adv_G}(X_{FR}, X_{PR}, D) = -E_{x_{PR} \sim X_{PR}}[\log D(z_1(x^i_{PR}))], \tag{7.18}$$

and

$$L_{cont}(z_1(x^i_{PR}), z_2(x^j_{FR}), Y) = (1 - Y)\frac{1}{2}(D_z)^2 + (Y)\frac{1}{2}(\max(0, m - D_z))^2. \tag{7.19}$$

As shown in Equations (7.17), and (7.18), frontal image encoder CNN (source CNN) is fixed during the second stage, we just need to optimize the discriminator loss L_{adv_D} and profile encoder loss L_{adv_G} over the profile encoder CNN to generate z_1 without revisiting the source domain encoder. Finally, along with the adversarial losses, we also optimize the contrastive loss, L_{cont}, between the output of the Frontal CNN and Profile CNN as shown in (7.19). This contrastive loss is similar to the loss used for cpCNN and PF-cpGAN.

During testing, profile images (target domain) are mapped with the profile image encoder to the shared feature space, and frontal images (source images) are mapped with the frontal image encoder to the shared feature space. Finally, the profile to frontal matching is performed in the shared feature space. Dashed lines in Fig. 7.7 indicate fixed network parameters.

We have used the VGGFace2 dataset for training and testing of the PF-ADDA. For fair comparison, the train and test split of the dataset for the PF-ADDA is consistent with the split for cpCNN.

7.3.8 Performance Comparison of PF-cpGAN vs cpCNN vs PF-ADDA.

We have performed several experiments to compare the performance of our proposed PF-cpGAN approach with cpCNN and PF-ADDA. These experiments are performed on the VGGFace2 dataset [202]. As already mentioned in the previous section 5.7, for implementing cpCNN and PF-ADDA, we have used a common network architecture as PF-cpGAN. Furthermore, we have been consistent in the training procedure (optimizer, batch-size, learning rate decay schedule, etc.). The performance comparison is plotted in terms of ROC and shown in Fig. 7.5(b). The ROC results curves show that the proposed PF-cpGAN method outperforms other methods and gives much better performance for face verification under different pose variations. This demonstrates the effectiveness of coupled-GAN compared to other implementations. The improvement in performance using PF-cpGAN could be attributed to individual discriminators in the PF-cpGAN, which generate more domain specific features. The improvement can also be attributed to the sharpening of the features due to the perceptual loss terms.

7.3.9 Coupled-GAN Qualitative Results on VGGFace2

In this section, we test the robustness of our proposed approach Pf-cpGAN on VGGFace2 dataset by reconstructing frontal images from input profile images. In VGGFace2 [202], two networks are trained to estimate the pose of images in the dataset. Specifically, a 5-way classification ResNet-50 is trained on the large-scale CASIA-WebFace dataset [71] to estimate head pose (roll, pitch, yaw). This model is then leveraged to predict pose of all the images in the dataset. As a result, VGGFace2 published different pose templates for 368 identities. Specifically, there are six templates for each subject: two templates each for frontal view, three-quarter view and profile view. There are five images per template. Here, we used about 250 identities to construct our pairs for training our coupled-GAN framework. Next, we test our network for frontalization of profile images. We follow the same procedure as discussed in Sec. 7.3.5 to frontalize our images. The results for frontalized images are shown in Fig. 7.8. From these images, it can be observed that PF-cpGAN can preserve the identity and generate high-fidelity faces from VGGFace2 dataset. These results demonstrate the robustness and effectiveness of our coupled-GAN framework for frontalizing pose-variant images in the latent common embedding subspace.

7.3.10 Ablation Study

The objective function defined in (7.16) contains multiple loss functions: coupling loss (L_{cpl}), perceptual loss (L_P), L_2 reconstruction loss (L_2), and GAN loss (L_{GAN}). It is important to understand the relative importance of different loss functions and the benefit of using them in our proposed method. For this experiment, we use different variations of PF-cpGAN and perform the evaluation using the IJB-A dataset. The variations are: 1) PF-cpGAN with only coupling loss and L_2 reconstruction loss ($L_{cpl} + L_2$); 2) PF-cpGAN with coupling loss, L_2 reconstruction loss, and GAN loss ($L_{cpl} + L_2 + L_{GAN}$); 3) PF-cpGAN with all the loss functions ($L_{cpl} + L_2 + L_{GAN} + L_P$).

We use these three variations of our framework and plot the ROC for profile to frontal

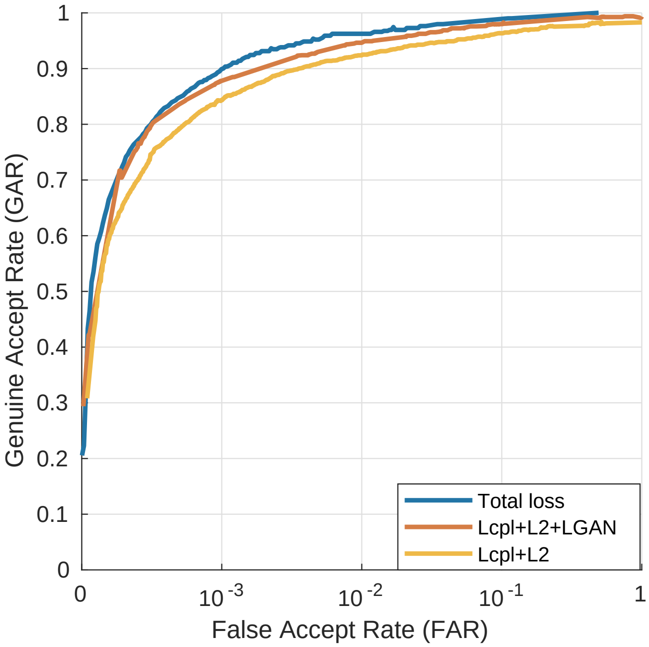


Figure 7.9: ROC curves showing the importance of different loss functions for ablation study.

face verification using the features from the common embedding subspace. We can see from Fig. 7.9 that the generative adversarial loss helps to improve the profile to frontal verification performance, and adding the perceptual loss (blue curve) results in an additional performance improvement. The reason for this improvement is that using perceptual loss along with the contrastive loss leads to a more discriminative embedding subspace resulting in better face recognition performance.

7.4 Summary

We proposed a new framework which uses a coupled GAN for profile to frontal face recognition. The coupled GAN contains two sub-networks which project the profile and frontal images into a common embedding subspace, where the goal of each sub-network is to maximize the pair-wise correlation between profile and frontal images during the process of projection. We thoroughly evaluated our model on several standard datasets and the results demonstrate that our model notably outperforms other state-of-the-art algorithms for profile to frontal face verification. For instance, under extreme pose of ±90°, PF-cpGAN achieves improvements of approx. 11% (i.e., 77% to 88%), when compared to the state-of-the-art methods for CMU-MultiPIE dataset. We have also explored two other similar implementations in the form of coupled CNN (cpCNN) and domain adaptation network (ADDA) for profile to frontal face recognition. We have compared the performance of the proposed approach with cpCNN, and ADDA and shown that the proposed approach performs much better than these two implementations. Moreover, we have also evaluated the frontal image reconstruction performance of the proposed approach. Finally, the improvement achieved by different losses including perceptual and GAN losses in our proposed algorithm has been investigated in an ablation study.

Chapter 8

Conclusion

In this dissertation, the concepts of deep hashing and channel coding have been explored, along with their various applications in biometric security and biometric retrieval. A deep hashing model has been integrated with hybrid secure architectures to develop a multimodal biometric secure system. The deep hashing model has also been integrated with a neural network decoder to develop a multimodal biometric authentication system. Furthermore, the deep hashing model has been extended to a deep cross-modal hashing framework and integrated with a neural network decoder for facial image retrieval using an attribute query. Finally, the dissertation also explored coupled conditional GAN architectures for application in profile-to-frontal face recognition and low-resolution to high-resolution face recognition.

This chapter summarizes the major contributions of this dissertation and also presents ideas for future work, motivated by results achieved in the previously described contributions, exploring new applications of deep hashing and neural error decoding.

8.1 Summary of Contributions

Chapter 1 provided an introductory framework for this dissertation, Chapter 2 presented an algorithm that uses a combination of deep hashing and neural network based error correction for face template protection. The proposed architecture improves upon existing face template protection techniques to provide better matching performance with one-shot and multi-shot enrollment. The novelty of this algorithm is that it can even be used for zero-shot enrollment, where the subject has not been seen during training of the deep CNN yet can be enrolled.

Chapter 3 presents a feature-level fusion and binarization framework using deep hashing to design a multimodal template protection scheme that generates a single secure template from each user's multiple biometrics. The resulting hybrid secure architecture combines the secure primitives of *cancelable biometrics* and *secure-sketch* and integrates it with a deep hashing framework, which makes it computationally prohibitive to forge a combination of multiple biometrics that passes the authentication. Two deep learning based fusion architectures have been developed, *fully connected architecture* and *bilinear architecture* that could be used to combine more than two modalities. Moreover, the matching performance and the security of the proposed secure multibiometric system have been analyzed. Experiments using the WVU multimodal dataset, which contain face and iris modalities, demonstrate that the matching performance does not deteriorate with the proposed protection scheme. In fact, both the matching performance and the template security are improved when using the proposed secure multimodal system.

Chapter 4 extends the framework from Chapter 2 for a single biometric template protection to the fusion of multiple modalities and presents a multiomodal biometric authentication system. This chapter presents a novel multimodal deep hashing neural decoder (MDHND) architecture. MDHND integrates a deep hashing framework with a neural network decoder (NND) to create an effective multibiometric authentication system. The MDHND consists of two separate modules: a multimodal deep hashing (MDH) module, which is used for feature-level fusion and binarization of multiple biometrics, and a neural network decoder (NND) module, which is used to refine the intermediate binary codes generated by the MDH and compensate for the difference between enrollment and probe biometrics (variations in pose, illumination, etc.)

Chapter 5 presents a novel iterative two-step deep cross-modal hashing method called Deep Neural Decoder Cross-Modal Hashing (DNDCMH) that takes facial attributes as query and returns a list of images based on a Hamming distance similarity. The DNDCMH network consists of two separate components: an attribute-based deep cross-modal hashing (ADCMH) module, which uses a margin (m)-based loss function to efficiently learn compact binary codes to preserve similarity between modalities (i.e., facial attribute modality and image modality) in Hamming space, and a neural error correcting decoder (NECD), which is an error correcting decoder implemented with a neural network. The goal of integrating the NECD network with the ADCMH network is to error correct the hash codes generated by ADCMH to improve the retrieval efficiency. The experimental results show that the NECD significantly improves the retrieval performance of the attribute-based deep cross-modal hashing network. Moreover, the results indicate that the proposed framework outperforms most of the other face image retrieval approaches.

Chapter 6 presents an embedding model for cross-resolution face recognition based on novel attribute-guided deep coupled learning framework using generative adversarial networks (GANs) to find the hidden relationship between the high-resolution and low-resolution images in a latent common embedding subspace. The coupled framework also exploits the facial attribute to further maximize the correlation between the low-resolution and high-resolution domains, which leads to a more discriminative embedding subspace to enhance the performance of the main task, which is cross-resolution face recognition. Additionally, in this approach, the attributes for low-resolution images can also be predicted along with cross-resolution face recognition in a multi-tasking paradigm.

Finally, Chapter 7 presents a coupled GAN architecture, which is similar to the architecture in Chapter 6, but for the application of profile-to-frontal face recognition. The coupled GAN contains two sub-networks which project the profile and frontal images into a common embedding subspace, where the goal of each sub-network is to maximize the pair-wise correlation between profile and frontal images during the process of projection. The model has been thoroughly evaluated on several standard datasets and the results demonstrate that the model notably outperforms other state-of-the-art algorithms for profile to frontal face verification.

8.2 Future Work

While this dissertation has advanced the state of the art in multibiometric security and heterogeneous face recognition, there is still room for additional work. This section provides ideas for future work for those researchers that are motivated to continue to work in this area.

Extension of multimodal biometric secure system: The multimodal secure system presented in Chapter 3 can also be extended to other modalities such as voice, fingerprint, and gait. It will be interesting to research how well the multimodal secure system is feasible for other difficult modalities such as voice, and gait. It will also be exciting to extend the framework to more than two biometric modalities and verify the scalability of such a system.

Extension of multimodal authentication system integrated with a neural network decoder: On the similar lines to the multimodal secure system, the multimodal authentication system using a neural network decoder from Chapter 4 could very well be extended for other difficult modalities such as voice and gait. It will be of great research value to extend the authentication framework to more than 2 biometric modalities and verify the scalability of the system.

Implementation of multimodal authentication system for real world: The multimodal secure system developed in Chapter 3 is a proof of concept. It will prove to be very beneficial if the proof of concept is actually implemented in the real world. It can be implemented as a Kiosk system, smartphone-based system or a cloud-based system. it is very important to understand the time complexity required to actually process a single secure authentication using this implementation in the real world on a Kiosk or a smartphone and also important to understand how this time complexity could be improved.

Integration of Deep Hashing and Neural Error Decoding for Profile to Frontal Face Verification: The problem of profile to frontal face verification can also be defined in the realm of integration of deep cross modal hashing and neural error decoding. As opposed to the generative model proposed in Chapter 7, a deep cross modal hashing (DCMH) architecture can also be implemented for profile to frontal face verification. The motivated researcher could try to have a similar architecture as in Chapter 5, but for the application of profile to frontal face verification

References

[1] S. Rane, Y. Wang, S. C. Draper, and P. Ishwar, "Secure biometrics: Concepts, authentication architectures, and challenges," *IEEE Signal Processing Magazine*, vol. 30, pp. 51–64, Sept. 2013.

[2] A. Nagar, K. Nandakumar, and A. K. Jain, "Multibiometric cryptosystems based on feature-level fusion," *IEEE Trans. on Information Forensics and Security*, vol. 7, pp. 255–268, Feb. 2012.

[3] A. Ross and A. K. Jain, "Multimodal biometrics: An overview," in *Proc. European Signal Processing Conf.*, pp. 1221–1224, Sept. 2004.

[4] Y. Sutcu, Q. Li, and N. Memon, "Protecting biometric templates with sketch: Theory and practice," *IEEE Trans. on Information Forensics and Security*, vol. 2, pp. 503–512, Sept. 2007.

[5] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. ACM Conference on Computer and Communications Security*, pp. 28–36, 1999.

[6] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. IEEE Int'l Symposium on Information Theory*, p. 408, July 2002.

[7] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Trans. on Information Forensics and Security*, vol. 2, pp. 744–757, Dec. 2007.

[8] A. Nagar, K. Nandakumar, and A. K. Jain, "Securing fingerprint template: Fuzzy vault with minutiae descriptors," in *Proc. IEEE Int'l Conf. on Pattern Recognition*, Dec. 2008.

[9] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, "Generating cancelable fingerprint templates," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 561–572, Apr. 2007.

[10] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern Recognition*, vol. 39, no. 7, pp. 1359–1368, 2006.

[11] J. Zuo, N. K. Ratha, and J. H. Connell, "Cancelable iris biometric," in *Proc. IEEE Int'l Conf. on Pattern Recognition*, pp. 1–4, 2008.

[12] A. B. Teoh, Y. W. Kuan, and S. Lee, "Cancellable biometrics and annotations on biohash," *Pattern Recognition*, vol. 41, no. 6, pp. 2034–2044, 2008.

[13] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," *IEEE Signal Processing Magazine*, vol. 32, pp. 54–65, Sept. 2015.

[14] Y. Sutcu, S. Rane, J. S. Yedidia, S. C. Draper, and A. Vetro, "Feature extraction for a Slepian-Wolf biometric system using LDPC codes," in *Proc. IEEE Int'l Symposium on Information Theory*, pp. 2297–2301, July 2008.

[15] Y. Sutcu, Q. Li, and N. Memon, "Secure biometric templates from fingerprint-face features," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2007.

[16] K. Nandakumar and A. K. Jain, "Multibiometric template security using fuzzy vault," in *Proc. IEEE Int'l Conf. on Biometric Theory, Appl. and Sys.*, Sept. 2008.

[17] A. M. Canuto, F. Pintro, and J. C. Xavier-Junior, "Investigating fusion approaches in multi-biometric cancellable recognition," *Expert Systems with Applications*, vol. 40, pp. 1971–1980, May 2013.

[18] E. J. C. Kelkboom, X. Zhou, J. Breebaart, R. N. J. Veldhuis, and C. Busch, "Multi-algorithm fusion with template protection," in *Proc. IEEE Int'l Conf. on Biometric Theory, Appl. and Sys.*, Sept. 2009.

[19] A. Gionis, P. Indyk, R. Motwani, *et al.*, "Similarity search in high dimensions via hashing," in *Proc. Int'l Conf. on Very Large Data Bases*, pp. 518–529, Sept. 1999.

[20] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2916–2929, Dec. 2013.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems*, pp. 1097–1105, Dec. 2012.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016.

[24] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1915–1929, Aug. 2013.

[25] S. Mohamadi and H. Amindavar, “Deep bayesian active learning, a brief survey on recent advances,” *arXiv preprint arXiv:2012.08044*, 2020.

[26] S. Mohamadi, D. A. Adjeroh, B. Behi, and H. Amindavar, “A new framework for spatial modeling and synthesis of genomic sequences,” in *Proc. IEEE International Conference on Bioinformatics and Biomedicine*, pp. 2221–2226, 2020.

[27] F. Taherkhani, H. Kazemi, A. Dabouei, J. Dawson, and N. M. Nasrabadi, “A weakly supervised fine label classifier enhanced by coarse supervision,” in *Proc. IEEE Int’l Conf. on Computer Vision*, October 2019.

[28] F. Taherkhani, A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi, “Transporting labels via hierarchical optimal transport for semi-supervised learning,” in *Proc. European Conference on Computer Vision* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), pp. 509–526, 2020.

[29] F. Taherkhani, A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi, “Self-supervised wasserstein pseudo-labeling for semi-supervised image classification,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 12267–12277, June 2021.

[30] S. Mohamadi, H. Amindavar, and S. A. T. Hosseini, “Arima-garch modeling for epileptic seizure prediction,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 994–998, 2017.

[31] S. Mohamadi, F. Yeganegi, and N. M. Nasrabadi, “Detection and statistical modeling of birth-death anomaly,” *arXiv preprint arXiv:1906.11788*, 2019.

[32] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, “Supervised hashing for image retrieval via image representation learning,” in *Proc. AAAI Conf. on Artificial Intelligence*, July 2014.

[33] H. Lai, Y. Pan, Y. Liu, and S. Yan, “Simultaneous feature learning and hash coding with deep neural networks,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3270–3278, June 2015.

[34] K. Lin, J. Lu, C. S. Chen, and J. Zhou, “Learning compact binary descriptors with unsupervised deep neural networks,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1183–1192, June 2016.

[35] H. Liu, R. Wang, S. Shan, and X. Chen, “Deep supervised hashing for fast image retrieval,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2064–2072, June 2016.

[36] H. Zhu, M. Long, J. Wang, and Y. Cao, “Deep hashing network for efficient similarity retrieval.,” in *Proc. AAAI Conf. on Artificial Intelligence*, pp. 2415–2421, Feb. 2016.

[37] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. AAAI Conf. on Artificial Intelligence*, Feb. 2017.

[38] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3232–3240, June 2017.

[39] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. AAAI Conf. on Artificial Intelligence*, pp. 2177–2183, July 2014.

[40] Y. Zhen and D.-Y. Yeung, "Co-regularized hashing for multimodal data," in *Proc. Advances in Neural Information Processing Systems*, pp. 1376–1384, Dec. 2012.

[41] Y. Cao, M. Long, J. Wang, and S. Liu, "Collective deep quantization for efficient cross-modal retrieval," in *Proc. AAAI Conf. on Artificial Intelligence*, 2017.

[42] K. Wang, Q. Yin, W. Wang, S. Wu, and L. Wang, "A comprehensive survey on cross-modal retrieval," *arXiv preprint arXiv:1607.06215*, 2016.

[43] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 341–346, Sept. 2016.

[44] E. Nachmani, E. Marciano, D. Burshtein, and Y. Be'ery, "RNN decoding of linear block codes," *arXiv preprint arXiv:1702.07560*, 2017.

[45] L. Lugosch and W. J. Gross, "Neural offset min-sum decoding," *Proc. IEEE Int'l Symposium on Information Theory*, pp. 1361–1365, June 2017.

[46] T. Gruber, S. Cammerer, J. Hoydis, and S. t. Brink, "On deep learning-based channel decoding," in *Proc. IEEE Annual Conference on Information Sciences and Systems (CISS)*, 2017.

[47] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, pp. 563–575, Dec. 2017.

[48] V. Talreja, M. C. Valenti, and N. M. Nasrabadi, "Zero-shot deep hashing and neural network based error correction for face template protection," in *Proc. IEEE Int'l Conf. on Biometric Theory, Appl. and Sys.*, Sept. 2019.

[49] V. Talreja, M. C. Valenti, and N. M. Nasrabadi, "Multibiometric secure system based on deep learning," in *Proc. IEEE Global Conf. on Signal and Information Processing*, pp. 298–302, Nov. 2017.

[50] V. Talreja, M. C. Valenti, and N. M. Nasrabadi, "Deep hashing for secure multimodal biometrics," *IEEE Trans. on Information Forensics and Security*, vol. 16, pp. 1306–1321, 2021.

[51] V. Talreja, S. Soleymani, M. C. Valenti, and N. M. Nasrabadi, "Learning to authenticate with deep multibiometric hashing and neural network decoding," in *Proc. IEEE Int'l Conf. on Commun.*, May 2019.

[52] F. Taherkhani, V. Talreja, H. Kazemi, and N. Nasrabadi, "Facial attribute guided deep cross-modal hashing for face image retrieval," in *Proc. IEEE Int'l Conf. of the Biometrics Special Interest Group*, pp. 1–6, Sept. 2018.

[53] V. Talreja, F. Taherkhani, M. C. Valenti, and N. M. Nasrabadi, "Using deep cross modal hashing and error correcting codes for improving the efficiency of attribute guided facial image retrieval," in *Proc. IEEE Global Conf. on Signal and Information Processing*, pp. 564–568, Nov. 2018.

[54] F. Taherkhani, V. Talreja, M. C. Valenti, and N. M. Nasrabadi, "Error-corrected margin-based deep cross-modal hashing for facial image retrieval," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, pp. 279–293, Apr. 2020.

[55] V. Talreja, F. Taherkhani, M. C. Valenti, and N. M. Nasrabadi, "Attribute-guided coupled gan for cross-resolution face recognition," in *Proc. IEEE Int'l Conf. on Biometric Theory, Appl. and Sys.*, Sept. 2019.

[56] F. Taherkhani, V. Talreja, J. Dawson, M. C. Valenti, and N. M. Nasrabadi, "Pf - cpgan: Profile to frontal coupled gan for face recognition in the wild," in *Proc. Int'l Joint Conf. on Biometrics*, Oct. 2020.

[57] V. Talreja, T. Ferrett, M. C. Valenti, and A. Ross, "Biometrics-as-a-service: A framework to promote innovative biometric recognition in the cloud," in *Proc. IEEE International Conference on Consumer Electronics*, Jan. 2018.

[58] S. Soleymani, A. Dabouei, S. M. Iranmanesh, H. Kazemi, J. Dawson, and N. M. Nasrabadi, "Prosodic-enhanced Siamese convolutional neural networks for cross-device text-independent speaker verification," in *Proc. IEEE Int'l Conf. on Biometric Theory, Appl. and Sys.*, Oct. 2018.

[59] B. Chaudhary, P. Aghdaie, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Differential morph face detection using discriminative wavelet sub-bands," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 1425–1434, 2021.

[60] P. Aghdaie, B. Chaudhary, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Attention aware wavelet-based detection of morphed face images," in *Proc. Int'l Joint Conf. on Biometrics*, 2021.

[61] P. Aghdaie, B. Chaudhary, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Detection of morphed face images using discriminative wavelet sub-bands," *arXiv preprint arXiv:2106.08565*, 2021.

[62] R. K. Pandey and V. Govindaraju, "Secure face template generation via local region hashing," in *Proc. IAPR Int'l Conf. on Biometrics*, pp. 299–304, May 2015.

[63] R. K. Pandey, Y. Zhou, B. U. Kota, and V. Govindaraju, "Deep secure encoding for face template protection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 77–83, June 2016.

[64] A. K. Jindal, S. Chalamala, and S. K. Jami, "Face template protection using deep convolutional neural network," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 575–5758, June 2018.

[65] F. Taherkhani, N. M. Nasrabadi, and J. Dawson, "A deep face identification network enhanced by facial attributes prediction," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, June 2018.

[66] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," in *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 5609–5618, Oct. 2017.

[67] X. Yuan, L. Ren, J. Lu, and J. Zhou, "Relaxation-free deep hashing via policy gradient," in *Proc. The European Conference on Computer Vision*, Sept. 2018.

[68] Z. Chen, X. Yuan, J. Lu, Q. Tian, and J. Zhou, "Deep hashing via discrepancy minimization," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 6838–6847, June 2018.

[69] H. Kazemi, S. Soleymani, F. Taherkhani, S. Iranmanesh, and N. Nasrabadi, "Unsupervised image-to-image translation using domain-specific variational information bound," in *Advances in Neural Information Processing Systems*, pp. 10348–10358, Dec. 2018.

[70] H. Kazemi, F. Taherkhani, and N. M. Nasrabadi, "Unsupervised facial geometry learning for sketch to photo synthesis," in *Proc. IEEE Int'l Conf. of the Biometrics Special Interest Group*, Sept. 2018.

[71] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[72] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, Dec. 2009.

[73] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," in *Proc. IEEE Int. Conf. on Automatic Face Gesture Recognition*, pp. 53–58, May 2002.

[74] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. on Pattern Analysis and Mach. Intelligence*, vol. 23, pp. 643–660, June 2001.

[75] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-Pie," in *Proc. IEEE Int'l Conf. on Automatic Face Gesture Recognition*, Sept. 2008.

[76] A. Dabouei, S. Soleymani, F. Taherkhani, and N. M. Nasrabadi, "Supermix: Supervising the mixing data augmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 13794–13803, June 2021.

[77] U. M. Osahor and N. M. Nasrabadi, “Design of adversarial targets: fooling deep ATR systems,” in *Automatic Target Recognition XXIX*, 2019.

[78] S. N. Ferdous, M. Mostofa, and N. M. Nasrabadi, “Super resolution-assisted deep aerial vehicle detection,” in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019.

[79] Y. C. Feng, P. C. Yuen, and A. K. Jain, “A hybrid approach for generating secure and discriminating face template,” *IEEE Trans. on Inform. Forensics and Sec.*, vol. 5, Mar. 2010.

[80] Y. C. Feng and P. C. Yuen, “Binary discriminant analysis for generating binary face template,” *IEEE Trans. on Inform. Forensics and Sec.*, vol. 7, pp. 613–624, Apr. 2012.

[81] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[82] K. D. Nguyen, C. Fookes, A. Ross, and S. Sridharan, “Iris recognition with off-the-shelf CNN features: A deep learning perspective,” *IEEE Access*, vol. 6, pp. 18848–18855, 2017.

[83] Z. Zhao and A. Kumar, “Towards more accurate iris recognition using deeply learned spatially corresponding features,” in *Proc. IEEE Int’l Conf. on Computer Vision*, pp. 3809–3818, 2017.

[84] S. Minaee, A. Abdolrashidiy, and Y. Wang, “An experimental study of deep convolutional features for iris recognition,” in *IEEE Signal Processing in Medicine and Biology Symposium*, 2016.

[85] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 815–823, June 2015.

[86] Y. Sun, D. Liang, X. Wang, and X. Tang, “Deepid3: Face recognition with very deep neural networks,” *arXiv preprint arXiv:1502.00873*, 2015.

[87] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *Proc. of the British Machine Vision Conf.*, Sept. 2015.

[88] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Convolutional two-stream network fusion for video action recognition,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[89] T. Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear CNN models for fine-grained visual recognition,” in *Proc. IEEE Int’l Conf. on Computer Vision*, pp. 1449–1457, Dec. 2015.

[90] "WVU multimodal dataset." Available at `http://biic.wvu.edu/`.

[91] K. W. Bowyer and P. J. Flynn, "The ND-IRIS-0405 iris image dataset," *CVRL, University of Notre Dame*, 2010.

[92] N. Othman, B. Dorizzi, and S. Garcia-Salicetti, "Osiris: An open source iris recognition software," *Pattern Recognition Letters*, vol. 82, pp. 124–131, 2016.

[93] H. Yang, K. Lin, and C. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 437–451, Feb. 2018.

[94] M. Gomez-Barrero, J. Galbally, C. Rathgeb, and C. Busch, "General framework to evaluate unlinkability in biometric template protection systems," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 6, pp. 1406–1420, 2018.

[95] M. Mostofa, F. Taherkhani, J. Dawson, and N. M. Nasrabadi, "Cross-spectral iris matching using conditional coupled gan," in *Proc. Int'l Joint Conf. on Biometrics*, 2020.

[96] Y. Xin, L. Kong, Z. Liu, C. Wang, H. Zhu, M. Gao, C. Zhao, and X. Xu, "Multimodal feature-level fusion for biometrics identification system on iomt platform," *IEEE Access*, vol. 6, pp. 21418–21426, Mar. 2018.

[97] Y. Shi and R. Hu, "Rule-based feasibility decision method for big data structure fusion: Control method.," *International Journal of Simulation–Systems, Science & Technology*, vol. 17, no. 31, 2016.

[98] S. Soleymani, A. Dabouei, H. Kazemi, J. Dawson, and N. M. Nasrabadi, "Multi-level feature abstraction from convolutional neural networks for multimodal biometric identification," in *Proc. IEEE Int'l Conf. on Pattern Recognition*, pp. 3469–3476, Aug. 2018.

[99] F. Taherkhani and M. Jamzad, "Restoring highly corrupted images by impulse noise using radial basis functions interpolation," *IET Image Processing*, vol. 12, no. 1, pp. 20–30, 2018.

[100] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 119–131, Feb. 2018.

[101] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[102] C. Liu and H. Wechsler, "A shape and texture-based enhanced Fisher classifier for face recognition," *IEEE Trans. on Image Processing*, vol. 10, pp. 598–608, Apr. 2001.

[103] J. Yang, J. Yang, D. Zhang, and J. Lu, "Feature fusion: parallel strategy vs. serial strategy," *Pattern Recognition*, vol. 36, pp. 1369–1381, 2003.

[104] Q.-S. Sun, S.-G. Zeng, Y. Liu, P.-A. Heng, and D.-S. Xia, "A new method of feature fusion and its application in image recognition," *Pattern Recognition*, vol. 38, pp. 2437–2448, Dec. 2005.

[105] M. Haghighat, M. Abdel-Mottaleb, and W. Alhalabi, "Discriminant correlation analysis: Real-time feature level fusion for multimodal biometric recognition," *IEEE Trans. on Information Forensics and Security*, vol. 11, pp. 1984–1996, Sept. 2016.

[106] A. Mackiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Computers and Geosciences*, vol. 19, pp. 303–342, 1993.

[107] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, Jan. 1967.

[108] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. IEEE Symposium on Foundations of Computer Science*, pp. 459–468, 2006.

[109] Y. Gong, S. Kumar, V. Verma, and S. Lazebnik, "Angular quantization-based binary codes for fast similarity search," in *Proc. Advances in Neural Information Processing Systems*, pp. 1196–1204, 2012.

[110] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2938–2945, June 2013.

[111] P. Jain, B. Kulis, and K. Grauman, "Fast image search for learned metrics," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2008.

[112] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Advances in Neural Information Processing Systems*, pp. 1042–1050, Dec. 2009.

[113] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proc. 28th International Conference on Machine Learning*, pp. 353–360, July 2011.

[114] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Proc. Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1061–1069, 2012.

[115] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. Advances in Neural Information Processing Systems*, pp. 1509–1517, Dec. 2009.

[116] J. Wang, S. Kumar, and S. F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3424–3431, June 2010.

[117] J. Wang, S. Kumar, and S. F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2393–2406, Dec. 2012.

[118] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in *Proc. Advances in Neural Information Processing Systems*, pp. 1753–1760, Dec. 2009.

[119] Z. Lin, G. Ding, M. Hu, and J. Wang, “Semantics-preserving hashing for cross-view retrieval,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3864–3872, June 2015.

[120] Z. Cao, M. Long, J. Wang, and Q. Yang, “Transitive hashing network for heterogeneous multimedia retrieval,” in *Proc. AAAI Conf. on Artificial Intelligence*, 2017.

[121] Z.-D. Chen, W.-J. Yu, C.-X. Li, L. Nie, and X.-S. Xu, “Dual deep neural networks cross-modal hashing,” in *Proc. AAAI Conf. on Artificial Intelligence*, 2018.

[122] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, “Cross-modal deep variational hashing,” in *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 4097–4105, 2017.

[123] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, “Triplet-based deep hashing network for cross-modal retrieval,” *IEEE Trans. on Image Processing*, vol. 27, no. 8, pp. 3893–3903, 2018.

[124] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu, “Deep visual-semantic hashing for cross-modal retrieval,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1445–1454, ACM, 2016.

[125] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009.

[126] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” in *Workshop on faces in ‘Real-Life’ Images: Detection, Alignment, and Recognition*, 2008.

[127] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proc. IEEE Int'l Conf. on Computer Vision*, Dec. 2015.

[128] B. Siddiquie, R. S. Feris, and L. S. Davis, “Image ranking and retrieval based on multi-attribute queries,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 801–808, June 2011.

[129] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, “An efficient boosting algorithm for combining preferences,” *Journal of Machine Learning Research*, vol. 4, no. Nov., pp. 933–969, 2003.

[130] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation,” in *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 309–316, Sept. 2009.

[131] S. Ge, S. Zhao, C. Li, and J. Li, “Low-resolution face recognition in the wild via selective knowledge distillation,” *IEEE Trans. on Image Processing*, vol. 28, pp. 2051–2062, 2018.

[132] S. Kolouri and G. K. Rohde, "Transport-based single frame super resolution of very low resolution face images," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4876–4884, 2015.

[133] M. Jian and K.-M. Lam, "Simultaneous hallucination and recognition of low-resolution faces based on singular value decomposition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1761–1772, 2015.

[134] M.-C. Yang, C.-P. Wei, Y.-R. Yeh, and Y.-C. F. Wang, "Recognition at a long distance: Very low resolution face recognition and hallucination," in *Proc. IAPR Int'l Conf. on Biometrics*, pp. 237–242, 2015.

[135] M. Singh, S. Nagpal, M. Vatsa, R. Singh, A. Majumdar, and IIIT-Delhi, "Identity aware synthesis for cross resolution face recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 479–488, 2018.

[136] T. Uiboupin, P. Rasti, G. Anbarjafari, and H. Demirel, "Facial image super resolution using sparse representation for improving face recognition in surveillance monitoring," in *Proc. IEEE Signal Processing and Communication Application Conference*, pp. 437–440, 2016.

[137] C. Dong, C. C. Loy, K. He, X. Tang, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, "Learning a deep convolutional network for image super-resolution," in *Proc. European Conference on Computer Vision*, pp. 184–199, 2014.

[138] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016.

[139] C.-X. Ren, D.-Q. Dai, and H. Yan, "Coupled kernel embedding for low-resolution face image recognition," *IEEE Trans. on Image Processing*, vol. 21, no. 8, pp. 3770–3783, 2012.

[140] A. Dantcheva, P. Elia, and A. Ross, "What else does your biometric data reveal? a survey on soft biometrics," *IEEE Trans. on Information Forensics and Security*, vol. 11, pp. 441–467, Mar. 2016.

[141] F. Taherkhani, H. Kazemi, and N. M. Nasrabadi, "Matrix completion for graph-based deep semi-supervised learning," in *Proc. AAAI Conf. on Artificial Intelligence*, 2019.

[142] S. Soleymani, A. Dabouei, J. Dawson, and N. M. Nasrabadi, "Adversarial examples to fool iris recognition systems," in *Proc. IAPR Int'l Conf. on Biometrics*, 2019.

[143] H. Kazemi, S. M. Iranmanesh, and N. Nasrabadi, "Style and content disentanglement in generative adversarial networks," in *Proc. IEEE Winter Conf. on Applications of Computer Vision*, pp. 848–856, 2019.

[144] U. M. Osahor and N. M. Nasrabadi, "Deep adversarial attack on target detection systems," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019.

[145] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Neural Information Processing Systems*, pp. 2672–2680, 2014.

[146] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[147] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning*, 2016.

[148] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 5967–5976, 2017.

[149] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. European Conference on Computer Vision*, 2016.

[150] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

[151] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 539–546, 2005.

[152] Y. Zhong, J. Sullivan, and H. Li, "Face attribute prediction using off-the-shelf cnn features," in *Proc. IAPR Int'l Conf. on Biometrics*, 2016.

[153] L. Wolf, T. Hassner, and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1978–1990, Oct. 2011.

[154] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int'l Conf. on Computer Vision*, Dec. 2015.

[155] M. Grgic, K. Delac, and S. Grgic, "SCface – surveillance cameras face database," *Multimedia Tools and Applications*, vol. 51, pp. 863–879, 2009.

[156] A. Sapkota and T. E. Boult, "Large scale unconstrained open set face database," in *Proc. IEEE Int'l Conf. on Biometric Theory, Appl. and Sys.*, Sept. 2013.

[157] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang, "Studying very low resolution recognition using deep networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4792–4800, 2016.

[158] M. Haghighat and M. Abdel-Mottaleb, "Low resolution face recognition in surveillance systems using discriminant correlation analysis," in *Proc. IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 912–917, 2017.

[159] P. Li, L. Prieto, D. Mery, and P. J. Flynn, "On low-resolution face recognition in the wild: Comparisons and new techniques," *IEEE Trans. on Information Forensics and Security*, vol. 14, no. 8, pp. 2000–2012, 2019.

[160] S. P. Mudunuri, S. Venkataramanan, and S. Biswas, "Dictionary alignment with reranking for low-resolution nir-vis face recognition," *IEEE Trans. on Information Forensics and Security*, vol. 14, pp. 886–896, Apr. 2019.

[161] E. M. Rudd, M. Günther, and T. E. Boult, "Moon: A mixed objective optimization network for the recognition of facial attributes," in *Proc. European Conference on Cmputer Vision*, 2016.

[162] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy, "Pose-robust face recognition via deep residual equivariant mapping," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 5187–5196, 2018.

[163] S. Sengupta, J. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *Proc. IEEE Winter Conf. on Applications of Computer Vision*, pp. 1–9, Mar. 2016.

[164] F. Taherkhani, J. Dawson, and N. M. Nasrabadi, "Hyperspectral band selection for face recognition based on a structurally sparsified deep convolutional neural networks," in *Proc. IAPR Int'l Conf. on Biometrics*, 2019.

[165] I. Masi, S. Rawls, G. Medioni, and P. Natarajan, "Pose-aware face recognition in the wild," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4838–4846, June 2016.

[166] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1701–1708, June 2014.

[167] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1283–1292, July 2017.

[168] J. Yim, H. Jung, B. Yoo, C. Choi, D.-S. Park, and J. Kim, "Rotating your face using multi-task deep neural network," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 676–684, 2015.

[169] X. Yin, X. Yu, K. Sohn, X. Liu, and M. K. Chandraker, "Towards large-pose face frontalization in the wild," *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 4010–4019, 2017.

[170] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman, "Synthesizing normalized faces from facial identity features," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3386–3395, 2017.

[171] Y. Qian, W. Deng, and J. Hu, "Unsupervised face normalization with extreme pose and expression in the wild," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[172] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: A deep model for learning face identity and view representations," in *Proc. International Conference on Neural Information Processing Systems*, p. 217–225, 2014.

[173] M. Kan, S. Shan, and X. Chen, "Multi-view deep network for cross-view classification," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4847–4855, 2016.

[174] X. Peng, X. Yu, K. Sohn, D. N. Metaxas, and M. Chandraker, "Reconstruction-based disentanglement for pose-invariant face recognition," in *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 1632–1641, 2017.

[175] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. on Image Processing*, vol. 27, pp. 964–975, 2017.

[176] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning identity-preserving face space," in *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 113–120, 2013.

[177] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[178] Mallikarjun B R, V. Chari, and C. V. Jawahar, "Efficient face frontalization in unconstrained images," in *Proc. National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pp. 1–4, 2015.

[179] Xiangyu Zhu, Z. Lei, Junjie Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 787–796, 2015.

[180] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3722–3731, 2017.

[181] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee, "Weakly-supervised disentangling with recurrent transformations for 3d view synthesis," in *Advances in Neural Information Processing Systems*, pp. 1099–1107, 2015.

[182] J. Zhao, Y. Cheng, Y. Xu, L. Xiong, J. Li, F. Zhao, K. Jayashree, S. Pranata, S. Shen, J. Xing, S. Yan, and J. Feng, "Towards pose invariant face recognition in the wild," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2207–2216, June 2018.

[183] S. Shekhar, V. M. Patel, and R. Chellappa, “Synthesis-based robust low resolution face recognition,” *arXiv preprint arXiv:1707.02733*, 2017.

[184] P. Zhang, X. Ben, W. Jiang, R. Yan, and Y. Zhang, “Coupled marginal discriminant mappings for low-resolution face recognition,” *Optik*, vol. 126, no. 23, pp. 4352–4357, 2015.

[185] J. Jiang, R. Hu, Z. Wang, and Z. Cai, “Cdmma: Coupled discriminant multi-manifold analysis for matching low-resolution face images,” *Signal Processing*, vol. 124, pp. 162–172, 2016.

[186] Y. Hu, X. Wu, B. Yu, R. He, and Z. Sun, “Pose-guided photorealistic face rotation,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 8398–8406, June 2018.

[187] B. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. E. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain, “Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1931–1939, June 2015.

[188] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother, “IARPA Janus benchmark - c: Face dataset and protocol,” in *Proc. IAPR Int’l Conf. on Biometrics*, pp. 158–165, Feb. 2018.

[189] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother, “IARPA Janus benchmark-b face dataset,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 592–600, July 2017.

[190] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, June 2005.

[191] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 2037–2041, Dec. 2006.

[192] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Fisher vector faces in the wild,” in *Proc. of the British Machine Vision Conf.*, 2013.

[193] Q. Cao, Y. Ying, and P. Li, “Similarity metric learning for face recognition,” in *Proc. IEEE Int’l Conf. on Computer Vision*, pp. 2408–2415, Dec. 2013.

[194] J. Chen, V. M. Patel, and R. Chellappa, “Unconstrained face verification using deep CNN features,” in *Proc. IEEE Winter Conf. on Applications of Computer Vision*, pp. 1–9, Mar. 2016.

[195] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1931–1939, June 2015.

[196] J.-C. Chen, V. M. Patel, and R. Chellappa, "Unconstrained face verification using deep CNN features," in *Proc. IEEE Winter Conf. on Applications of Computer Vision*, pp. 1–9, 2016.

[197] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Towards large-pose face frontalization in the wild," in *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 3990–3999, 2017.

[198] C. Ding and D. Tao, "Pose-invariant face recognition with homography-based normalization," *Pattern Recognition*, vol. 66, pp. 144–152, 2017.

[199] C. Xiong, X. Zhao, D. Tang, K. Jayashree, S. Yan, and T.-K. Kim, "Conditional convolutional neural network for modality-aware face recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3667–3675, June 2015.

[200] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2439–2448, June 2017.

[201] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–9, June 2015.

[202] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *International Conference on Automatic Face and Gesture Recognition*, 2018.

[203] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 7167–7176, June 2017.