SECURITY AND PROTECTION OF FACIAL BIOMETRICS SYSTEMS

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Electronics and Communication Engineering

by

Srinivasa Rao Chalamala
200850022
srinivas.chalamala@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
MARCH 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **”Security and Protection of Facial Biometrics Systems”** by Srinivasa Rao Chalamala, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Bayya Yegnanarayana

I dedicate this work to my wife, Padma Gayatri, and my children, Sree Sanjana, Sadhana and Bhargava Nandan

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Bayya Yegnanarayana for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. Every time I meet him, my motivation gets a boost. He taught us how research can be carried out effectively. I could not have imagined having a better advisor and mentor for my Ph.D. study.

I thank IIIT Hyderabad, for providing me an opportunity to pursue my PhD. Special thanks to the Director Prof. P J Narayanan, and Dean (R&D) Prof. C V Jawahar, Former Director Prof. Rajeev Sangal, and Prof. Ponnurangam Kumaraguru for providing me the opportunity to pursue PhD. I want express my sincere thanks to all non-teaching staff of IIIT Hyderabad (Late Mr. Phani, Ms. Prathima, Mr. Y.Kishore, Ms. Pushpalatha, Mr. Appaji and others in accounts section) for all the help rendered during my stay at IIIT.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Vijayakumar Bhagavatula, Prof. C. Krishna Mohan, Dr. Ashok Kumar Das, Prof. Anoop M. Namboodiri, for their encouragement, insightful comments during the review. I thank Dr. Suryakanth V Gangashetty for providing ample resources at Speech lab.

I thank my fellow lab mates of Speech and Vision lab Dr. Anil Sao, Dr. Guru Prasad, Dr. Dhanunjay, Dr. Anand Joseph, Dr. Nivedita, Dr. Gangamohan, Dr. Ravi Shankar Prasad for all their support and discussions. Special thanks to Dr. Vishala Pannala for helping me on various aspects related to my research.

I also would like to express my thanks to my colleagues, Bala Mallikarjuna Rao, Ajeet Singh, Krishna Rao K, Arun Kumar Jindal, Santosh Kumar Jami, Balakrishna Gudla, and my supervisors Dr. Sachin Lodha and Sitaram Chamarty for their continued support through out.

Last but not the least, special thanks to my wife Gayatri for her continuous support. I would like express my gratitude to my parents (late) Venkateswarlu and Sitaravamma who are the important pillars of my life and my greatest strength.

Abstract

Biometrics are physical and behavioral traits, which are unique and specific to individuals. Some of the widely used physical traits include face, fingerprint, iris, and behavioral traits include voice, signature, typing rhythm, and gait. Identifying a person based on any physical and behavioral traits is referred to as biometric authentication.

Biometrics authentication can be more convenient and secure than passwords, because biometric traits are relatively fixed and cannot be easily stolen or shared. However, biometrics cannot be recovered and lost forever when compromised. Attackers strive to subvert the biometric systems and gain unauthorized access to digital and physical assets. Attacks on biometric systems can be classified into impersonation and obfuscation attacks. These attacks are the result of the following biometric vulnerabilities (i) An attacker can exploit a compromised template database either, to replace a template with an imposter template or, to present a stolen template directly to the matching module, (ii) Invertible transform function could lead to the estimation of biometric features, which can be used to create a physical fake or spoof of the biometric, and (iii) Attackers also exploit a higher false accept rate to impersonate a victim user, and (iv) Finally, biometric systems are sensitive to carefully crafted perturbations to the input biometric data. These perturbations can be used for impersonation attacks, as well as obfuscation attacks. Researchers proposed several template protection methods to overcome some of the above vulnerabilities of biometric systems and to defend against adversarial attacks. A template protection method converts an original template into a protected template in a non-invertible manner, intending to protect the biometric identifier even if a template is stolen. An ideal template protection mechanism must meet the following requirements: (i) Security or non-invertibility, (ii) Revocability, (iii) Diversity, and (iv) Matching performance.

This thesis addresses some of these issues and proposes methods for facial template protection and for defending adversarial attacks on facial verification systems. The following studies were conducted in this thesis. (i) A modular Siamese network based method is proposed to improve the robustness of the face verification systems against adversarial attacks and simultaneously provide interpretability. In this approach, facial feature representations for each individual facial part such as eyes, nose and mouth are learned in latent space through feature disentanglement. (ii) A template protection method based on deep neural

networks is proposed, to improve the security of the biometric template without compromising on the matching performance. Another deep neural networks based template protection method is proposed, for which ancillary data is derived from the adversarial perturbations, (iii) A random projection based approach proposed for improving the non-invertibility of facial feature vectors, to prevent the reconstruction of biometric features, (iv) Federated learning for face recognition and its security and privacy implications are explored, (v) A novel facial feature descriptor based on local binary patterns has been proposed for face recognition and its application to gender classification is discussed.

Contents

Chapter Page

List of Figures

Figure Page

List of Tables

List of Symbols and Abbreviations

BCE Binary Cross Entropy

CCPA California Consumer Protection Act

DOG Difference of Gaussians

DP Differential Privacy

EER Equal Error Rate

FAR False Accept Rate

FFGSM Fast Fast Gradient Sign Method

FGSM Fast Gradient Sign Method

FHE Fully Homomorphic Encryption

FL Federated Learning

FMR False Match Rate

FNMR False Non-match Rate

FRR False Reject Rate

GDPR General Data Protection Rights

GAR Genuine Accept Rate

HoG Histogram of Gradients

LDA Linear Discriminant Analysis

LLE Locally Linear Embedding

IoU Intersection of Union

KPCA Kernel PCA

LBP local Binary Pattern

LIME Local Interpretable Model-Agnostic Explanations

LSH Locality Sensitive Hashing

LRP Layer wise Relevant Propagation

MMOD Max-Margin Object Detection

MSN Modular Siamese Networks

MNN Modular Neural Network

MPC Multi-Party Compuatation

MTCNN Multi-Task Cascaded Neural Network

NMS Non Maximal Suppression

PCA Principal Component Analysis

PLDA Probabilistic Liner Discriminant Analysis

RoI Region of Interest

RP Random Projections

SHAP SHapley Additive exPlanations

SSD Single Shot Multi-box Detector

SVM Support Vector Machine

XLBP Cross Local Binary Pattern

XLRBP Cross Local Radon Binary Pattern

XLGBP Cross Local Gabor Binary Pattern

Chapter 1

Introduction

Biometric authentication refers to verifying the identity of individuals based on their physiological (face, iris, fingerprint) or behavioral (gait, signature) traits. These are also called as primary or hard biometrics. Another category of biometrics, called soft biometrics, provide secondary information such as gender, ethnicity, skin color, etc., to augment the primary traits. The secondary traits alone cannot provide reliable authentication. Face biometric is used for illustration in the studies conducted in this thesis.

A biometric authentication system consists of a sensor, feature extractor, template generator, template matching, and decision making. The sensor captures the user's biometric data, and salient features of the data are extracted by the feature extractor. A template is prepared from the features and is stored in the database. A template is a compact representation of the sensed biometric trait, containing descriptive and discriminating information suitable for person authentication. During verification, the template matching produces a similarity measure between the presented biometric and the stored template. Based on the similarity measure, the decision making strategy gives a response as either accept or reject the claim of the user.

Biometric systems can be more secure and convenient than tokens and passwords, as it is not required to memorize them for authentication. It is easy to steal or trick people into giving the tokens or passwords. But, it is difficult to replicate a biometric trait. While biometric authentication may be more secure than using passwords, but it is not fool-proof as yet. There can be several types of attacks that make a biometric authentication system insecure. These attacks can be divided into two broad categories - (i) Impersonation attacks, (ii) Obfuscation attacks. These attacks are the consequence of the following vulnerabilities in a biometric system.

1. **Compromised templates**: Compromised templates in the database could lead to impersonation attacks. An adversary can replace a biometric template in the database by an imposter template for gaining unauthorized access. The attacker can replay

the victim's stolen template for gaining access. This type of attacks are called replay attacks.

2. **Invertible processes**: If one or more stages of a biometric system use an invertible transform or process, then the adversary may reconstruct the biometric data either from the template or from any intermediate data such as feature vectors. The reconstructed data could be used for replay attack or to create a physical fake of the biometric and present it, resulting spoofing attack.

3. **False match**: False match rate (FMR) refers to the frequency of a false biometric matching with the stored template. High FMR could lead to impersonation attacks. Often the higher FMR is due to lower inter-class variability and higher intra-class variability. An adversary may intentionally create variations in the presented biometric by exploiting the high FMR for impersonating a victim.

4. **True reject**: Biometric systems are sensitive to carefully crafted perturbations to the biometric input, resulting in true rejections. The process of creating these perturbations are called adversarial attacks. These perturbations can also be used to impersonate others in the database.

Attackers can exploit these vulnerabilities either to steal the biometric data causing privacy issues or maliciously gain access to the system resulting in security issues. To be more effective, a biometric system should meet the following requirements: (i) security or non-invertibility (inability to invert any transform operation to gain access to the input), (ii) revocability or renewability (cancel and regenerate new templates) (iii) diversity (uniqueness of the template for a user), and (iv) performance (in terms of measures like false accepts (FA) and false rejects (FR). This thesis addresses some of these issues and proposes methods for template protection, non-invertibility, and defenses against adversarial attacks. These proposed methods are discussed in the context of face biometric. The following studies are conducted in this thesis.

1. **Feature extraction:** A novel facial feature descriptor is proposed using local binary patterns (LBP). The descriptor is a combination of local binary patterns and Radon transform. The proposed feature descriptor gives 9% improvement for face recognition over the normal LBP based descriptors. The same feature descriptor is used for gender detection with good performance.

2. **Template protection:** The basic idea in template protection is to store a transformed and encrypted template in the database, instead of the original template. This is called a protected template. If the protected template is compromised, then it should be

computationally hard to retrieve the raw biometric from the transformed template. The compromised template is revoked, and the user will be enrolled again.

A deep neural network (DNN) based transformation is used to map the input facial features of an individual to a pre-generated binary code. This binary code is the original unprotected template. The binary code generation process ensures the diversity requirement of template protection. A cryptographic hash, then, is applied on the unprotected template to obtain a secure template. A 6% improvement in matching performance is achieved with the proposed secure template generation mechanism.

Adversarial perturbations, based on input data, are obtained as auxiliary data and is stored in the database along with the secure target template. This is to prevent the reconstruction of biometric features from them. The auxiliary data along with the secured template are used for arriving at template matching performance.

Whenever the protected template is compromised, then the template along with the auxiliary data is removed and the user will be enrolled again.

3. **Non-invertibility:** A non-invertible transformation can improve the security at intermediate levels of processing of the biometric data. A random projection matrix is generated based on a unique key associated with each user. The random projection matrix on the feature vector produces a reduced dimensional vector preserving the distance between the input vectors. This will prevent reconstruction of biometric features from intermediate data.

4. **Adversarial attacks:** Carefully crafted adversarial perturbations can trick a biometric system either to reject a genuine user or to falsely accept an imposter, leading to impersonation attacks. Modular networks and feature disentanglement approaches are proposed to improve the robustness of the system against these attacks.

Many countries have laws constraining the collection, use, and disclosure of many types of personal data. The processing or use of personal data is often subject to explicit requirements and protective measures. These data privacy laws impose restrictions on when and under what circumstances organizations may collect sensitive personal information, the purposes for which it can be used, and whether or not individuals first need to give consent before their data can be stored, processed, or disclosed. Policies on data ownership, informed consent, confidentiality, and security would be beneficial for identifying liabilities.

Regulations: General Data Protection Regulations (GDPR) [10], provides rights to data owners to control their data. The California Consumer Privacy Act (CCPA) [11] is about creating transparency and rights to its consumers. The principal user rights of the CCPA and GDPR include the right to be informed, the right to access, the right to deletion, the right to prior consent, right to opt-out. With rigorous requirements of these regulations, machine learning (ML) algorithms shall be designed to accommodate sufficient privacy-preserving techniques. India's **NITI Aayog** [12, 13] document on *Resposible AI* recommends that AI should maintain the privacy and security of data of individuals or entities that are used for training the system.

Ethical Issues: While the data protection regulations broadly cover the aspects of privacy in handling the users' data, they may not cover some aspects and may result in ambiguity in interpretation based on the regional, cultural practices. Organizations while adhering to the regulations may take a step ahead and follow ethical principles in handling the users' data for privacy. Recently, EU's HLEG, on AI recommended guidelines to promote Trustworthy AI. Particularly on privacy, it recommends to (i) respect for privacy, (ii) quality and integrity of data, and (iii) access to data. Organizations developing AI solutions are recommended to follow human-centered approach and values.

To comply with the above regulations and ethical practices, organizations have to come up with processes that can protect the data both in static and moving forms. As the number of organizations adapting to deep learning technology due to their exceptional performance, they need to collect huge amounts of data. But, several regulations prevent them collecting and storage of private and sensitive data without the consent of the users. However, the users concerned with the safety of the data may not contribute their (biometric) data to the deep learning based models. Federate learning, a new technology helps the organizations in learning their deep learning models without obtaining a copy of the data. The learning happens at the user site/device, eliminating the privacy concerns to some extent. This federated learning process also is not immune to other privacy and adversarial attacks. Hence there is a need to study and mitigate these attacks. In this thesis we explore different types of attacks on federate learning systems in general.

1.1 Main contributions of the thesis

1. A modular network based face verification approach that can counter adversarial attacks on facial images while providing interpretability is proposed.

2. Multiple approaches for face template protection are proposed. Through simulations, we show that the proposed face template protection mechanisms improve the security and performance simultaneously. These approaches reduce intra-class variability in the biometric samples by incorporating deep neural network concepts.

3. Non-invertibility, an important property of a secure biometric systems, of facial biometric data is improved through the use of random projections. The random projections achieve many-to-one mapping while preserving distance between input samples.

4. In this thesis, a lightweight facial feature descriptor called Cross-Local Radon Binary Patterns (XLRBP) is proposed for face recognition. Several studies conducted using the facial feature descriptor on various datasets indicate its superiority in face recognition performance. Also probabilistic LDA is applied to achieve better face recognition performance.
5. Explored various security issues of federated learning and comparison of federated learning and multiparty computation and showcased the trade-off between privacy, computational cost and communication cost.

1.2 Organization of the thesis

The thesis is organized as follows.

In **Chapter 2**, literature survey related to the research work presented in this thesis is provided. In this we reviewed different components related face recognition. We also reviewed various attacks on biometric systems and methods to improve the security of biometric data. We reviewed some of the state of the art biometric template protection methods relevant to the proposed research work.

In **Chapter 3**, we described a face verification method based on modular neural networks [1]. We present modular Siamese network and its applicability to improve robustness of face verification against the adversarial attacks are described. We showcase how the individual facial feature representations (eyes, nose, mouth) can be encoded and used for final face verification task while providing interpretability.

In **Chapter 4**, studies on the proposed template protection mechanism based on deep neural networks is described [2] We show how the feature embedding can be encoded into cancelable binary template preventing invertibility without compromising on the recognition performance. Also, we described another template protection mechanism that is based on adversarial perturbations based helper data [3].

In **Chapter 5**, a method that prevents an attacker to estimate biometric data from feature embeddings is provided [4] We show how Random projections can achieve non-invertibility by projecting the embedding on to a lower a dimensional subspace and also provide cancelability without affecting the matching performance.

In **Chapter 6**, we discuss the importance of federated face recognition for protecting the privacy and the need for preventing attacks on these distributed learning systems [5].

In **Chapter 7**, we describe a novel facial feature descriptor that is based on local binary patterns [6, 7]. We present studies on the proposed cross local binary pattern based facial feature extraction and show that the proposed descriptor is superior to other LBP based methods.

In **Chapter 8**, a summary of the research work done as part of this thesis is provided.

Chapter 2

Review of methods for biometric security

Conventional methods of user authentication, such as passwords and hardware tokens, pose significant security issues, prompting the wide adoption of biometric authentication systems. Uniquely identifying persons using their physiological or behavioral characteristics is known as biometric authentication [14]. Recently, multi-biometric systems [15, 16] are being used for user authentication where more than one biometric characteristic is used for authenticating users. These systems are heavily employed large-scale applications due to their numerous benefits, reducing the inter-class similarity to produce low error rates, and increasing the accuracy and reliability. Additionally, multi-biometric systems are relatively robust against any spoofing attacks because it is very hard to spoof more than one biometric attributes than a single attribute.

This chapter is organized as follows. In Section 2.1, various method of face recognition were discussed. Also, different face detection methods, local binary patterns and the associated face recognition approaches were reviewed. In this section, probabilistic linear discriminant analysis is also described. Various attacks on biometric systems such as presentation attacks, obfuscation attacks have been discussed in Section 2.2. Section 2.4, reviews various template protection methods that exists in the literature. Also, in Section 2.3, effect of adversarial attacks on deep learning models is discussed and finally conclusion is provided in Section 2.5.

2.1 Face recognition

Biometric systems use a set of recognizable and verifiable attributes, which are unique and specific to a person to authenticate them. Face is a non-intrusive and preferred biometric that helps identify people. The process of identifying or verifying a person's identity by their face is known as facial recognition. Face recognition captures, analyzes, and compares patterns based on the person's facial details. A typical block diagram of a face-recognition system is shown in Figure 2.1.

Broadly a face recognition approach involves the following essential steps.

1. **Face Capture**: A face is converted into a collection of digital images (or vectors) during the face capture process based on the subject's facial traits.
2. **Face detection**: The face detection process is an essential step in detecting and locating human faces in images and videos, eliminating non-essential parts of the picture/video
3. **Face registration/enrollment**: A person's identity is pre-registered into the database by computing a template of unique facial features from the captured data.
4. **Face Recognition/Validation**: The face match process verifies if two faces belong to the same person. A person's identity is verified by computing the face template following similar steps of the registration process and then matched against the template(s) in the pre-computed template database.

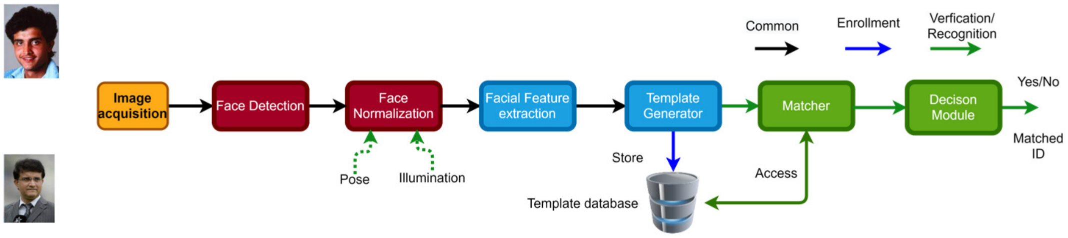


Figure 2.1: Face recognition block diagram

2.1.1 Applications of face recognition

Face recognition can play a vital role in numerous application areas. Some of the application areas include:

- **Security**: Face recognition is used in access control in airports, seaports, ATM machines, buildings and computer/ network security to name a few.
- **Surveillance**: Face recognition can be used to look for known criminals.
- **Services**: General identity verification (i) electoral registration (ii banking, electronic commerce, national IDs, passports, drivers' licenses, employee IDs).
- **Law Enforcement**: Face recognition has a critical and important role in criminal justice systems and forensics.

- **Search and Retrieval**: Face recognition makes it simple to search and retrieve images from vast image databases of licensed drivers, beneficiaries, missing children, and unauthorized immigrants.

- **Video indexing**: Face identification is also used for labeling faces in videos.

In the recent past, many approaches have been proposed to extract complete and high-dimensional local features from images and combining them via machine learning algorithms to deal with high variance and noise present in the images to fulfill a robust face recognition objective.

2.1.2 Face recognition methods

One of the major factors that impact the most face-related computer vision applications is the pose. In a real-world application with a non-cooperating user, the pose could be different each time the picture is captured; hence it is difficult to distinguish and recognize the faces from images with changing poses. Several researchers have proposed many techniques for pose-invariant face recognition. For pose-invariant face recognition, the techniques either extract pose-invariant features for recognition or normalize the face images to frontal pose before extracting any features. However, several issues still exits, including the lack of proper understanding about pose varying sub-spaces in images, pose-robust feature extraction, and complex face synthesis methods, etc.

Projection direction can be discovered by latent space discriminant analysis [17] for various poses so that the same subject projected images in various poses are maximally correlated in the discovered latent space. Multi-task Convolutional Neural Network (CNN) [18] are used for face recognition where person identification is main task and other attributes such as identifying pose, illumination, and expression are the side tasks. For face identification, Deformable Face Net (DFN) is effective against various pose variations. The DFN employs a deformable convolution module that learns identity-preserving feature extraction and face recognition-oriented alignment simultaneously. The intra-class variance is reduced in conventional multi-view subspace techniques, which learn sophisticated nonlinear transformations that project pictures taken in various positions to the same space [19–22]. The definition of similarity varies with the method; for instance, CCA [23] makes them maximally correlated, while PLS [24] maximizes the covariance between them. CCA learns a set of ‘M‘ different projectors from a set of observed content under ‘M‘ different poses such that the projections of different poses of a particular face are maximally correlated in the projected space.

Regression methods use face images or patches to be the basis of the representation scheme by assuming that face/face patches in a pose can be represented as a linear combination of a set of face images or patches. Then the coefficients of linear combinations remain approximately constant across different poses [25, 26]. Similarly, patch-based matching methods for pose invariant face recognition [27] divide face image into patches and extract features as in [28–30], and a representative patch is stored in the gallery for each pose and used as a proxy to match against the patch(s) of the test face during the prediction. Generative models based face recognition methods [31] generate face images of a person across different poses from a common latent variable. At the time of recognition, the images are transformed to the latent space using a pose-specific linear transformation, and perform the recognition task in that space. Pose Invariant Model (PIM) [32] performs both extraction of pose invariant feature extraction and face frontalization for feature extraction. This joint task of pose invariant pose extraction and pose normalization allows them to benefit from each other. The PIM includes a Face Frontalization sub-Net (FFN) and a Discriminative Learning sub-Net (DLN) network to learn the representations. The FFN contains a dual-path unsupervised cross-domain Generative Adversarial Network (GAN) that simultaneously recovers global facial structures and local details.

Shift to deep learning based face descriptors happened after Alexnet [33] reported significant improvement in performance on image recognition tasks. AlexNet showed how a cascade of convolutional neural networks and other layers can effectively do feature extraction and transformation tasks. DeepFace is one the first face recognition models to achieve human level performance. Inspired by the extraordinary success in the ImageNet challenge, the typical CNN architectures, e.g. AlexNet, VGGNet [34], GoogleNet [35], ResNet [36] are introduced and widely used as the baseline models in Face recognition. Several works have focused on developing novel loss functions. The primary goal is to handle noisy data, improve robustness, and pursue generalization goals. In most cases, metric learning tasks, such as face verification, do not involve cross-entropy loss. Loss functions such as contrastive loss and triplet loss, have been proposed in the literature to improve discriminability. Contrastive loss is particularly useful in the face verification task, and is primarily designed based on how they handle the images by comparing with one another and is widely used in Siamese networks. In this, an image pair is fed into the model. If they are similar, the model infers it as '1', otherwise '0'. The goal of triplet loss is to maximize the difference between anchor and negative pairs and anchor and positive pairs. A center loss learns centers for deep features of each identity and used the centers to reduce intra-class variance. FaceNet [37] adopted a triplet loss function based on triplets of roughly aligned matching and non-matching face patches, and is based on GoogleNet [35]. VGGNet [34] is also based on the triplet loss function as used in FaceNet, and is trained on a large number of faces.

Later, several new face models were proposed in the literature which are largely based on angular loss. The large-margin softmax (L-Softmax) loss [38] is also used which tries to maximize the intra-class compactness and inter-class separability among learned features. The main purpose of L-Softmax loss is to learn discriminative features with a large angular margin. Another type of loss uses the angular margin penalty to enforce intra-class compact-

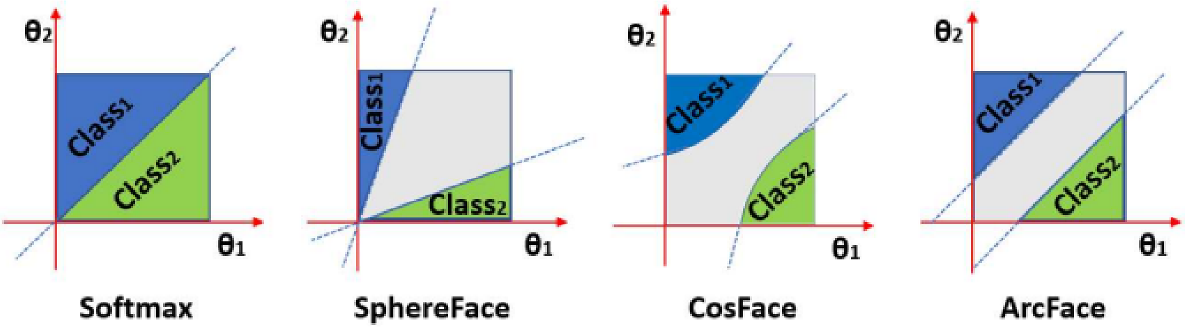


Figure 2.2: Decision margins of different loss functions (source: [38])

ness and inter-class difference of the embeddings on the hypersphere surface. SphereFace [39] uses ResNet [36] architecture and proposed an angular softmax (A-Softmax) [40] loss to extract discriminative face embeddings with angular margin. ArcFace [38] and CosFace [41] introduced an additive angular and cosine margin $cos(\theta + m)$ and $cos(\theta - m)$, respectively. CosFace removes its radial variations by normalizing the L2 norm of the feature, and uses a cosine margin term to maximize the decision distance of different categories further in angular space. ArcFace is based on SphereFace to normalize the feature vector and maximize the classification boundary in the angular space. A comparison on different loss funtions is provided in Figure 2.2 These architectures considered to be having two logical parts. (i) a feature extractor and (ii) a classifier. Once these networks have been trained on large enough databases using an appropriate loss function, layers corresponding to the classification task can be separated to extract feature embedding for a given input face. This enables one to use any classifier on the extracted embeddings and develop applications based on these extracted feature embeddings. In our simulations VGG feature embeddings were extracted by removing the fully connected layers, which are further processed for template protection

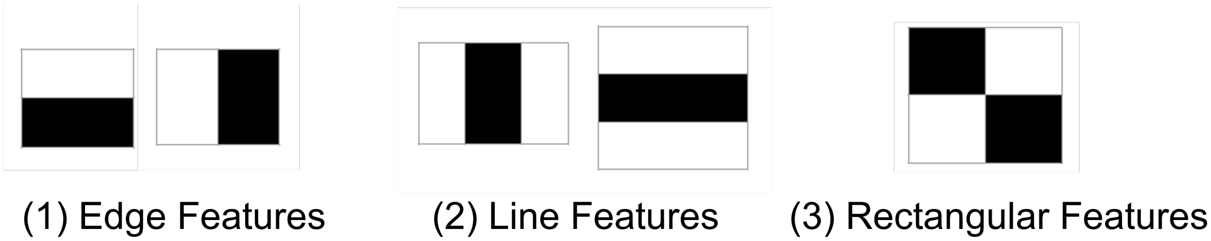
2.1.3 Face detection

Pre-processing for face recognition begins with the detection of the location and orientation of the human face. Geometric transformations are used to create different orientations of the human face by turning the face directly against the camera axis. Then, the pre-processing separates the human face area from regions of the distracting characteristics with the aid of easily recognizable facial elements such as the nose. This process is called segmentation. The majority of early methods for detecting faces were based on classifiers constructed on top of manually created features collected from local image regions, such as Haar Cascades and Histogram of Oriented Gradients (HOG). However, these approaches fail when there are large face variations such as pose and illumination.

Region-based Convolutional Neural Networks(R-CNN) [42] models have had great success with object detection, and some works have also used them to detect faces. A multi-task Region Proposal Network (RPN) is used as the first step to predict candidate face regions and related facial landmarks at the same time. The second stage, which is a convolutional neural network (CNN), then verifies if the candidate regions are valid faces or not. Faster R-CNN [43] has recently demonstrated impressive results on various object detection benchmarks. RetinaFace [44] performs pixel-wise face localization on various scales of faces by taking advantage of joint extrasupervised and self-supervised multi-task learning.

2.1.3.1 Face detection with Haar-Cascade

Viola-Jones (Haar-Cascade) is the most commonly used face detection algorithm [45] and is also used in other object detection tasks. In this, multiple Haar-like features are computed at different scales and positions of the input image, and the Adaboost algorithm is used to select the essential features. A single classifier is trained using each feature (illustrated in Figure 2.3) failed to produce good accuracy, so multiple such classifiers are cascaded to improve the accuracy score. This cascade of classifiers checks the presence or absence of the face region. Adaboost algorithm is used to identify features that belong to a face from all available features. After the face is detected, it can be cropped and stored as an example image for further analysis.


(1) Edge Features

(2) Line Features

(3) Rectangular Features

Figure 2.3: Haar features used for face detection

2.1.3.2 Face detection using HOG and SVM

Like OpenCV, Dlib [46] is a powerful library with wide adoption in the image processing community. Dlib includes two different types of face detection algorithms (i) Histogram of Oriented Gradients and (ii) Max-Margin Object Detection (MMOD); HOG [47] is a powerful descriptor and was initially proposed for human detection tasks. However, HOG can also be used for any object detection task, including face detection. In HOG, the object shape is characterized using the local intensity gradient distribution and edge direction. HOG uses mainly 5 filters during the pre-processing step, which are as follows: (i) Frontal face, (ii) Right profile face, (iii) Left profile, (iv) frontal but rotated right (v) frontal but rotated left. HOG-based face detection has some limitations, i.e., HOG Face detection does not work on faces at odd poses/angles works well with straight and front faces only.

2.1.3.3 Max-margin object detection (MMOD)

Traditional face or object detection algorithms do not make efficient use of the available training data since it trains on only a subset of image windows by reducing the candidate windows of face or objects using Non-maximal suppression. Also, windows partially overlapping an object are a common source of false alarms, and cannot be used in the training set. Max-Margin Object Detection (MMOD) does not perform sub-sampling, but instead optimizes over the entire set of candidate windows. Max-margin approach [48] is used in MMOD that requires a label for each training sample to be correctly predicted with a large margin, and to control the relative importance of achieving high recall and precision.

2.1.3.4 Single shot-multibox detector (SSD)

A deep neural network known as Single Shot-Multibox Detector (SSD) [49] for detecting faces in images. The SSD detector only needs an input image and ground truth boxes for each object during training. It is based on convolution neural networks where they have modified the VGG16 network by adding auxiliary feature layers as a base network. It generates a fixed-size collection of bounding boxes and the corresponding scores for the presence of object class instances in these boxes. It uses non-maximum suppression (NMS) to produce the final object detection output. The SSD model adds several feature layers to the end of a base network, which predicts the offsets to default boxes of different scales and aspect ratios and their associated confidences. The objective loss function is a weighted sum of the localization loss (loc) and the confidence loss, as shown in Equation 2.1.

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)), \tag{2.1}$$

where N is the number of default boxes.

2.1.3.5 Multi-task cascaded convolutional networks (MTCNN)

Multi-task Cascaded Convolutional Networks (MTCNN) [50] is a framework developed as a solution for both face detection and face alignment. The intuition behind MTCNN is that, there exists a strong correlation between face detection and alignment. MTCNN-based face detection and alignment in an unconstrained environment are challenging due to various poses, illuminations, and occlusions. First, the input image is scaled to different sizes to build an image pyramid and then passed on to a three-stage cascade network. In the first stage, it uses a shallow CNN called the Proposal Network (P-Net) (see Figure 2.4) to obtain the candidate windows and their bounding box regression vectors in a similar manner. This stage uses cross-entropy loss. The second stage is called Refine-Network (R-Net)(see Figure 2.4) refines the proposed candidate windows by rejecting a large number of false candidate windows through a more complex CNN and bounding box regression using Euclidean loss between the candidate bounding box and ground truth coordinates. In the third stage, called the output network (O-Net) (see Figure 2.4) it uses a third CNN, more complex than the others, to further refine the result by regressing over Euclidean loss between candidate landmark points and ground truth points. The final output is five facial landmark positions. These landmarks include the left eye, right eye, nose, left mouth corner, and right mouth corner. Examples of MTCNN are provided in Figure 2.5.

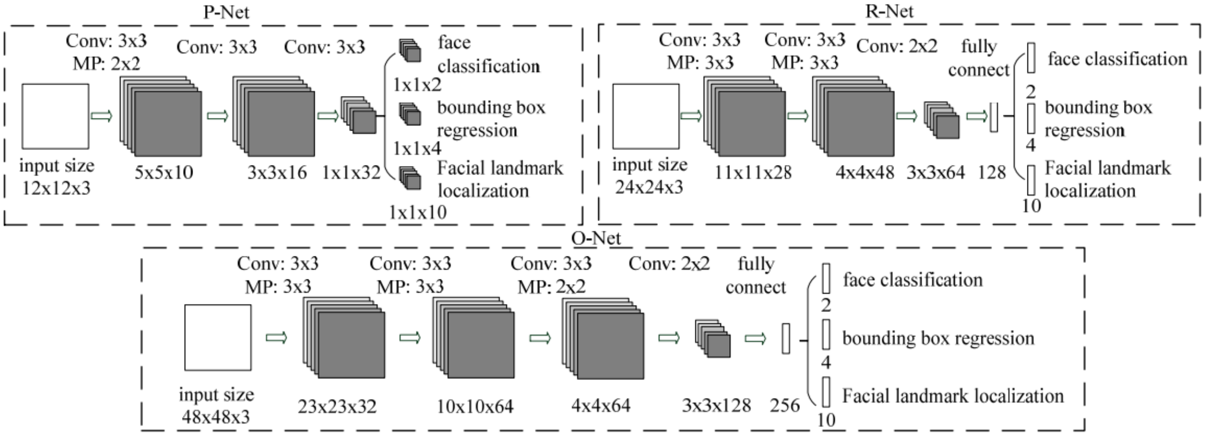
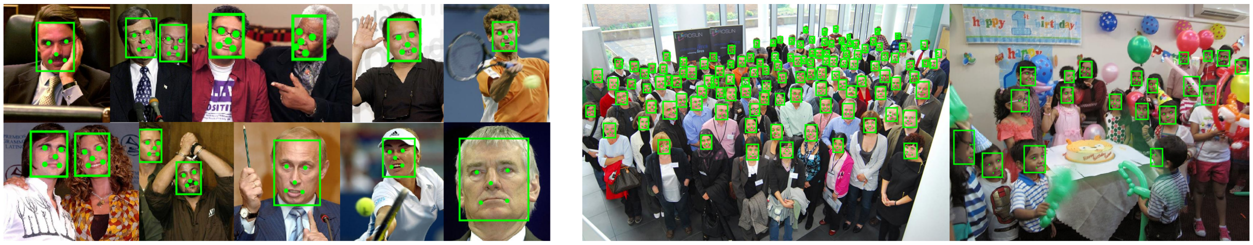


Figure 2.4: MTCNN: Architectures of P-Net, R-Net, and O-Net (source: [50])


(a) Examples of results on FDDB

(b) Examples of results on WIDER FACE

Figure 2.5: MTCNN Performance Results (source: [50])

To measure the accuracy of an object detection algorithm, a commonly used metric is, *Intersection of Union (IoU)*. IOU is used to measure how perfect the prediction bounding boxes match the ground-truth bounding boxes in the dataset. It is calculated as shown in Equation 2.2

$$IoU = \frac{Area\,of\,Overlap}{Area\,of\,Union}. \tag{2.2}$$

IoU ranges between (0, 1) with IoU of '1' indicates better detection. By running on the whole dataset, the average of the IoU of all the samples is reported.

In this research work, Dlib HOG features-based face detectors and MTCNN face detectors were used for the simulations.

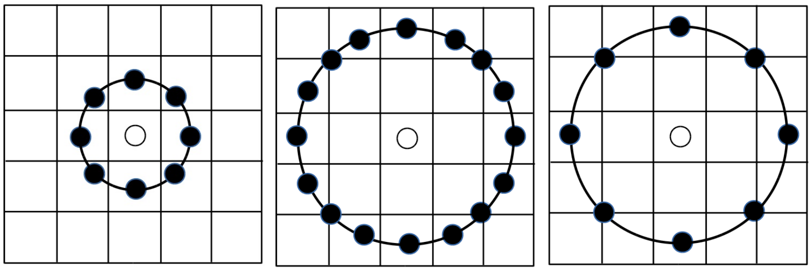
2.1.3.6 Pre-processing

Face recognition is applied to unconstrained scenarios. Raw face data cannot be directly used as the input of feature extraction algorithms because the data contains the human faces with many distracting features such as hair, ear, neck, eyeglasses, and jewelry, and could be different at different instances of time. Also, the face recognition task is challenging due to high variability such as in illumination, scales, pose, and occlusion. While humans can easily relate the faces even with these variations, it is not easy for the machines. Ear and neck features are not reliably identifiable for different head poses. These features could be misleading to the current state-of-the-art, and therefore should be removed before feature extraction. Gamma Correction is a nonlinear gray-level transformation that enhances the local dynamic range of the image in dark or shadowed regions, while compressing it in bright regions and highlights edges. Difference of Gaussian (DOG) another popular pre-preprocessing technique that computes two blurred versions of the input images and calculates their differences. DOG helps in identifying the edge of faces for effective face representations. MTCNN can detect and align faces on the fly for further processing by face recognition algorithms.

2.1.4 Local binary patterns based face recognition methods

Texture representation and analysis is an important area of research for computer vision scientists. Several techniques for discriminating texture patterns exist in the literature. The initial methods are based on signal processing techniques and statistical methods. The recognition stage involves feature extraction, which is critical since a robust set of extracted features will result in a successful classification. Numerous features have been created over the past few decades, but the Local Binary Patterns (LBP) are by far the most effective. Ojala in 1996 [29],proposed the original LBP, in which a 3x3 neighborhood region around each pixel is used to describe the pixels in an image. In this, each of the eight neighborhood pixels is compared or thresholded against the center pixel. If the resulting value is negative, the pixel is set to '0', otherwise, it is set to '1', which when concatenated together to give an

8-bit code corresponding to an integer ranging from 0 to 255. Each binary digit is weighted by its position in the binary pattern to arrive at a decimal value. Therefore, a total of 256 levels can be obtained to represent the relative values around the center pixel within a 3x3 block.



(a) R=1, P=8 (b) R=2, P=16 (c) R=2, P=8

Figure 2.6: LBP Computation with varying R and P

Suppose the number of neighborhood pixels are P and the radius of its surrounding neighborhood is denoted by R, then the notation of the LBP operator can be denoted as LBP_R^P. The LBP operator produces 2^P different output values corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighborhood set. The LBP operator is illustrated in Figure 2.6. A local binary pattern is called uniform if it contains at most two bitwise transitions from ‘0‘ to ‘1‘ or vice versa when the binary string is circular. Following Equation 2.3 calculates the decimal form of a binary string for the pixels.

$$LBP_R^P(I_c) = \sum_{n=0}^{P-1} s(I_n - I_c)2^n. \tag{2.3}$$

I_n and I_c are the values of neighboring and center pixels, respectively. The threshold function is given in Equation 2.4

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2.4}$$

Later several researchers adopted LBP and its extensions for various applications in pattern recognition. We present the discussion of various adaptations of LBP in later chapters.

Uniform LBP: An extension to the original operator can be made by using uniform patterns [51] as shown in Figure 2.7. This uniformity measure of a pattern U (”pattern”) is the number of bit-wise transitions from ‘0‘ to ‘1‘, or vice versa when the bit pattern is considered circular. A local binary pattern is called uniform if its uniformity measure is at most 2. For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform, while the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. Each pixel in an image is labeled with the code of the texture primitive that best matches the local neighborhood. Thus each LBP code can be regarded as a micro-texton. Local primitives detected by the LBP include spots, flat areas, edges, edge ends, curves, and so on.

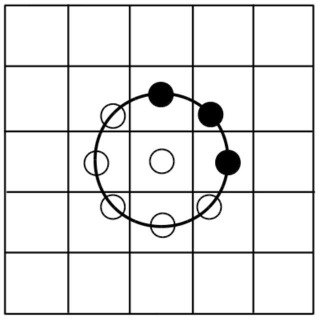


Figure 2.7: Uniform LBP computation

Rotation Invariant LBP: Features that are resistant to input image rotations or invariant to them are desirable in many applications of texture analysis. The rotation of the input image has two effects: each local neighborhood is rotated into a different pixel location, and within each neighborhood, the sampling points on the circle surrounding the center point are rotated into a different orientation. This is because the LBP_R^P patterns are obtained by circularly sampling around the center pixel.

The rotations of a textured input image cause the LBP patterns to translate into a different location and to rotate about their origin. Computing the histogram of LBP codes normalizes for translation, and rotation-invariant mapping leads to rotation normalization. In this mapping, each LBP binary code is circularly rotated into its minimum value as shown in Figure 2.8.

$$LBP_{P,R}^{r,i} = min_i \; ROR(LBP_{P,R}^{i}) \tag{2.5}$$

LBP descriptors originally developed for texture classification [29], have been proved to be effective for face recognition [52–54]. Numerous variations have been suggested due to their simplicity and effectiveness, concentrating on different configurations such as pixel neighborhood topology, thresholding and quantification, encoding, and grouping complementary features. Researchers identified different LBP operators to deal with different tasks. In this thesis, we present an LBP-based feature representation that is invariant to pose and rotation called Cross-Local Binary Patterns (XLBP).

In LBP based approach, the face image is initially divided into small regions. Local binary pattern features are extracted from each of these regions and labeled, and histograms

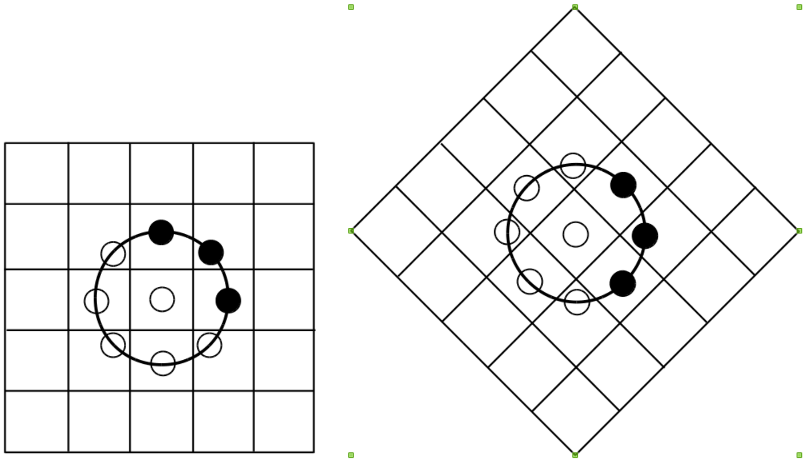


Figure 2.8: Uniform LBP computation on a rotated image

of all labeled regions are computed and concatenated to construct a feature histogram. This histogram can represent a face image efficiently. LBP encodes the texture of the facial regions, and the whole shape is recovered by the construction of concatenated histogram [52, 53, 55]. In Radon transform-based approach, all training images are transformed to Radon space and applied linear discriminant analysis to compute feature vector set [56]. A new image matching is performed using a K-NN classifier using $l1$ norm and Mahalanobis distance measure. Other methods for face recognition are proposed in the literature based on LBP and Gabor Transform, for example, local Gabor binary pattern (LGBP) [53], and histogram of Gabor phase pattern (HGPP).

2.1.5 Probabilistic linear discriminant analysis (PLDA)

Many of the conventional face representation methods use the distance-based approach in which the probe and gallery images are linearly projected to a lower-dimensional plane to estimate the feature vectors, and a match is carried out using distances between these feature vectors. Some of these feature-based methods are, e.g., Local Binary Patterns (LBP) [55], Local Phase Quantization (LPQ) [57, 58], Dual-Cross Patterns (DCP) [59], Binarized Statistical Image Features (BSIF) [60], in which face image is represented using patterns that can discriminate persons efficiently. Dimensionality reduction algorithms such as Principal Component Analysis (PCA) [61], Linear Discriminant Analysis (LDA) [62], and Independent Component Analysis (ICA) [63] reduce the dimension of the feature vector by considering the statistical properties (like mean, standard deviation, etc.) of the gallery image feature vectors. These methods project the feature vectors into lower dimensional space, while preserving the characteristics of the feature vector. Probabilistic Linear Discriminant Analysis [64] is a probabilistic version of Linear Discriminant Analysis (LDA) (which is a linear dimensionality reduction method) with abilities to handle more complexity in data. While PCA identifies the linear subspace in which most of the data's energy is concentrated, LDA identifies the subspace in which the data between different classes is most spread out, relative to the spread within each class. Hence, LDA can be used for classification. PLDA has been used for feature extraction specific to a given class, recognition, verification, and the generation of similarity scores for clustering. The PLDA generative model operates on the assumption that samples of input data are drawn from a distribution, frequently a Gaussian distribution. We must find the parameters of the model which best describes the training data. Simon et.al. [65] proposes a generative model that creates a one-to-many mapping from an idealized *identity* space to the observed data space.

$$x_{ij} = g(\mathbf{h}_i, \theta) + \epsilon_{ij}, \tag{2.6}$$

where x_{ij} is modelled as being generated by function $g(.)$ along with additive noise. In this pose-dependent vectors are generated from an underlying pose-invariant representation and estimate the invariant vector for a given individual using the inverse process using the identity at a given pose. [64] extends [65] by adding additional factors representing the mean of the dataset and pose variations. In this thesis, [64] is extended to represent identities based on their local binary patterns and Gabor filters.

Gabor filters [66] are used as pre-filters to extract the shape information from the face images, e.g., Local Gabor Binary Pattern (LGBP) convolved with multi-scale and multi-oriented Gabor filters to extract shape in different orientations and scales, and later applied to pattern extraction methods to represent the face image. This representation is robust to translations, small pose variations, and illumination changes. In these methods, if the feature vector is large; then, we must use dimensionality reduction algorithms for further processing

2.2 Attacks on biometric authentication systems

Attacks at various stages on a biometric authentication system are shown in Figure 2.9. Attackers target either the functioning of the modules or the intermediate data, with the objective of impersonation and obfuscation [67]. The attacker might interfere with the function of the modules in biometric systems to force them to produce a target output. Additionally, the attacker may intercept the intermediate data and try to reconstruct the biometric features from it or use this intermediate data to replay during the verification stage, bypassing other modules to obtain unauthorized access. These attacks can be broadly classified into (i) Impersonation attacks and, (ii) Obfuscation attacks. Vulnerabilities of an unsecured biometric system include,

1. Present fake biometrics at the sensor

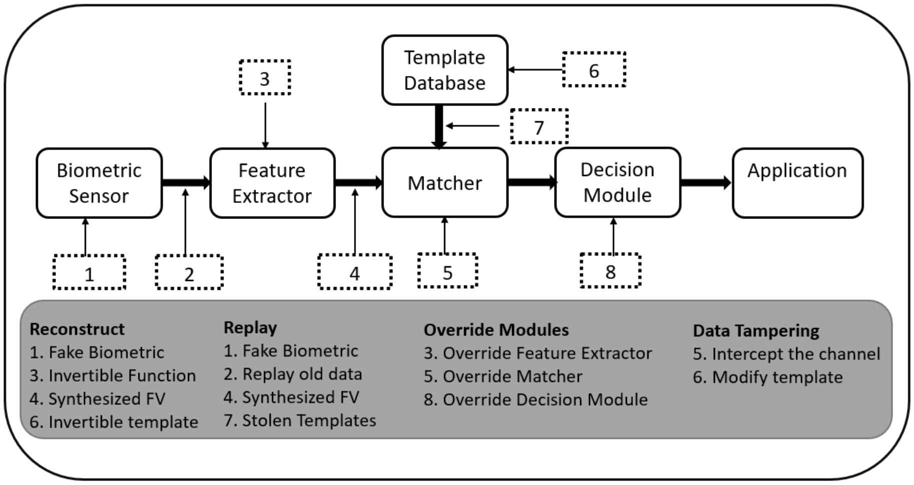


Figure 2.9: Attacks on a generic biometric authentication system (source: [67])

2. Replay recorded biometric to the system bypassing the sensor
3. Reconstruction of biometric from the feature set
4. Produce feature set pre-selected by the intruder
5. Insert synthesized feature set for template computation
6. Reconstruct feature set from the template
7. Modify or replace the templates in the database
8. Insertion and modification of the templates
9. Forced to produce pre-selected match score by corrupted matcher
10. Hacker overriding the final match decision

2.2.1 Presentation attacks

Presentation attacks also known as Impersonation, Replay or Synthesis attacks, affect the security and privacy of the users as an attacker can gain access by pretending to be the actual user. Impersonation or zero-effort imposter is a spoofing technique that requires no assistance from electronic devices. Impersonation is carried out by mimicking a person's biometric (such as voice and face). In impersonation attacks, the attacker tries may claim the identity of a victim by using his biometric during verification as shown in Figure 2.10, and Figure 2.11. Impersonation attacks mainly exploit stolen templates and algorithmic and software vulnerabilities. The replay attack is the most popular type of spoofing attack as it is the simplest to conduct. Biometric data can be stolen through social media, sensors, or transmission. The transmission attack can be avoided and deflected by having a secure transmission protocol and assistance from security software. By comparison, it is much more difficult to defend against sensor-level attacks and requires considerable attention due to the ease of obtaining the biometric data. Synthesis attacks are mostly conducted by professional adversaries and require the knowledge about biometrics. Common synthesis attacks include deep fakes, speech synthesis, and voice conversion.

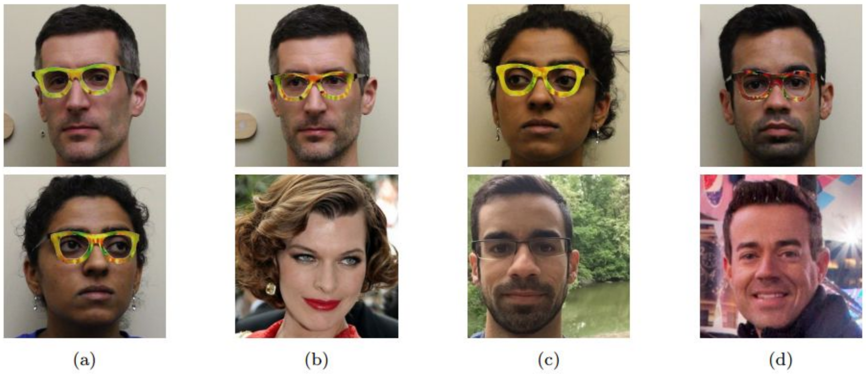


Figure 2.10: Examples of successful impersonation and dodging attacks (source: [68])

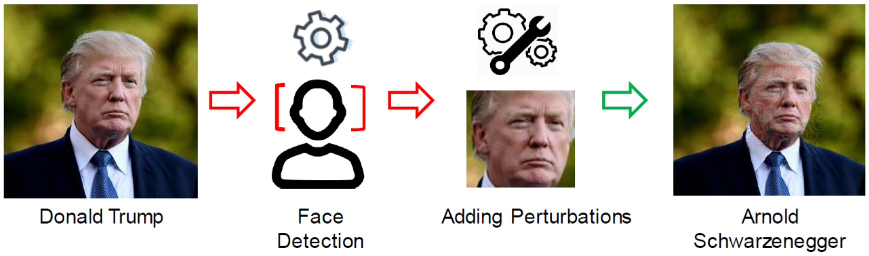


Figure 2.11: Adversarial perturbations for impersonation

When the attacker gains unauthorized access to the secure database in which templates are stored, the following things can happen (i) Template replacement - the biometric template of a victim is replaced with the adversary's template to gain unauthorized access. (ii) Cross-application Matching - stolen template of the victim can be used to impersonate other biometric systems. In some cases, an attacker might gain physical access to the sensor to extract the biometric data of the victim and gain access to the biometric feature representations of the victim, which can be used to produce synthetic or fake biometrics to impersonate. These types of attacks are called spoofing attacks. When the biometric system suffers from a high false match rate (FMR) and false non-match rate (FNMR), an attacker may repeatedly attempt to access a system, adjusting a biometric feature until a sufficiently close match is obtained [69]. Vulnerabilities in the design of the software modules of biometric systems enable the attacker to bypass or override a software module to produce the desired output for maliciously gaining access. The attacker may directly present the stolen template to the matching module or override the matching module to produce an incorrect matching result. These types of attacks are called replay attacks. Reconstruction attacks on neural network based models are called model inversion attacks in which an adversary attempts to recover the private dataset is used to train a supervised neural network. The main goal of the adversary is to generate realistic and diverse samples that can describe each class in a private dataset. In the problem of model inversion (MI) attack, the attacker has access to a “target classifier”. Authors of [70, 71] successfully recovered faces of the users used in the training using model inversion attacks. Some examples of model inversion attacks are given in Figure 2.12.

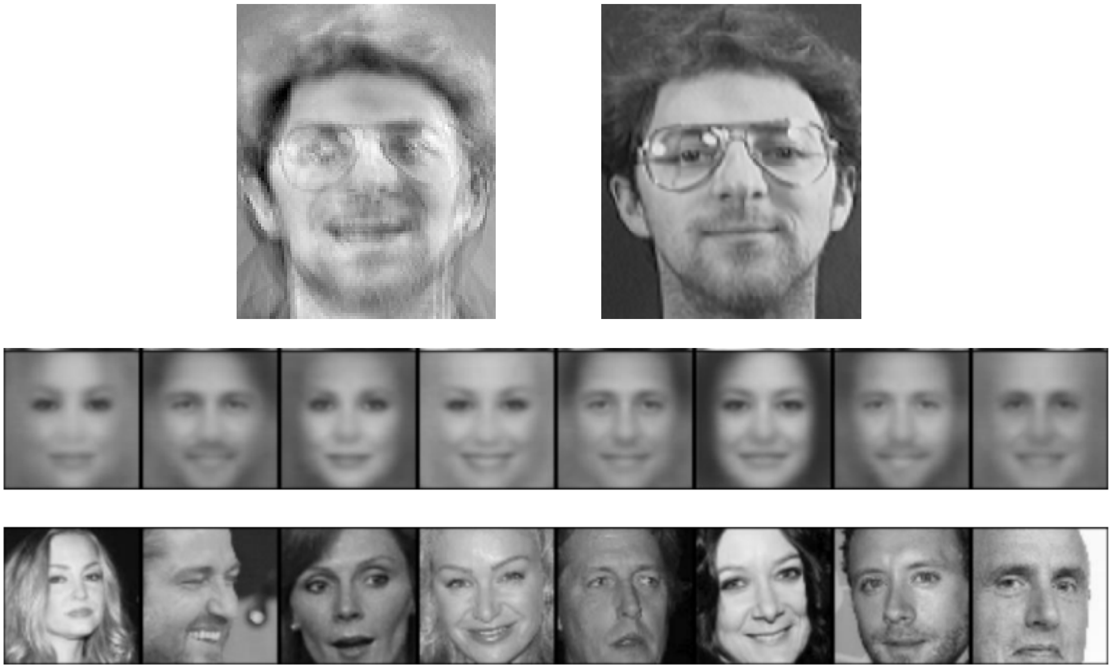


Figure 2.12: Face reconstruction from model inversion attack [70]

To protect the biometric data from leaks, face verification models can be trained on-device using *Federated Learning*, enabling only the model updates to be sent to the service provider instead of actual raw biometric data. In federated learning a central aggregator, upon receiving model updates from various clients, aggregates to build a complete model. However, federated learning is also not immune to attacks. These attacks can happen at various stages of the federated learning process and include (i) poisoning of training data and (ii) data reconstruction from the model updates.

Apart from the above attacks, biometrics data needs to be protected from insider attacks, where a privileged insider can access either the raw biometric data or templates of one or more users of the system. Similarly, when the biometric data get regularly updated to account for the variations in biometrics due to age and other conditions, an attacker could introduce poisoning samples into the training data to be able to manipulate the biometric system in the future.

2.2.2 Obfuscation attacks

Obfuscation attacks on biometric systems have increased significantly in recent times. The goal of the adversary in such attacks is to impersonate an individual to gain unauthorized access to sensitive resources. The biometric obfuscation is achieved by falsifying or masking the biometric data, before or after the enrollment of the biometric attributes by the system, which might prevent the system from recognizing an individual.

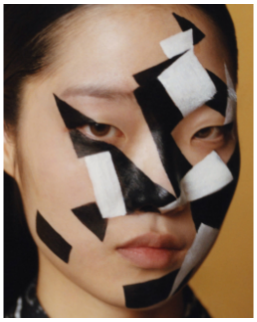


Figure 2.13: Face paint for obfuscation

Obfuscation attacks, are used mainly for evading the recognition of the identity. The obfuscation is sometimes done by removing some details from the biometric data to reduce key features. Blurring of faces, tattoos, and paintings on faces are some examples of obfuscation attacks on face recognition systems, see Figure 2.13. Physical alteration of one's own biometric data can happen either by deterioration (fingerprints) or by surgery (face). Note that most people who perform this type of attack are usually on checklists, and most are wanted by law enforcement.

Designers of biometric systems must consider these attacks and provide several security measures to protect sensitive personal data and equipment.

2.2.3 Cancelable templates

A secure biometric system shall not only accurately authenticate the user and deny access to imposters, but it should also store the templates in a secure manner. Biometric data (raw biometric data, template) is permanently associated with the user and cannot be changed or replaced, unlike passwords. Cancelable templates refer to the process that can revoke a template when compromised and reissue a new template and new helper data such as a cryptographic key. Also, when used in other applications, a different set of keys are used to generate templates for the same user so that there is no correlation between the templates in the two biometric systems. A typical biometric template protection mechanism is shown in Figure 2.14. An ideal biometric template protection scheme is expected to meet the following requirements [72–75]:

1. **Security:** It should be computationally hard to reconstruct the original biometric template from the protected template (non-invertibility).

2. **Performance:** Any biometric template protection mechanism shall not degrade the recognition performance of the biometric system

3. **Diversity:** Different protected biometric templates can be generated based on the same biometric data of an individual for use in different applications. These different protected templates should not allow cross-matching across applications.

4. **Revocability/Cancellability:** It should be easy to revoke the compromised biometric template and generate a new protected biometric template based on the user's biometric data.

A major challenge in a biometric template protection methods satisfying the above requirements is, high intra-user variability in the biometric data and low inter-user variability. High intra-user variability results in high FRR (False Rejection Rate) (due to variations in pose, illumination, expression, etc.), whereas low inter-user variability leads to high FAR (False Acceptance Rate). Many biometric template protection algorithms are reported in the literature to address the above problems, but most have a trade off between template security and matching performance.

2.3 Adversarial attacks on face recognition

Deep learning-based face recognition has surpassed handcrafted feature-based systems and shallow learning systems in performance. In [34], the authors proposed a deep learning architecture called VGGFace for generating facial feature representations or face embeddings. These face embeddings can be further used for identifying the person using a similarity measure or a classifier. DeepID2 [76] uses a Bayesian learning framework for learning metrics for face recognition. In FaceNet [37] authors proposed a compact embedding learned directly from images using triplet-loss for face verification. Different loss functions that maximize the intra-class similarity and enhance discriminability for faces proposed in ArcFace [77],

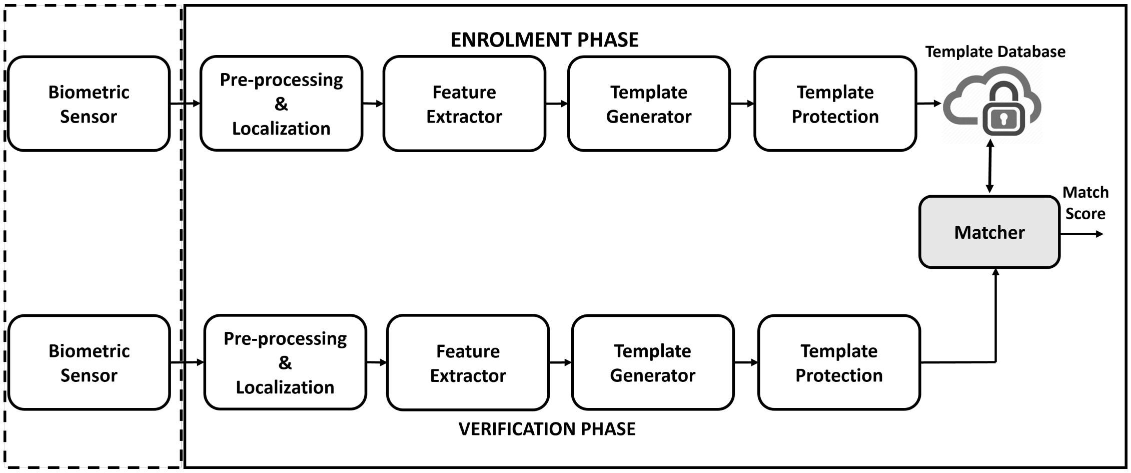


Figure 2.14: A typical template protection mechanism

CosFace [41], SphereFace [78], CoCoLoss [79]. While significant improvement in performance is achieved through better generalization, attackers try to exploit some of the vulnerabilities in the learning process.

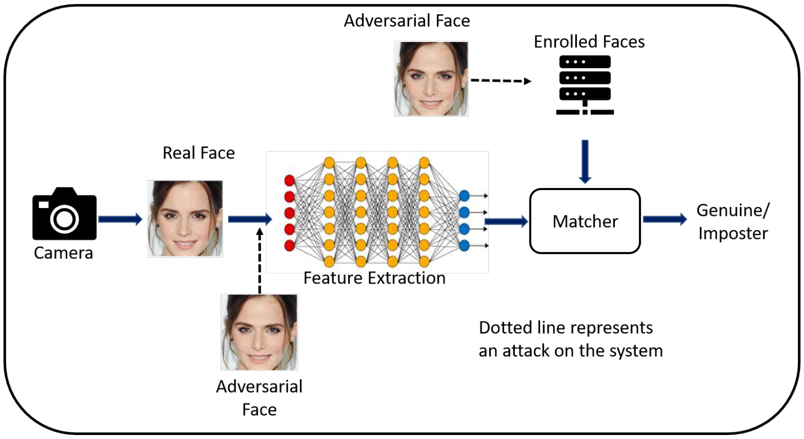


Figure 2.15: Adversarial attacks on a typical face recognition system
s

Adversarial attacks on deep learning models are one of the major sources of attacks, and it is easy to generate adversarial examples. Adversarial examples are created by carefully crafting perturbations to non-robust features that are learned by the deep networks through training. An attacker, through adversarial examples, tries either to fool the deep-learning based face verification models or to achieve targeted impersonation, as shown in Figure 2.15. A method to realize adversarial attacks by introducing a pair of eyeglasses is provided in [68]. These glasses could evade detection or be used to impersonate others. Another approach for fooling ArcFace using adversarial patches is proposed in [80]. In [81], the authors have proposed an approach for detecting adversarial attacks on the face. Researchers developed defenses against this kind of attack, but these defenses are continuously broken, leading to a race between the attackers and security experts. Recently, certified robustness was proposed by [82] and [83] provides bounds on the robustness of deep learning models.

2.4 Review of biometric template protection methods

The need for biometric template protection schemes that could maintain the overall system's security while delivering high recognition performance got the attention due to the security concerns and privacy requirements associated with a biometric authentication system. This could be a consequence of user expectations or the government regulations. Simple and naive cryptographic approaches, such as encryption and hashing are used to protect the biometric templates. For encryption, an additional key or a password is required, which is a security overhead. Later, the encrypted template must be decrypted during the authentication phase, which might leak some information about the original biometric template. (i) A system's generated key or a user provided password or a key used for encryption of the biometric template is not sufficient for the system's security. (ii) In the case of loss of the underlying key/password or the protected template itself, re-enrollment of biometric templates is required, which is not easy in the case of biometrics, unlike passwords. (iii) A biometric-generated key could help enhance the security of biometric systems. Hashing is another way of securing the biometric templates. In the hashing approach, any two samples of the same biometric instance are never the same. Thus, bit errors always exist in the biometric templates, which makes one-way hashing in biometrics infeasible. Recently, biometric template protection schemes have been introduced. These are broadly categorized as biometric cryptosystems or bio-cryptosystems, cancelable biometrics, and Homomorphic encryption schemes. Error-tolerant cryptographic algorithms are useful in many circumstances in which security depends on human factors.

2.4.1 Biometric cryptosystems:

Bio-cryptosystems (refer Figure 2.16) use cryptosystem based security for template protection, thus offering high security. Popular biometric cryptosystem based approaches such as *fuzzy commitment scheme* [84, 85] and *fuzzy vault scheme* [86, 87] output an encrypted template thus offering high security.

2.4.1.1 Fuzzy commitment scheme:

In a fuzzy commitment scheme [84, 85], given a witness x, a fuzzy commitment function F is computed to conceal 'c' using a conventional hash function h. A biometric template,

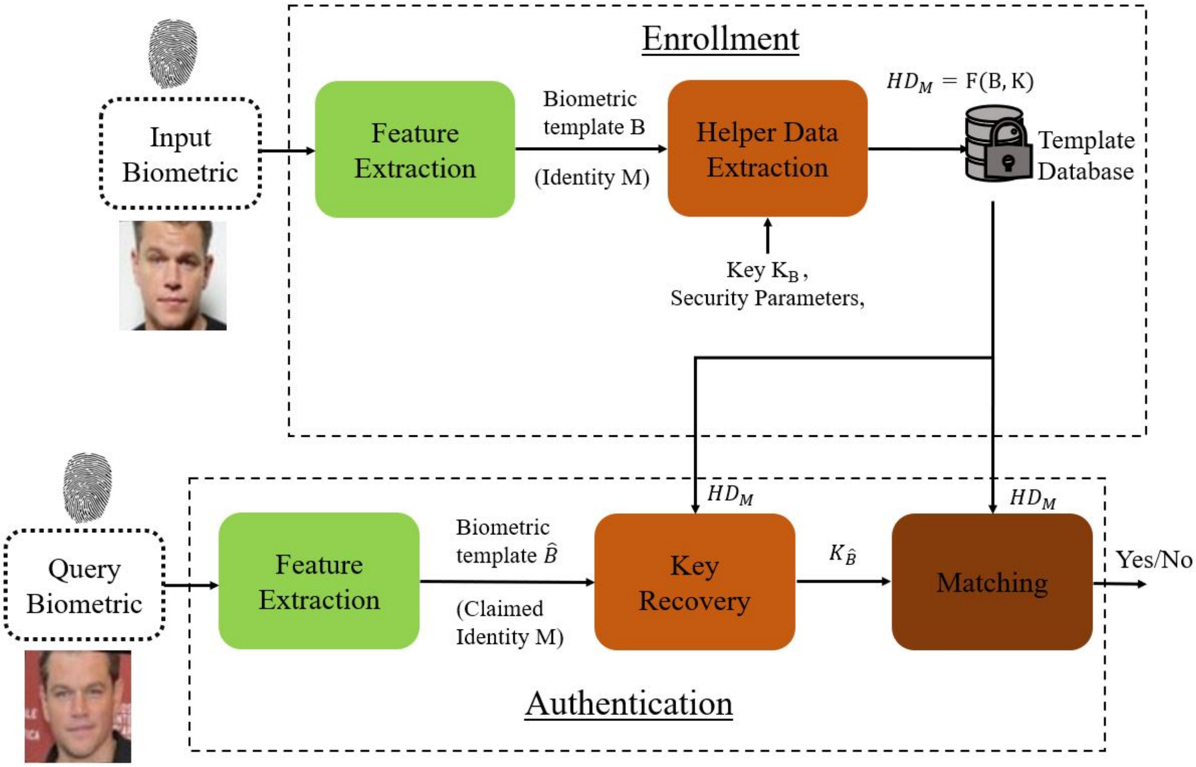


Figure 2.16: Block diagram of a typical bio-cryptosystem; HD indicates helper data, K'= K if same biometric is presented during verification time

such as a fingerprint is typically represented as x. The code word c represents a secret key protected under this template. For example, c might be a decryption key protected under the user's fingerprint x as the commitment *F(c, x)* . Here, *F(c, x)* is stored on a server as a commitment, and c is drawn from a large enough space C to ensure that *F(c, x)* does not reveal x. To unlock and reveal this key, it suffices for the user to present a corrupted fingerprint image x sufficiently close to x to successfully de-commit *F(c, x)* .

In helper data system (HDS) based biometric systems, during enrollment, biometric features such as facial fiducial points are used to bind a cryptographic key, creating one of the helper data. The operation involved in the binary XOR. Here, the system's objective is to disqualify an unauthorized subject who lacks the original face features utilized during enrollment during the verification procedure. A real subject with the appropriate face features will be accepted. The verification method must, more crucially, be exclusively based on the helper data and not require direct access to the original facial features. In the key binding process of the fuzzy commitment scheme, the cryptographic key is encoded for error tolerance initially. Then it gets bound to a binarized feature vector. This process is similar to storing and locking the cryptographic key in a 'secure box,' where the 'key' to this box is the biometric feature vector itself. Without access to the correct physiological features, the cryptographic key cannot be recovered. In the key unbinding process, a user claiming some identity submits his or her facial features for verification, and a binarized feature vector is extracted from them. If the two feature vectors, during enrollment and during verification, match exactly, then the original cryptographic key is unbound successfully. However, in a practical scenario, there are typically differences in the two vectors requiring the need for error correction codes.

2.4.1.2 Fuzzy vault scheme

Fuzzy vault scheme [86] is a cryptographic construction whereby an user 'Alice' can lock her biometric B_A using the set $\boldsymbol{A}$, yielding a vault denoted by V_A. If Bob tries unlocking the vault V_A using his own set $\boldsymbol{B}$, he will succeed provided that $\boldsymbol{B}$ overlaps largely with $\boldsymbol{A}$. Anyone who tries to unlock V_A with a biometric differing substantially from Alice's will fail, ensuring that Alice's biometric remains private. Thus, a fuzzy vault may be thought of as a form of error-tolerant encryption operation where keys consist of sets. A fuzzy vault may be thought of as a form of error-tolerant encryption operation where keys consist of sets.

Biometric cryptosystems use error-correcting coding techniques to deal with intra-class variations but fail to handle large intra-class variations, thus leading to low performance. To overcome this limitation, deep CNN is used to minimize intra-class variations and maximize inter-class variations.

Transform-based approaches, transform the original template into a new domain for generating templates. This transformation can be achieved using a non-invertible transform and salting. However, these approaches have a trade-off between performance and security. Ratha et al. [88] provided three non-invertible transforms, namely Cartesian, polar and functional, for generating cancelable face and fingerprint templates. They achieve high template security, but the matching performance is low.

2.4.2 Hybrid schemes

Hybrid approaches combine the biometric cryptosystem and transform-based approaches. Feng et al. [72] proposed a hybrid approach for generating secure face templates. The proposed approach extracts the face template through a feature vector extractor. Random projections is used on the extracted face template to project the original template into a subspace, generating a cancelable template. Discriminability preserving transform is applied to the cancelable template to enhance the discriminability and convert the real-valued template into a binary template. Finally, the fuzzy commitment scheme is used to protect the binary face template.

Pandey et al. [89] extract features from the set of chosen local regions of the face using the histograms of gradients (HoG) and local binary patterns (LBP). Quantization is done after each local region's features have been extracted, then cryptographic hashing (SHA-256) is done. The collection of hashed local features collected from the face made up the transformed face template. However, the feature space being hashed was not distributed equally, and the proposed technique had poor matching accuracy. To overcome the shortcomings of the algorithm, authors of [90] provide another secure face template protection algorithm. The algorithm employs Convolutional Neural Network (CNN) to train a mapping from facial images to the binary codes, assigning each user a distinct maximum entropy binary code that is bit-wise randomly generated. Each user's binary code is hashed using a cryptographic hash function (SHA-512). The cryptographic hash of the user-assigned binary code is therefore the transformed face template.

2.4.3 Data augmentation

Data augmentation is used to synthetically generate more data samples representing the original real distribution of the data. In face recognition more face images (from the existing images) are generated synthetically per user. The number of training face images per user are limited and deep neural networks require a large number of training images to achieve good performance. This is important especially for biometric data, as the data are scarce. One-shot enrollment of users where strictly one face image of the user is used for enrollment. Hence data augmentation is very important. Data augmentation is performed on each face image (as in [2]), using Keras image data generator. This includes operations such as horizontal flip, re-scaling, zoom, changing the shear angle, and rotation to generate five augmented/synthetic face images per identity image from a single face image. For each image of size $m \times m$, we extract all possible crops of size nxn. This gives a total of $(m-n+1) \times (mn+1)$ crops. Each cropped image is resized back to the original size $m \times m$. This data augmentation process results in a total of 6*(m-n+1) *(m-n+1) images for each input face image.

2.5 Summary and conclusions

The COVID-19 pandemic has really sped up the digital transformation and the associate technologies due to the social distancing norms. This has led to many people coming online and accessing services raising the importance of security and authentication. Biometrics is one such authentication mechanism in which users not required to memorize complex passwords and type them. Face as a biometric is easy to use, does not require users' co-operation and a non-technical person can validate the match. However, these biometric systems have several issues with respect to their security and privacy. Also, several regulations mandate the organization to store the biometric data securely. Even with a high level of security, we see several data security breaches. Attacks on biometrics can happen at different stages of the biometrics system. As biometrics can not be replaced like passwords or tokens, it is essential to find ways to mitigate the effects even if the biometric data servers are compromised. Through template protection, one can protect biometric data by generating cancelable templates. This allows the service provider generating multiple virtual (cancelable) templates when they are compromised. We briefly reviewed different biometric security methods. We discussed Biocrypto-systems that enable us to generate cancelable templates. Transform-based approaches proved to be performing better. Hybrid approaches that combine Biocrypto-systems and transform-based approaches changed the direction of template protection research. Model explainability allows us to understand the predictions well, and hence help in taking actions that improve the security. We also discussed some stat-of-the-art-face detection techniques such as HOG based face detection and MTCNN face detection techniques. We also described adversarial attacks and their impact on recognition and security. Poisoning attacks in a federated learning setting are also a cause of concern and need to be addressed. Techniques used for protecting biometric data depend on the type of descriptors/models and need to be custom built based on the security requirements. We also discussed LBP as a facial feature descriptor, which is extended and described in the following chapters.

Chapter 3

Face data protection

Over the past decade, many deep learning methods for face verification have been implemented, some of which have even outperformed humans. These deep learning methods, while enabling exceptional performance, do not explain how they arrived at their predictions. Relying solely on the outcomes of these black boxes without interpreting the causes of their judgments could be damaging, particularly in medical, financial, and security domain applications. The chapter is organized as follows. Section 3.1 describes the need for robust and interpretable face verification. In Section 3.2, modular Siamese network (MSN) is described and how this can be used to improve the robustness and interpretability and the methodology for training and verification. Section 3.3 describes the experimental results of MSN based face verification. Discussions of the experimental results provided in Section 3.4. In Section 3.5, a feature based face retrieval method based on the MSN and locality sensitive hashing (LSH) method for finding projections that handle intra-class variations. The process for retrieving the face based on a query image is explained in this section. Section 3.6 provides the conclusion of the chapter.

3.1 Robust and interpretable face verification

Explainable face recognition (XFR) is the problem of explaining why a face verification system matches a face. The explainable face recognition can be grouped into two categories: one refers to the post-hoc procedure relying on elaborate perturbation on pipelines, then visualizing the impacts served as the basis for explainable insights, and the other category is leading the representation learning to be explainable during the training stage. Various interpretability methods have been proposed in the literature for image recognition tasks, and are also relevant to face recognition. These methods include GradCAM [91], GRADCAM++ [92], LRP [93], Integrated Gradients [94], SHAP [95], LIME [96] etc. These methods, in general, highlight parts of the objects that contribute heavily to the final prediction of the models. Literature on explaining the face recognition (XFR) predictions is very limited.

Jiang et al. [97] considered both local (component-based representation), and global feature maps to compute feature representations using self-attention. Randomized Input Sampling for Explanation (RISE) [98] is used to construct a saliency map associated with a particular class by randomly perturbing the input image by masking selected pixels, and evaluating it using a blackbox system. However, the current Explainable Face Recognition (XFR) methods are difficult to balance the explainability, and the recognition performance. Deep learning models' vulnerability to adversarial attacks is another serious issue which needs attention. Trivial noise that is undetectable to the human eye and can deceive Deep learning models. Different black box, and white box adversarial attack methods were proposed in the literature [99–101]. The problem of detecting, and defending the adversarial attacks on deep learning models is still largely unsolved. Particularly, these attacks on face verification systems pose a serious security threat.

Despite the existence of several post-hoc interpretability methods, it is desirable to have a system that is inherently capable of producing interpretation of its decisions. It is important to present the explanation in a manner that can be easily understood by the end-users, instead of heatmaps and quantitative parameters. For example, if the heatmaps can explain the logical components of the face regions, it will be easier for a non-technical person to understand, and validate the predictions.

The biological modularity of the human brain served as the inspiration for the class of composite neural networks known as modular neural networks (MNN) [102]. MNNs are inherently simpler to interpret than monolithic neural networks because of their architecture, and divide-and-conquer strategy. MNNs also intrinsically introduce structural interpretability due to their modular structure. Some of the benefits of MNNs include *Efficiency, Robustness, and Independent training*. In this thesis, we propose a face verification approach [1] based on Modular Siamese Networks(MSN) as shown in Figure 3.1 that addresses both the interpretability, and the robustness of adversarial attacks by learning independent latent representations of high-level facial features. The suggested method creates understandable heatmaps on the fly, and is demonstrated to be far more resistant to adversarial samples.

3.2 Modular Siamese network

A Modular neural network (MNN) [102] is made up of independent neural networks that are linked together to achieve the desired task. Each module learns a part of the specific task, and is inherently more interpretable than monolithic neural networks. Studies have shown that MNNs are better at handling noise than monolithic networks. In a MNN based face recognition system the face is composed of several individual features such as eyes, nose, and mouth. These individual facial features can be used to identify people to a certain extent. In the proposed MNN based face verification approach for interpretability, we allocate dedicated

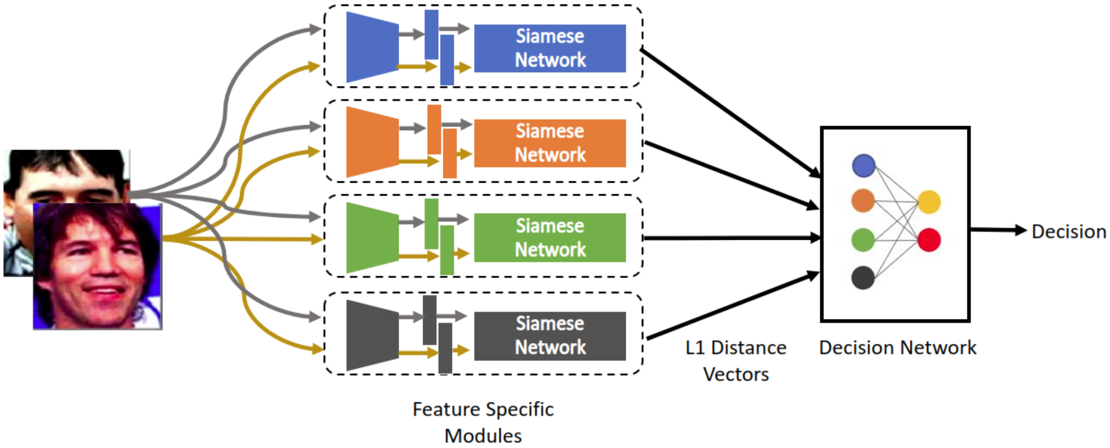


Figure 3.1: Proposed Modular Siamese Network [1]. Feature-specific encoders first de-entangle the image to produce feature-wise embedding pairs, which are then fed to Siamese networks to calculate the distance vectors. The decision network is then provided with the concatenated set of distance vectors for a final decision.

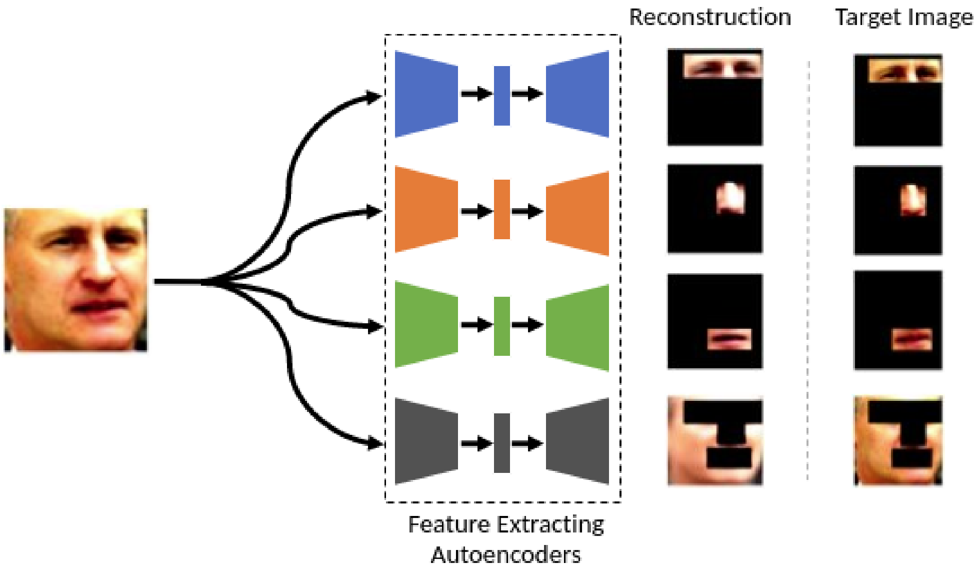


Figure 3.2: Proposed modular Siamese network [1] Pre-trained VGGFace network is used to obtain face embedding; these embeddings are then fed to individual modules which will encode them to feature specific latent representations. These latent representations are fed to the decision network for verification.

neural network modules for the eyes, nose, mouth, and one module for the rest of the features as shown in Figure 3.2. These neural network modules could be any module that can effectively compute the feature embedding for the given task. To develop independent and distinct latent representations for various facial features such as eyes, nose, and mouth, we used autoencoders. To learn latent facial part representation, we mask the input face image to retain only the region of interest (eyes, nose, mouth), and present the ROIs as the target image for this autoencoder. These autoencoder modules are trained to distinguish between face identities based on one facial feature. These encoded facial component embeddings were further processed for overall face identity. We used under-complete autoencoders [103] in this work, a type of autoencoder whose latent dimension is smaller than the input dimension. To ensure that only the most important features are maintained in the encoded latent vectors, under complete autoencoders are trained to reconstruct the original image as accurately as possible while limiting the latent space to a small enough dimension. To reconstruct only the necessary portion of the image, the autoencoder first learns a latent representation, including crucial information about the feature. Once these autoencoders are trained, we retain the encoder, and substitute each decoder with a Siamese network which results in Modular Siamese Networks (MSN) as shown in Figure 3.1. A pair of images, either a legitimate pair or an impostor pair, is provided as an input for the face verification task. In the proposed MSN architecture, both the images are fed to a common pre-trained feature extractor to obtain latent face embeddings. These embeddings are then fed to different modules in MSN. Each module has a feature-specific encoder that produces latent representations of the region of interest.

Learning complex features is computationally expensive, and when the number of samples available for training a neural network is limited, these learned features may not be representative of the underlying data distribution. In *one-shot learning* highly discriminative features are computed using only one sample for making predictions. Systems that incorporate one-shot learning tend to excel in similar instances but fail to offer robust solutions. *Siamese Networks* [34, 104] is a framework that trains a model to discriminate between a collection of the same class or a different class of features. Siamese Networks aim to first learn two mirror neural networks having the same parameters that can discriminate between the class-identity of image pairs, which is the standard verification task for image recognition. This model learns to identify input pairs according to the probability that they belong to the same class or different classes. Once learned, this model can be used to evaluate new images in a pairwise manner. The pair with the highest prediction score is declared to be of the same class, and the class is assigned to the test image. The weights, and parameters are the same between the two Siamese twin networks. The underlying assumption of this architecture is that if the inputs $x1$ and $x2$ are similar, then the distance between the output vectors $h1$ and $h2$ will also be closer. The network has been trained to maximize the distance between unmatched pairs, and minimize it between matching pairs. To accomplish this objective, loss functions such as triplet loss [105] and contrastive loss [106] can be used. A few improvised variations of these loss functions have also proposed in the literature [107, 108].

We employ Siamese networks in our model to discriminate between feature-specific latent vectors of impostors, and valid pairs. The latent vectors $x1$ and $x2$ are obtained from the feature-extracting autoencoders. $l1$ distance vectors are calculated using the output vectors $h1$ and $h2$ retrieved from the Siamese twins for each module. The decision network is then given the concatenated distance vectors from every module as input as shown in Figure 3.1.

The concatenated input from all modules is fed into the decision network, which is a feed-forward network. This network enables us to incorporate information from all the modules to predict the final decision.

In the face verification task, a pair of face images is given as input, which could be either a valid pair or an invalid pair. In the proposed MSN architecture, the feature extraction encoders included in each feature-specific module provide disentangled face feature embeddings of the input images. These feature embedding pairs are then fed to the Siamese networks present in each module which compute the $l1$ distance vectors of latent embedding pairs. A shared decision network makes the final prediction is then made using the distance vectors from all the modules concatenated.

3.2.1 Achieving interpretability using MSN

The individual modules, and the sub-network in MSN are trained to learn patterns from each facial component from masked face images. A good match by these individual modules indicates that the facial component matches with the facial component of the other face in the pair and are inherently explainable at a facial component level. The distance generated by each MSN sub-network represents how visually similar the features are. This is achieved by computing the Euclidean distance between the twin output vectors produced by the Siamese networks for each module representing a certain feature. A pairwise heatmap that incorporates the similarity or dissimilarity of the feature is created using these distance measurements and is then superimposed on both images. As can be seen in Figure 3.4, the proposed system can efficiently localize the similarities, and dissimilarities of features in a pair of images. These heatmaps might be used as a tool for understanding the decisions taken by the verification system as describe in Section 3.3.1.

3.2.2 Achieving adversarial robustness using MSN

Adversarial attacks introduce carefully calculated perturbations in the facial images either to evade recognition or to introduce backdoors to be exploited by the attacker later. Attackers generated these adversarial perturbations on the whole face with the aim of fooling the target model using any standard face recognition model available in the state of the art, and these adversarially perturbed faces are transferred to the target face recognition model. Unlike some image recognition problems, a face can be recognized from individual facial features to a large extent. This motivated us to use a modular approach to counter adversarial attacks. As the general adversarial perturbations cover the entire face region, and they can only influence the model only if the face is provided in its entirety. Our hypothesis is that, in this MSN based face verification approach [1] we process the face with respect to individual facial components. These partial adversarial perturbations do not impact the verification performance as these perturbations are not computed against the sub-modular networks.

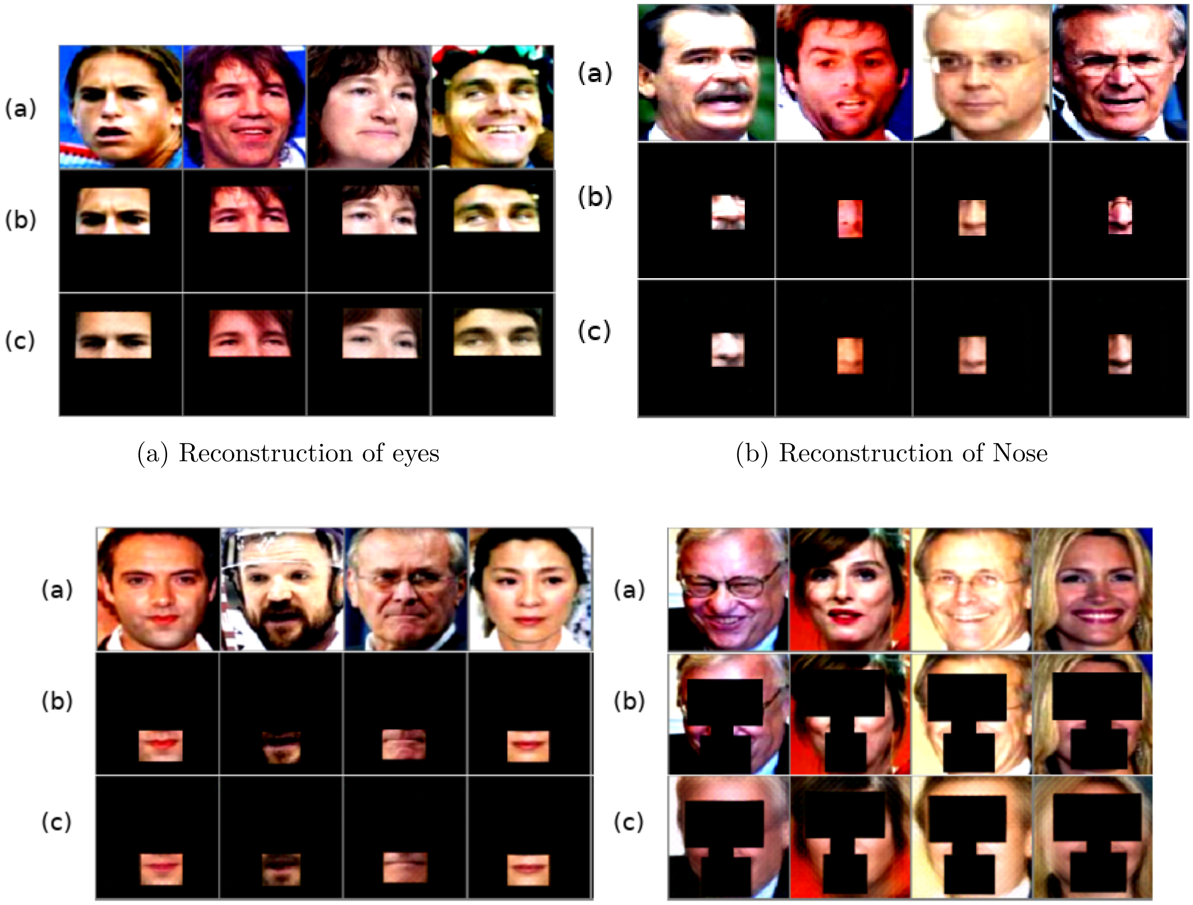
3.2.3 Training MSN for face verification

In the proposed MSN [1], the training is carried out in three training phases. Firstly, the feature extracting autoencoders are trained with perceptual loss [109]. Autoencoder reconstructs each facial component corresponding to the face to learn discriminative latent features against each face component. The input to each feature extracting auto-encoder is the masked facial component image which is obtained by facial landmark detection using MTCNN [50]. These landmark points are used to mask the required facial component before feeding it to the feature extracting autoencoder. We used the VGGFace2 dataset [110] for training the individual modules in the MSN, i.e., component autoencoder, Siamese network, and decision network. Once the autoencoders are trained, latent feature embedding is obtained from the encoder output. In the next phase, the encoder layers trained in the previous step are frozen, as the decoder sections in each module are replaced with the Siamese network, and trained using the triplet loss. Finally, Binary Cross-Entropy is used for training the decision network. The Adam optimization technique [111] was used for training the network in all the three training phases.

From Figure 3.3, we observe that the feature extracting autoencoders are able to learn, and generate high quality reconstructions of the intended facial feature. Once the training is complete, unmasked full facial images are given as input to the autoencoders, and only recreate the necessary facial region by adding pertinent data about that facial feature to the latent feature vector. Since the subnetworks are independent of one another, they can be trained concurrently. We obtain a complete end-to-end face verification system once the training is finished.

3.3 Results and discussion

The proposed modular Siamese network (MSN) was trained on the VGGFace2 dataset [110] In this first the *feature extracting autoencoders* are trained using perceptual loss, and then *Siamese network* is trained using contrastive loss, and finally the *decision network* is trained using binary cross-entropy loss. The trained MSN is evaluated on Labeled Faces in the Wild (LFW) dataset [112] face pairs. For reporting performance, we use 10-fold cross-



(a) Reconstruction of eyes

(b) Reconstruction of Nose

(c) Reconstruction of Mouth

(d) Reconstruction of rest of the face

Figure 3.3: Feature reconstruction results with MSN; Row (a) input image, row (b) masked target image, row (c) reconstructed image using MSN model (Best viewed in color)

validation using the splits defined by LFW *protocol* [112] which serves as a benchmark for comparison. The face verification accuracies of the individual modules, and the proposed MSN model are given in Table 3.1. The face verification accuracies for individual modules have been calculated by finding the optimum distance threshold that maximizes accuracy.

No.	**Module**	**Accuracy**
1.	Module 1 - Eyes	80.8%
2.	Module 2 - Nose	73.2%
3.	Module 3 - Mouth	74.5%
4.	Module 4 - Rest	78.3%
5.	Modular Siamese Network	98.5%

Table 3.1: Face verification performance of modular Siamese network and its sub-modules.

From the experimental results of the face verification, we observe that face verification with just the eye component outperforms the performance of other modules, indicating that eyes are the most discriminating feature in comparison to the mouth, and nose. The face verification accuracy of MSN is 98.5% is close to the state-of-the-art accuracies reported in the literature, which are greater than 99%, such as Facenet.

3.3.1 Feature-level heatmaps

Feature-level heatmaps are intuitive and easily interpretable as humans, unlike computers, look at features as a whole, and not at pixels individually. The proposed method automatically produces paired heatmaps that include relative information that takes into account both the input images. The heatmaps are produced using the feature-wise Euclidean distances calculated by individual MSN modules. As can be seen in each image in Figure. 3.4, features that are visually similar are represented in blue, and features that appear visually different are represented in red. The heatmaps show high similarity for features that are visually close for true positives, as expected. The method also demonstrates considerable nose area dissimilarity between the first impostor pair in Figure 3.4(b), which is consistent with the human perception given that their forms differ greatly. Studying system failures could be beneficial since they provide visual indications that could be used to improve the way the system functions. In the first pair of Figure 3.4(c), we observe that both the persons wearing eyeglasses caused the eyes module to assign a low distance score, and, when accompanied by another similar-looking feature, resulted in mis-classification. The heatmap of the second pair of Figure 3.4(c) demonstrates how spectacles and similar-looking facial hair fooled the system. The heatmaps in Figure 3.4(d) show how the verification can be impacted by closing eyes and a considerable variation in position. The eyes module computed a high distance

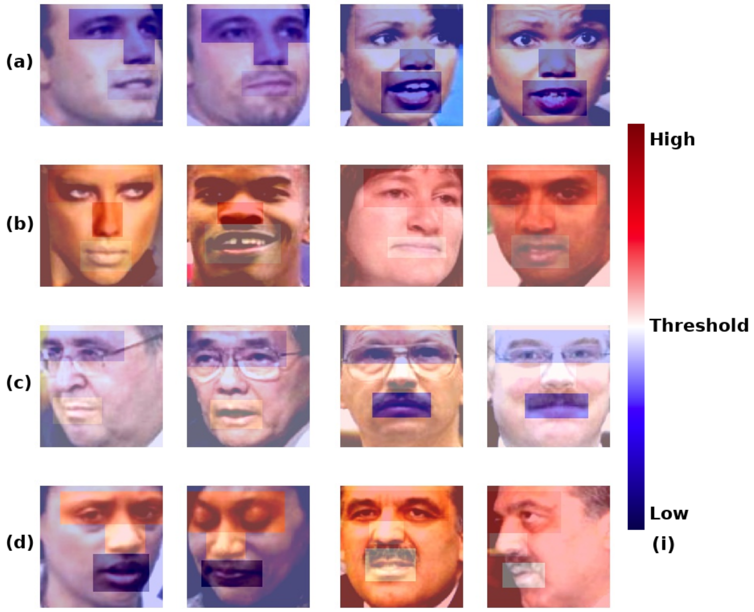


Figure 3.4: Demonstration of facial feature explanations: Each facial factor and its relevance to face verification indicated in color. Blue indicates similarity while red indicates dissimilarity. (a) True positives (b) True negatives (c) False positives (d) False negatives (e) Color map indicating dissimilarity. (Best viewed in color)

score due to the identical person's closed eyes in a photo in the first pair. In the second, the system predicted a high dissimilarity score due to the noticeably different position that caused the partial display of facial features in an image.

Since these computations at the feature level are carried out live, the system could instantly generate meaningful messages that can help the user correct any issues in the case of failure, such as removing eyeglasses or changing poses for better lighting.

3.3.2 Performance under adversarial attacks

We evaluated the robustness and its resistance of the proposed method to well-known adversarial techniques such as the Fast Gradient Sign Method (FGSM) [99], DeepFool [113] and FGSM in fast adversarial training (FFGSM) [114]. Assuming that one of the two image pairs is the anchor image, and the other is the test image, we exclusively attack the test image in a manner similar to the tests carried out in the studies [115, 116]. We used the well-known FaceNet model, which has previously reported state-of-the-art performance, as a point of reference. The results are plotted in Figure 3.5. In comparison to FaceNet, the proposed approach has shown much-improved robustness against all three adversarial attacks.

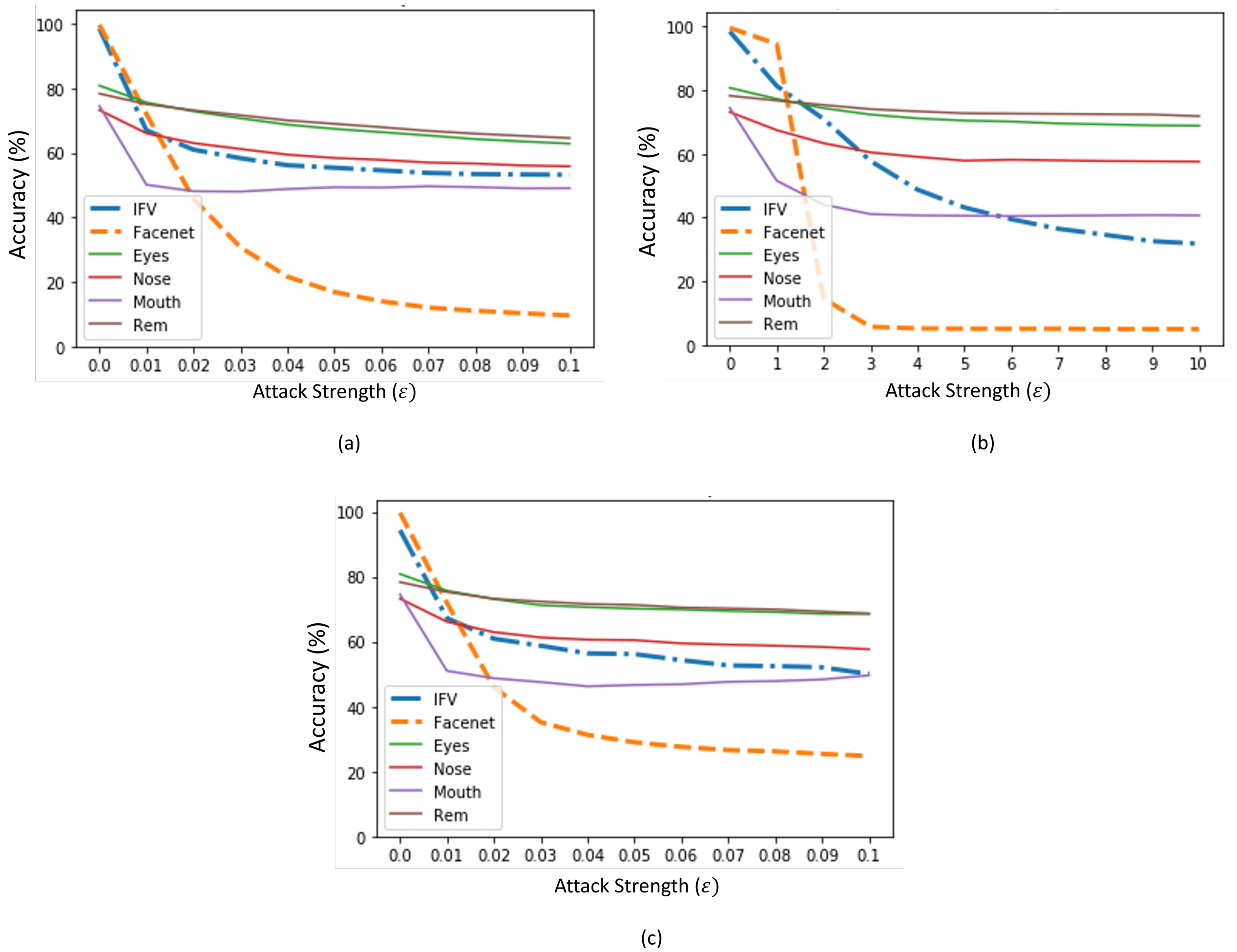


Figure 3.5: Robustness of MSN based face verification against different adversarial attacks, (a) FGSM attack, (b) DeepFool attack, (c) FFGSM attack

The accuracy of FaceNet for FGSM is less than 20% when ϵ is 0.05 while MSN is still close to 60% accurate is shown in Figure 3.5 (a). In the case of the DeepFool attack we observe a dramatic decline inaccuracy in Facenet that falls below 10%, but MSN exhibits much greater resilience by being more than 70% accurate as shown in Figure 3.5(b). Similarly, for FFGSM attack as shown in Figure 3.5(c), FaceNet's accuracy decreases to around 30% while MSN has an accuracy still above 60% at ϵ equals 0.03. As can be observed from the above results, individual modules appear noticeably more resistant to each of these attacks. MSN derives its robustness from these functionally independent modules because it bases its final prediction from individual module output.

The enhanced robustness is attributed to the fault-tolerant nature of MNN [102, 117]. Additionally, due to the bottleneck latent layer, and training to preserve only the most salient features, the encoders used to extract feature-specific latent representations, may be able to offer some immunity against noise or perturbations.

3.4 Discussion

In the literature, various face verification techniques have been proposed, the majority of which are concerned only with performance. As a result, face verification has already attained superhuman precision. Robustness, explainability, and fairness are the aspects where deep learning domain really needs to improve. The proposed method's intrinsic interpretability and resistance to adversarial attacks are its most crucial features. As far as we are aware, no other published face verification technique offers both of these benefits simultaneously. We believe that going in this direction is crucial for developing more trustworthy AI systems.

In many applications, having the ability to comprehend predictions or decisions made by deep learning models could be crucial. While post-hoc interpretations might help to understand the behavior of the model, they may not be of much help in generating real-time explanations. By facilitating contact with the user, explaining what went wrong, and recommending fixes, the system's interpretability could help us deal with human errors. In this chapter, we have explained a novel technique for discovering hidden representations of complex facial features. using deeply embedded feature-specific latent representations, we proposed a modular face verification system that automatically generates decision interpretations. The necessity, and significance of such easily interpretable systems were highlighted. We have also shown that the proposed system is more resistant to adversarial examples.

3.5 Feature based face retrieval

Content-based Image Retrieval is an active research area that deals with searching images by the objects of interest contained in them. Our factorized representation enabled us to query images using a pilot image at a granular level, i.e. individual facial features. In large datasets, adopting methods such as the nearest neighbor algorithm becomes resource-intensive and infeasible. One way to solve this issue is by using approximate nearest neighbor algorithms such as the Locality Sensitive Hashing (LSH). In our experiments, we have employed LSH on factorized latent embeddings to retrieve images containing similar facial features. Content-based image retrieval processes the information contained in an image data and creates an abstraction of its contents in terms of attributes. The problem of image retrieval has been studied in different applications. Specifically, the problem of face retrieval with one sample image becomes a common face retrieval problem [37]. The standard problem formulation for the image-to-image retrieval task is, given a query image, to find the most similar images to the query image among all images in the gallery.

Given an image database or gallery G of N images and a feature dissimilarity function d, find the subset J of images from G and $J \in G$ with the lowest dissimilarity $d(I, J)$ to the query image I. The discriminability of d of the feature vector directly affects the performance of the image query. However, in most cases, the performance of the retrieval systems depends on a perfect query image, and the attributes.

Since faces contain many identical features (areas), it might be difficult for retrieval systems to tell the differences between the faces of various persons. This makes facial image retrieval a challenging task. Particularly, in applications such as surveillance, where the scope for error is zero, it is necessary to improve the retrieval results by augmenting the query. Some methods enable a series of queries to refine the retrieval results, and some other methods present additional metadata along with the query to retrieve a perfect match. Faced with a large amount of data and high-dimensional data information in a database, the existing exact nearest neighbor retrieval methods cannot obtain ideal retrieval results within an acceptable retrieval time.

Due to its quick retrieval time, and inexpensive storage, hashing has been widely used for approximate nearest neighbor search. It has been observed that almost half of the images (about 45%) randomly collected from the Internet contained faces. Therefore, it is necessary to explore the approaches for large-scale face image retrieval. Its goal is to return images containing faces of the same person in the query image. Due to its fast query speed and low storage cost, various hashing methods have been proposed to generate compact binary codes for images.

A major challenge in artificial intelligence is learning the data representations that are disentangled. The disentangled components in the generative process can be specifically controlled and composed of generative models using disentangled latent representations. The foundation of our strategy is the mapping that employs a state-of-the-art pre-trained generator with high quality and fidelity, and to regulate its output with little training in a disentangled manner. Additionally, our method avoids the need for a discrete disentanglement, where the representation is split into two sections that hold totally separate information, because the mapping is trained to only extract the appropriate information from each face area.

3.5.1 Locality sensitive hashing (LSH)

Hashes are common in cryptography. A hash function is often used in computer science to reduce lookup complexity. Formally, a hash function takes an input of variable length, and maps to an output of constant length. To retrieve images having similar features, we use Locality Sensitive Hashing (LSH), which is an approximate nearest neighbor algorithm that reduces the computational complexity to O(log N). When a single bit is damaged, classic cryptographic hashing algorithms such as SHA1, are extremely sensitive to alterations, and produce widely diverse hashes. These can be handy for finding duplicate entries in particular. The opposite is true with LSH, where the objective is to provide identical hash codes for comparable features while accepting tiny modifications. LSH comes in various iterations that cater to various use cases. However, in our approach, we employ LSH based on Euclidean distance. LSH [118] is one of the representative methods, that maps nearby data into comparable binary codes using random linear projections. For retrieving massive amounts of images, LSH is frequently employed.

Given N image samples $X = \{x_i\}_{i=1}^{N} \in R^{d \times N}$, hash coding is to learn a collection of N K-bit binary codes $B \in \{-1, 1\}$. The binary codes are generated by the hash function $h(\cdot)$, which can be rewritten as $[h_1, \ldots, h_K]$. Generally speaking, hashing is learning hash functions to project data samples to obtain a set of binary codes. The core idea behind LSH used in this study is to generate k random hyper planes, and the i^{th} coordinate of the hash value of a given entry x is binary: it is a '1' if x is above the i^{th} hyperplane, and '0' otherwise. Once all entries were assigned a hash code, comparison can be made directly using the hash codes instead of the real entries.

We incorporate supervised hash functions learned from an N-point training set. We explore a hash function which provides hash codes h's for the data points x's that are close in the Hamming space and have equivalent semantic labels, similar to existing supervised "deep learning to hash" methodologies.

The two most important steps of LSH are as follows: 1) Find the *bucket* or *hash value* of the query item. 2) Within the bucket, compare the query item against all items using the Euclidean distance to obtain the most similar item(s).

In our proposed image retrieval system, we use the feature vectors generated by the Modular Network as entries for the described LSH algorithm. For querying through a particular feature, for example, 'eyes', we use the feature vectors generated by the "eyes" module in the MNN.

3.5.2 Methodology

Unlike the earlier approaches where the entire face is used for search, and retrieval in this work, we propose to use individual facial features for the retrieval task. As described in Section 3.2 we learn under complete autoencoders [103] that have a latent bottleneck layer whose dimension is much lower than the input dimension, forcing the model to learn only the most salient features, and ensure they are retained in the encoded latent vectors. We leverage the work we did in Section 3.2, and extend it to the retrieval problem as shown in Figure 3.6.

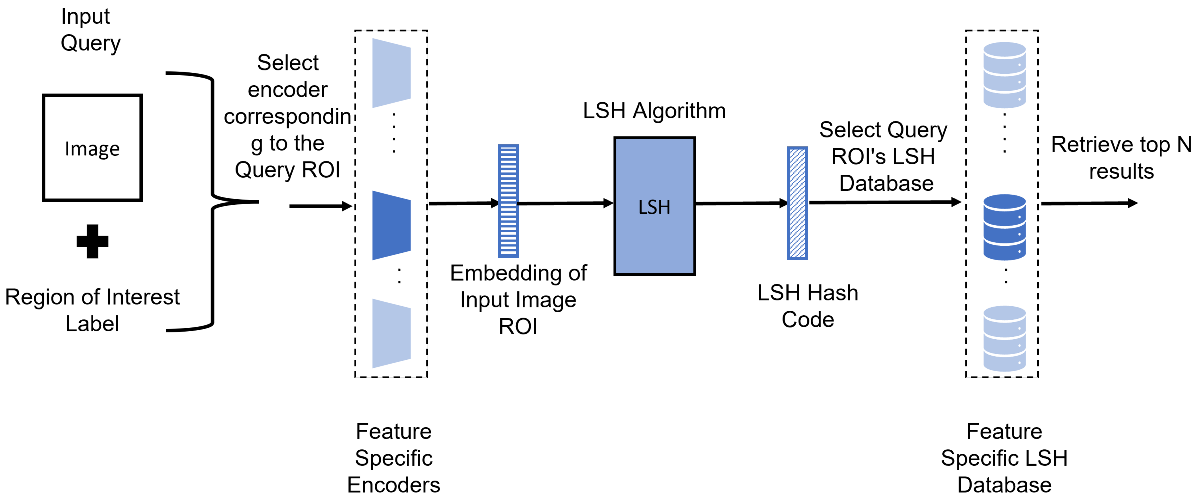

Input Query
Image
Region of Interest Label
Select encoder correspondin g to the Query ROI
Feature Specific Encoders
Embedding of Input Image ROI
LSH Algorithm
LSH
LSH Hash Code
Select Query ROI's LSH Database
Feature Specific LSH Database
Retrieve top N results

Figure 3.6: Proposed face retrieval approach

Facial landmarks are generated for the face to extract the Region of Interest. For example, for face detection MT-CNN (Multi-Task Cascaded Convolutional Networks) was used. Once the region of interest is extracted using landmark points, feature-specific autoencoders are trained using the proposed modified loss function (loss is computed between input image x, and ROI masked image x'). As shown in the Figure 3.6 each ROI will have a feature-specific autoencoder of its own. Once the training is complete, landmark generation is no longer required for the testing/deployment phase. For feature-based image retrieval, the feature-specific embeddings are used for creating *Hash Tables* by employing the LSH algorithm. Each ROI will have a hash table of its own.

During the retrieval, the end-user can submit an image, and specify the visual feature that serves as a primitive for search. As in the training process, the feature-specific embedding is extracted using the relevant feature-specific encoder. LSH hash code is then generated for the extracted embedding. The relevant hash table is queried using the generated hash code, and the top N results were retrieved. For all experiments in this work, VGGFace-2 dataset was used for training, and the LFW dataset was used for validation. LFW dataset contains

Table 3.2: Face retrieval performance with LSH and MSN

Face part	mAP (top 20)
Eyes	0.503
Nose	0.473
Mouth	0.480
Rest of the faces	0.499

13233 images of 5749 people, while VGGFace-2 contains more than 3.3 million images of 9000+ identities. MTCNN was used for aligning the faces, and computing facial landmarks.

To evaluate our image retrieval algorithm, we use Mean Average Precision (mAP) metrics. This metric provides a comprehensive measure of how the retrieval system is performed by considering various queries, and their results on the given dataset. The formula of mAP is given in the Equation 3.1.

$$mAP = \frac{1}{N}.\sum_{i=1}^{N} AP_i, \tag{3.1}$$

where N is the total of number of queries being used for evaluation, i is the query index, and AP is the Average Precision of the current query item.

3.5.3 Results and discussion

To see the effectiveness of the Modular Siamese network in retrieving it is queried with different facial features. Table 3.2 shows the mean average precision for queries based on eyes, nose, and mouth parts of face image. From the Figures 3.7, 3.8, and 3.9, we can observe that the resulting images for each query have similar visual attributes as the query image in the features of interest. In a few cases, we notice that the images may look different when viewed as a whole but very similar when focusing only on the region of interest, which is how a typical CBIR system is expected to work. For example, for the mouth attribute, the query image has mouth open Figure 3.9, and the resultant images also have similar mouth characteristics.

3.6 Conclusion

A huge amount of research happened in the field of face verification and the majority of which are concerned only with performance. Due to the critical nature of the biometrics systems, there is very less room for mistakes. Incorporating interpretability into the system itself could allow us to handle errors by informing them of what went wrong, and suggesting rectification.

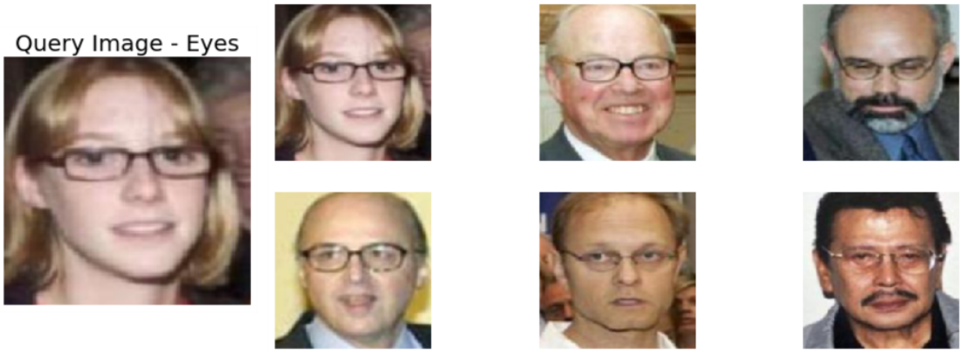


Figure 3.7: Face retrieval, querying using eyes. (left, query image; others, retrieved faces for the query image)

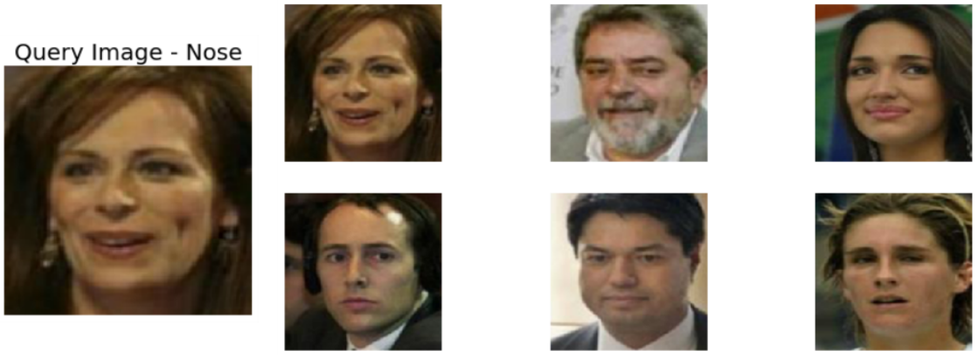


Figure 3.8: Face retrieval, querying using nose; (left, query image; others, retrieved faces for the query image)

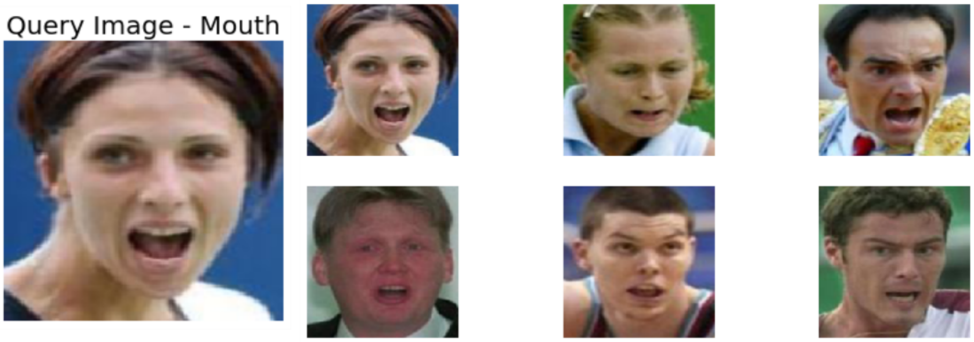


Figure 3.9: Face retrieval, querying using mouth;(left, query image; others, retrieved faces for the query image)

This chapter introduces a novel method for learning high-level latent representations of facial features. We suggest a modular face verification system that inherently generates interpretations of its decisions with the help of the learned feature-specific latent representations. The suggested method's intrinsic interpretability, and resistance to adversarial attacks are its most important features. As far as we are aware, no other published face verification technique offering both of these benefits simultaneously. We think that going in this path is crucial for creating more reliable systems. The necessity, and significance of such easily understandable systems were highlighted. We have also shown that the suggested system is more resistant to adversarial examples.

We extended the findings of feature-specific representations to retrieve facial images closely matching the query image. This query image could be itself a facial part, and the approach can retrieve all the faces having similar feature representations. This is critical in applications such as surveillance, where the face is only partly visible, and available for recognition.

While the security of the input facial data is important in a face recognition system, security of the template data is equally important. In the next chapter, we discuss a deep neural network based method that can protect the facial template data and enable recovery even if they are compromised without affecting the performance of the face verification.

Chapter 4

Face template protection

Due to the number of data breaches that affect the privacy of hundreds of millions of identities, individuals are becoming concerned about where and how their biometric data is stored and used. Facial recognition is an authentication mechanism for accessing both logical (e.g., network, data, transaction, etc.) and physical (e.g., building, door, elevator, etc.) resources, and serves as an effective security mechanism. However, attackers continuously seek to exploit the vulnerabilities of the biometric systems to launch attacks and steal private and confidential data. Also, the attackers could trick biometrics systems such as face recognition by cameras using photographs, video clips, and even 3D masks.

A major challenge for a biometric template protection method in satisfying the above requirements is the high intra-user variability in the biometric templates [119, 120] and low inter-user variability. High False Rejection Rates (FRR) are a result of high intra-user variability (due to variations in pose, illumination, emotion, etc.), and high FAR results from low inter-user variability. Present state of art methods tried minimizing intra-user variability and maximize inter-user variability by multiple acquisitions of the user's face with variations in the pose, illumination, and expression to achieve high template security and matching performance together. However, there is a trade-off between template security and matching performance.

In this chapter, a face template protection method using a deep neural network [2] is described in Section 4.1. In Section 4.2, the implementation details of the proposed method are discussed. In Section 4.3, studies on matching performance are presented. In Section 4.4, a template protection using adversarial perturbations is presented. In Section 4.5 details of auxiliary data generation process are provided. Section 4.7 describes the experimental details of the proposed method. In Section 4.8 results of the proposed method are described. Section 4.9 provides the conclusion of the chapter.

4.1 Face template protection using deep mapping network

Template protection mechanisms prevent impersonation attacks when the templates are compromised. Earlier approaches such as *Biometric cryptosystems* and the *Transform-based* template protection methods suffer from the trade-off between security and performance. A reason for the reduction in matching performance of the template protection methods is their inability to handle intra-user variations. To provide a more reliable face template protection solution with improved matching performance, enhancement to the architecture for face template protection put out by Pandey et al. [90]. In this work [2], deep neural networks combined with VGGNet for face template protection is used. A robust mapping is performed from between the face representations of a user and a pre-generated high entropy unique binary code during the enrolment phase using a deep neural network to overcome the limitations of intra-user and inter-user variability. The use of pre-trained VGG architecture and robust mapping network jointly minimizes the intra-user variations and maximizes inter-user variations. The performance of our face template protection method is compared with the other algorithms proposed in [72, 90, 121] on the CMU-PIE dataset. Compared to related work, the proposed solution provides strong template security, while increasing matching performance of about 4% and decreasing Equal Error Rate (EER) by about four times.

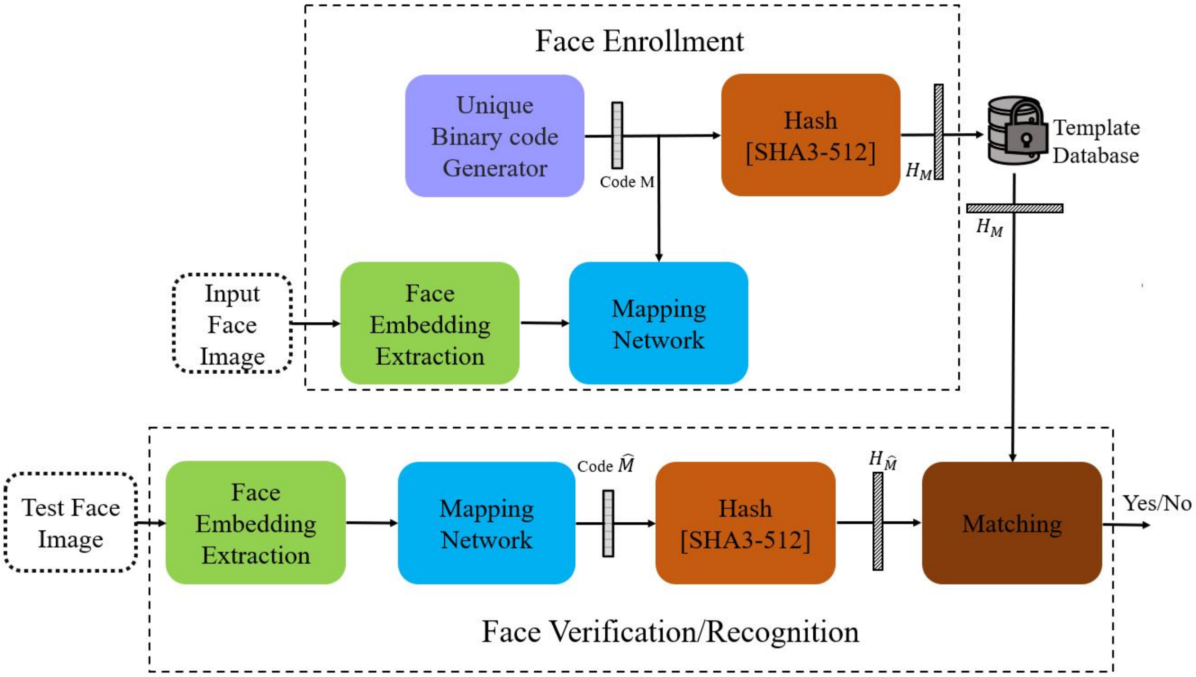


Figure 4.1: Block diagram of the proposed face template protection method [2]

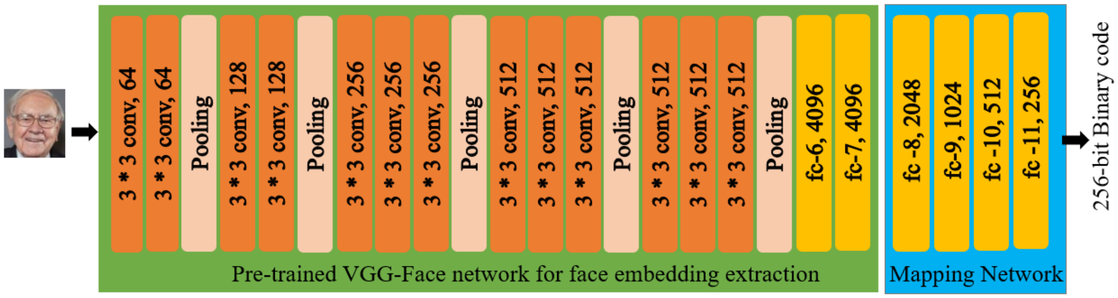


Figure 4.2: Architecture of the proposed mapping network

4.2 Methodology

As shown in Figure 4.1, the proposed template protection approach, for enrolling a user, a set of unique and high entropy 256-bit codes are generated by a random generator following uniform distribution, and each user is assigned a code. These codes have no correlation with the biometric data of any user, making it impossible to reconstruct biometric data from these codes alone. The code assigned to each user is called the original template or unprotected template. The code generation process satisfies the diversity and cancelability requirements of a template protection process. If a template is compromised, a new code is assigned to the user, and hence a new template is generated. Only the original template's cryptographic hash is kept in the template database as a protected template upon enrollment, and the original template is discarded after training the mapping network.

In our studies, a pre-trained VGGFace network is used for extracting facial feature embeddings. The input to the pre-trained VGG-Face network is a 224x224 face image, and the output is a 4096 dimensional feature embedding taken at fully connected (fc-7) layer, as shown in Figure 4.1. During the enrollment phase, multiple fully connected layers (fc-8, fc-9, fc-10, fc-11) are added to learn the robust mapping of the extracted feature embeddings representing the user's face image to the assigned binary code(unsecured template).

The modifications done to the network configuration include i) The binary codes supplied to each user replace the one-hot encoding of class labels, ii) the last layer of the network uses the Sigmoid activation functions instead of the Softmax function, and iii) binary cross-entropy loss function is employed.

The performance of the proposed template protection approach was evaluated on various face datasets. In this work, we use CMU-PIE [122] with 68 identities, FEI [123] with 200 identities. Recognition performance evaluation is done for multi-shot enrolment. In one-shot enrollment, a random image per user is used for training, and the remaining images were used for testing. In multi-shot enrolment, few images are chosen at random from the set for training, and the remaining images are used for testing as it is done in [72, 90].

Table 4.1: Secure Face recognition performance on different datasets

Dataset	**Enrollment Type**	**Code size (K)**	**GAR@FAR**	**EER**
PIE	Multi-Shot	256	97.35%@0%FAR	0.15%
		1024	96.53%@0%FAR	0.35%
FEI	Multi-Shot	256	98.54%@0%FAR	0.16%
		1024	99.10%@0%FAR	0.20%

4.3 Face matching performance

The simulation results of the proposed approach are shown in Table 4.1. The proposed method not only generates very low FAR values even at low matching scores, but it also achieves very low FRR (or high Genuine Accept Rate (GAR)) values even at matching scores as high as 1. The genuine acceptance rate is slightly low for one-shot enrolment cases, whereas for the multi-shot enrolment case, the accuracy is high and better than the state-of-the-art. If the dataset has pose, illumination, age and occlusion (glasses) variations,it may cause poor performance.

Table 4.2: Face recognition performance comparison on PIE Dataset

Dataset	**Enrollment Type**	**Code size (K)**	**GAR@FAR**	**EER**
Hybrid Approach [72]	Multi-Shot	210	90.61%@1%FAR	6.81%
BDA [121]	Multi-Shot	76	96.38%@1%FAR	-
MEB Encoding [90]	Multi-Shot	256	93.22%@0%FAR	1.39%
		1024	90.13%@0%FAR	1.14%
Our Approach	Multi-Shot	256	97.35%@0%FAR	0.15%

We studied the impact of the size of the binary codes on the overall performance of the method. In the mapping network, number of fully connected layers chosen depending on the binary code dimension (K=256) to be mapped. Four fully connected (fc-8, fc-9, fc-10, fc-11) layers are used to convert the 4096-dimensional feature vector to a 256-bit binary code, as shown in Figure 4.2. For the same enrolment scheme and dataset, a change in binary code length impacts the matching performance slightly, though not significantly. This can be attributed to an increase in the number of network parameters that are learned, but the size of the dataset is constant.

The performance of the proposed approach is compared with the state-of-the-art methods of Face template protection. Earlier methods were evaluated only for multi-shot enrolment. A comparison of simulation results is shown in Table 4.2. As can be observed from the table, the proposed approach outperforms the previous method[90] by approximately 4%, and exhibits significant improvement in EER.

4.3.1 Security analysis

The proposed approach needs to be analyzed for security, as this is one of the critical parts of the design consideration. The following parameters need to be evaluated using a typical template protection scheme.

1. **Revocability or cancellability:** If a template of one identity is compromised, then the template is removed from the database and a new unique code (original template) is assigned to the user, and the a new protected template is generated and replaced in the template database. The network is retrained with a new code assigned. Even if the attacker presents the stolen template to the Matcher, the templates do not match, and the revocability property is satisfied.

2. **Diversity:** As the codes are generated from a random generator, each code generated for the same user does not match, satisfying the diversity property. Even if the same user or identity is registered with another application that uses a similar scheme for face verification, the codes that are generated by the random generator would be different.

3. **Non-invertibility:** Each protected template generated for the users are the SHA-3 hashes. No information about the original template or binary code can be extracted from the stolen protected face template, as brute force attacks in this scenario are computationally infeasible. For a 256-bit binary code, the search space would be 2^{256}. Unless the SHA-3 is broken, the attacker may not be able to retrieve the original unsecured templates (binary codes C_M).

To evaluate the template security, with respect to FAR (FMR) and FRR (FNMR), studies on the genuine and imposter score distributions are conducted for the following dictionary attacks on the proposed method with the model trained with multi-shot enrollment: (i) In the first attack, the PIE database is used as an attacker dictionary and the FEI database as a genuine face database. (ii) In the second attack, the two datasets were swapped. Distributions of the genuine and impostor scores for the aforementioned dictionary assaults demonstrate that the genuine scores tend to be '1' and the imposter scores tend to be '0', demonstrating how improbable it is for the proposed approach to false accept the external faces for the enrolled ones. Hence the current scheme satisfies both the security and matching performance requirements as expected from any biometric template protection mechanism. The zero FMR can be attributed to the limited size of the dataset used for enrollment and verification.

A major limitation of the proposed face template protection approach is that, re-enrollment of a user requires retraining of entire mapping network. This may not be practically feasible in applications handling millions of users for face recognition.

4.4 Face template protection using adversarial perturbations

The deep learning based template protection scheme discussed in the previous Section 4.1 suffers from the fact that if one of the templates is compromised entire network requires retraining for re-enrolling the user. For systems having thousands of identities enrolled for authentication, this retraining time could be a roadblock. This motivated us to develop an approach that requires little or no retraining. Adversarial attacks [99, 124, 125] on machine learning models subtly and carefully modify the input data resulting in misclassification by target models. These adversarial attacks are evolving and attackers are finding new ways to perturb the input to force the model to produce incorrect predictions in a targeted and non-targeted manner. However, the very same idea of computing input perturbations is used to our benefit, i.e., to generate renewable face biometric templates to achieve high template security without compromising on the matching performance and to overcome the re-enrollment problem when one or more templates in the database are compromised. Here the goal is to protect the templates generated from attacker manipulations, and simultaneously, avoid retraining of the entire system.

4.5 Adversarial perturbations as auxiliary data

As mentioned in Section 2.2.3, an ideal biometric template protection scheme shall meet the following requirements: (i) Diversity (ii) Revocability, (iii) Security and (iv) Performance. To address the problems mentioned earlier, a hybrid method for facial biometric template protection is proposed [3], which provides better security without compromising on the matching performance. and addresses the re-enrollment issue. The proposed hybrid method combines the transform-based approach [88, 90, 126], adversarial noise generation (perturbation generation) approach, and the biometric cryptosystems approach [84, 86, 87] for template protection. This hybrid approach is a combination of the biometric cryptosystem and transform-based approaches, and is motivated by Feng et al. [119].

The idea of targeted adversarial example generation [99, 113], i.e., computing the adversarial noise as a function of gradients of the network to fool the network to generate class-specific perturbations where a randomly generated high entropy binary code is used as target label vector for each identity. The final perturbations generated after several iterations continued till the target label vector is obtained in the output and are used during verification. The assigned target binary codes are hashed using SHA3-512 and stored as a protected template in the database for matching.

Re-enrollment issue: Many of the biometrics approaches learn a classifier on the biometric feature descriptors. The objective of this kind of classifier is to maximize the inter-class distances and minimize the intra-class variations. This optimization process requires entire data to be present for learning purposes. This prevents any new identity is added without retraining the classifier on the entire data. Specifically, in template protection methods, whenever a template is compromised, a new template is generated and stored in the database. Although the template protection mechanisms prevent the templates from being used to gain access, this requires computing the templates by retraining the classifier. Our previous works [2, 4], use a deep neural network to learn a mapping from a user's face image to the unique binary code assigned to the user during the enrollment phase. These binary codes were hashed and stored as templates. However, these algorithms require retraining and re-enrollment, even if one of the user's protected templates is compromised. Re-enrollment problem is addressed by renewing the templates (binary codes) if the corresponding stored templates (a few or complete databases) are compromised. When the protected template is compromised, a new target binary code is assigned to the user whose template is compromised. A new set of perturbations or helper data is generated with respect to the new target binary code without affecting any network parameters.

Adversarial examples are examples with minor perturbations which result in incorrect model predictions. These examples are useful in exposing the adversarial scenarios where the models fail. However, these malicious perturbations are often unnatural and not semantically meaningful. Ian Goodfellow et al. [99] introduced the 'fast gradient sign' method, which computes the adversarial perturbations for a given classifier very efficiently. The method was applied in [127] to generate adversarial perturbations. The adversarial noise generation concept is used to overcome the re-enrollment and retraining problem. In this approach, targeted adversarial noise (perturbations for the feature vector) is generated, which can be used as helper data during verification as in biocrypto systems. In case any template is compromised, a new code to the user is assigned along with new adversarial perturbations corresponding to that code.

4.6 Methodology

The proposed method for face template protection follows these major steps: (i) Extracting feature vectors from the input face images. Here, VGG-16 pre-trained face model was employed to extract input image feature embeddings. (ii) Generating, assigning and mapping random binary codes to each user or identity. Then generate random perturbations (random noise) as in targeted adversarial learning such that the perturbed input feature vectors are mapped to assigned codes. (iii) While mapping the perturbed input to the assigned code, perturbations also called helper data are computed as a function of gradients of the mapping network.

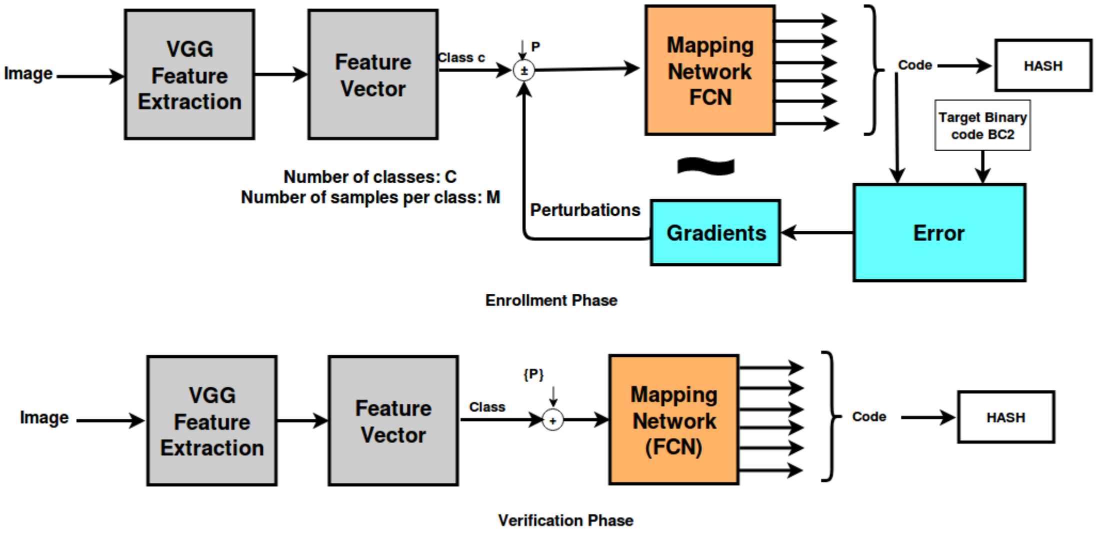


Figure 4.3: Block diagram of proposed template generation process using adversarial perturbations as helper data [3]

4.6.1 Pre-processing and feature extraction

Face Detection: Detected human faces are resized to a 224x224 image (such that the face is at the center of the image) and saved.
Augmentation: As deep learning models require large training data, and due to the limited size of the biometric data, data augmentation is a must to simulate different variations in the data.
Feature Vector Extraction: A pre-trained VGG-Face network (VGG16) was used to extract a 4096-dimensional feature vector corresponding to each input face image. The input to the pre-trained VGG-Face CNN is an image with a size 224x224, and the output is a one-hot vector of length 2600. The first 15 layers of the model are considered (i.e. fc7 layer

as output as in the Figure 4.4) for the feature vector extraction. The pre-trained VGG-Face architecture captures the uniqueness in the extracted feature set of each user, thus maximizing inter-user variations.

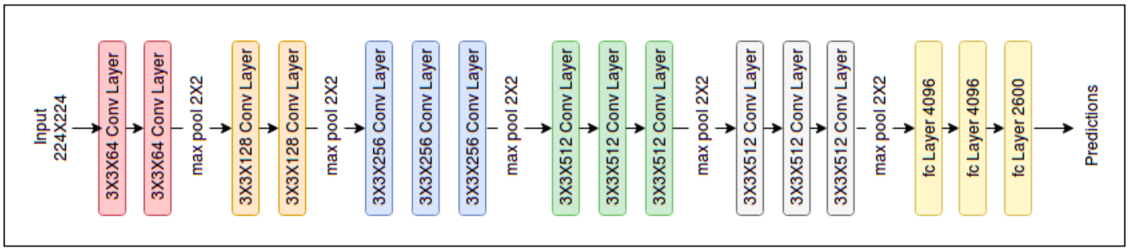


Figure 4.4: VGG-16 Deep face model Layers

4.6.2 Mapping binary codes

The first step of the enrollment process is the generation of unique binary codes of 256-bit length with maximum entropy. Initially, two sets of binary codes are generated, and each user in the training set was assigned a unique binary code from the first set of binary codes. These first sets of binary codes are used in training the fully connected network mapping the unperturbed feature vectors to assigned binary codes. This method of binary code generation and assignment satisfies the diversity and revocability requirements of a biometric template protection scheme. It is to be noted that different applications can assign mutually exclusive binary codes to each enrolled user. An application can also change the binary codes assigned to its enrolled users (or enroll new users by assigning them new binary codes), when the corresponding template is compromised. The last layer of the neural network uses sigmoid activation function instead of *softmax* and uses binary cross-entropy as the loss function. To map the 4096 dimensional feature vector to a 256 bit binary code and four fully connected layers are used (namely, fc-8[dim:2048], fc-9[dim:1024], fc-10[dim:512], fc-11[dim:256]) as shown in Figure 4.5. The output is binarized using a simple threshold operation, where the output value is set to 1 if greater than 0.5, else it is set to 0, thus predicting the binary code corresponding to the input face image. The mapping network described above minimizes the intra-user variations while mapping the extracted feature vector to the bitwise randomly generated unique binary code assigned to each user.

4.6.3 Adversarial attacks

4.6.3.1 Fast gradient sign method

The FGSM attack proposed by Goodfellow et al. [99], generates an adversarial example faster in a white box setting. In a white box scenario, attacks are aware of the model parameters and training data. FGSM performs one step gradient update along the direction of the sign of gradient at each pixel. The adversarial example generation can be defined as

$$x' = x + \epsilon \cdot sign(\nabla_x J_\theta(x, y)) \tag{4.1}$$

where ϵ is the magnitude of the perturbation and is chosen sufficiently small to be undetectable. J is the loss function associated with the neural network F. The adversarial example x' that was generated is obtained as $x' = x + p$. Back-propagation is used to calculate this perturbation.

4.6.3.2 DeepFool

In this technique [113] minimal perturbed adversarial samples are generated by minimizing the Euclidean distance between the original and perturbed samples. Perturbations are added iteratively by using the decision boundaries between classes. The main advantage of this method is, creating effective adversarial samples with higher misclassification rates.

4.6.4 Helper data generation

According to the adversarial example generation, infinitesimal changes in the input can add up to one large change in the output. Networks using *ReLU* and max-out activation functions are intentionally designed to behave linearly so that they are easier to optimize. However, in non-linear activation functions like sigmoid, networks always try to spend most of their time in the non-saturating (linear part of the sigmoid curve) region. This linear behavior of the sigmoid networks suggests that the analytical perturbations in the input, like in linear models, can affect the output of the sigmoid network. In this proposed approach [3], the linear behavior of the sigmoid activation function at the output of the mapping network is used to generate the linear perturbations as a function of gradients of the mapping network. For perturbation generation, a targeted adversarial noise generation approach is used, in which the gradients are computed based on the error computed between the target label and the generated label at the network output. The same approach has been applied here by computing the gradients of the network based on the error between the target binary code (second binary code assigned to each user) and the generated binary codes at the output of the mapping network.

In this, initial random perturbations(random noise) are generated during the first iteration, and added them with the feature vectors to generate the corresponding binary code

(BC') at the output. This generated binary code is compared with the second binary code (BC2) that is assigned to the user to compute the error. Based on this error, the gradients of the mapping network are computed, which were used to estimate the perturbations. These perturbations are cumulatively added to the initial random perturbations until the generated binary code (BC') is equal to the BC2 as given in the Algorithm 1.

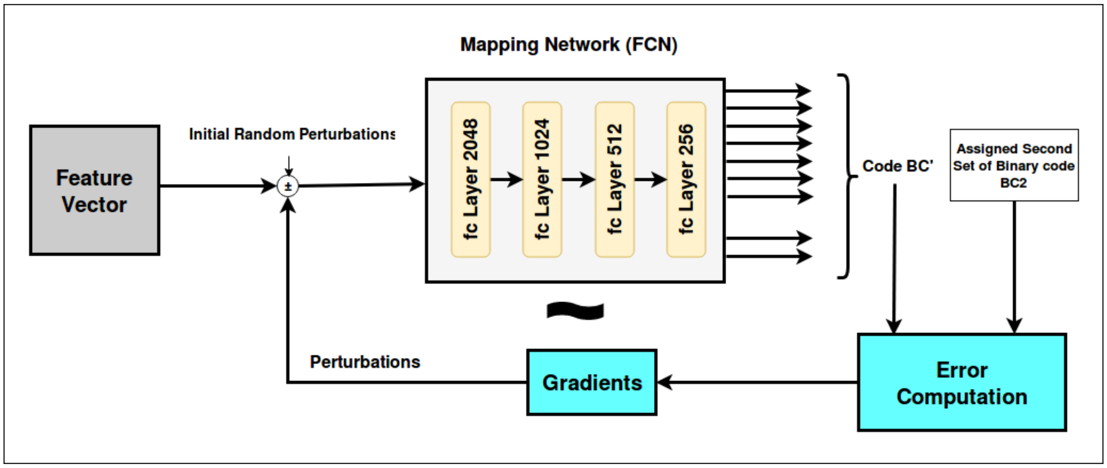


Figure 4.5: Feature embedding to binary code Mapping

4.6.5 Generating auxiliary data for template protection

The proposed method for face template protection as shown in Figure 4.3, contains the following major steps [3]:

1. First, the input face image is fed to pre-trained VGG-16 for extracting feature vectors.
2. Pre-train the mapping layers with facial feature vectors as the input and a generated set of binary codes (BC1) as the output.
3. Generate, assign and map a random binary code (BC2) to each user or identity, which acts as a label for targeted learning.
4. Generate and assign a set of random initial perturbations to each user.
5. Store the 512-bit cryptographic hash of these codes as protected templates in a database.
6. Adversarially attack the pre-trained mapping network and iteratively compute random perturbations 'P', such that the perturbed input feature vectors are mapped to assigned codes (BC2). Please refer to Algorithm 1 for the adversarial example generation *BC2*.

7. The final cumulative perturbations called helper data were stored in a different database for better security.

8. While mapping the input to the assigned code, perturbations are computed as a function of gradients of the mapping network.

Algorithm 1: Helper data generation

```
Input: feature vectors $\mathbf{V}$, initial Random Perturbations $\mathbf{P}$,
target labels $\mathbf{BC2}$ , mapping network $\boldsymbol{f}$
Output: Final Cumulative Perturbations $\mathbf{P}$
$V_0 \leftarrow V + P, i \leftarrow 0$ ;
while $BC'! = BC2$ do
    $W_k^{'} \leftarrow \bigtriangledown \mathbf{f_k} \left(\mathbf{V_0}\right)$;
    $\Delta\ \mathbf{P} \leftarrow \alpha \frac{W_k^{'}}{\left\|W_k^{'}\right\|_2^2}$;
    $\mathbf{P} \leftarrow \mathbf{P}\ +\Delta\ \mathbf{P}$;
    $V_{i+1} \leftarrow V + \mathbf{P}$;
    $i \leftarrow i + 1$;
return $\boldsymbol{P}$;
```

4.6.6 Cryptographic Hash

To protect the template, represented by the unique binary code assigned to each user, a Secure Hash Algorithm (SHA-3) is used. The input to the SHA3-512 algorithm is the binary code assigned to the user, and the output is the 512-bit cryptographic hash representing the protected face template. During the enrollment phase, the hash of each binary code, representing the protected face template, is stored in the database.

4.6.7 Matching

During verification, the VGG-Face model extracts the feature vectors, which are fed to the fully connected mapping network along with perturbations to obtain the binary code BC2' corresponding to the user. This binary code BC2' is hashed in a similar way as done during enrollement, and compared with the protected templates corresponding to the user in the database.

4.7 Experiments

4.7.1 Databases

To evaluate our approach three databases were considered: (i) The CMU PIE [122] database consisted of 41368 images of 68 subjects. Each subject consists of images under 43 different illumination conditions, 13 different poses, and four different expressions. Five poses (p05, p07, p09, p27, p29) and all illumination variations were considered for our experiments. In the enrollment, 10 images were randomly chosen, and the rest were used for testing. (ii) The FEI [128] database consisted of 2800 color images of 200 subjects. Each subject has 14 images with a pose rotation of up to about 180 degrees. Among the 14 images, nine poses (p03, p04, p05, p06, p07, p08, p11, p12, p13) were used for experiments. For the enrollment, four images are randomly chosen for training, and the rest five are used for testing. (iii) In the color-FERET [129] database, 237 individuals were selected, and each individual had four different face images with pose, illumination, age, and occlusion(glasses) variations. For the enrollment, two images are randomly chosen for training, and the rest two are used for testing.

4.7.2 Experimental parameters

In training, VGG-16 face model is used to extract the feature embeddings from the image. To map these extracted feature embeddings to a 256-bit binary code, fully connected layers (fc-8, fc-9, fc-10, and fc-11) are used with dimensions of 2048, 1024, 512, and 256, respectively as shown in Figure 4.5. To reduce the overfitting in the deep CNN, dropout [130] is applied in all these fully connected layers with a 0.25 probability of discarding one hidden activation. Each layer of these fully connected layers, except the last fully connected layer, uses ReLU activation function. The last fully connected layer uses a sigmoid activation function, as the output is not the one-hot vector but a 256 bit length code. To train the mapping network, Adam optimizer is used along with binary cross-entropy loss for 50 epochs with a batch size of 16. During verification, the network predicts the binary code from a given input face image. The binarized output is obtained through a simple thresholding operation, with a threshold of 0.5 on each output.

4.8 Results and discussion

To evaluate this approach, four experiments were conducted with untrained and trained fully connected networks, and with and without initial random perturbations. (i) In the first experiment, an untrained network is considered with random weights and updated the perturbations on the input (ii) In the second experiment, random initial perturbations(helper data) are generated, and then fed along with the input feature vectors into the mapping network. (iii) In the third experiment, a pre-trained mapping network with feature vectors as the input used, and perturbations on the input are computed from the gradients of a fully connected network with target codes. (iv) Finally, in the last step, perturbations are computed from the gradients by feeding feature vectors along with initial random perturbations and considering the second set of binary codes BC2 as target labels. Table 4.3 gives the results of our approach using the raw network, and Table 4.4 gives the results with a trained network.

Table 4.3: Face recognition results with untrained mapping network

with out initial noise			
Dataset	K	GAR% @ FAR%	EER
PIE	256	99.90% @ 2.75%	2.03± 0.05%
FEI	256	99.80% @ 2.72%	1.96± 0.25%
Color FERET	256	94.30% @ 1.97%	4.02± 0.40%

with initial noise			
Dataset	K	GAR% @ FAR %	EER
PIE	256	99.85% @ 2.43%	1.97± 0.05%
FEI	256	99.77% @ 2.57%	1.89± 0.35%
Color FERET	256	94.12% @ 1.70%	3.87± 0.47%

Table 4.4: Face recognition results with trained mapping Network

with out initial noise			
Dataset	K	GAR% @ FAR%	EER
PIE	256	97.90% @ 0.00%	0.30± 0.09%
FEI	256	98.80% @ 0.00%	0.32± 0.14%
Color FERET	256	93.75% @ 0.02%	3.20± 0.29%

with initial noise			
Dataset	K	GAR% @ FAR%	EER
PIE	256	97.87% @ 0.00%	0.20± 0.13%
FEI	256	98.75% @ 0.00%	0.27± 0.08%
Color FERET	256	93.72% @ 0.02%	3.03± 0.35%

Based on the results, it can be noticed that the proposed method not only has very low FAR values even at low matching scores, but also has very low FRR or high GAR.

4.8.1 Security analysis

In this method, the protected face template is the SHA3-512 hash of the unique binary code (bitwise randomly generated) assigned to a user during the enrollment phase. In this, the attacker would have access to only the protected face template with no knowledge of the feature extraction procedure as well as the mapping procedure, and no information about the original binary codes can be extracted from the stolen protected face template. This is due to the non-invertible nature of the cryptographic hash functions. Therefore, in such a scenario, only brute force attacks (trial and error) can reveal the binary codes. However, the brute force attacks in this scenario are computationally infeasible, as with 256-bit binary code the search space would be 2^{256}. To evaluate the template security, genuine and imposter score distributions are studied for the following dictionary attacks on the proposed method with the deep CNN model trained with multi-shot enrollment and K=256. (i) In the first attack, FEI database is used as the genuine face database and PIE database as an attacker database. (ii) In the second attack, PIE database is used as the genuine face database and FEI database as an attacker database. The genuine and imposter score distributions for the above dictionary attacks reveal that the imposter scores tend to zero and genuine scores tend to 1, thus showing that it is unlikely for the proposed method to falsely accept the external faces for the enrolled ones. One reason for the zero FMR could be the smaller size of the datasets.

During the verification phase, given a face image, facial feature vectors are computed. These feature vectors are added with the helper data(perturbations) P_i retrieved from the database corresponding to that identity, and then provided to the mapping network. The output of the mapping network is hashed using SHA3-512 before comparing with the template ($BC2_i$) stored in the database of the user for verification. This approach does not require re-enrollment of all users when one or more templates are compromised.

Table 4.5: Face verification performance results on various datasets

Dataset	**K**	**GAR% @ FAR%**	**EER**
PIE	256	97.87% @ 0.00%	0.20±0.13%
FEI	256	98.75% @ 0.00%	0.27±0.08%
Color-FERET	256	93.72% @ 0.02%	3.03±0.35%

Along with the non-invertibility provided by SHA3, an additional layer of non-invertibiltiy is achieved through the mapping network, as the mapping is many-to-one. Even if the attacker gets access to the mapping network with brute force attacks on the hashing algorithm, finding the input feature vector from the binary codes is difficult as the helper data(perturbations) P_i computed during enrollment are secret. If any stored template is

Table 4.6: Face verification performance comparison with other methods on PIE dataset

Method	K	GAR% @ FAR%	EER
Hybrid Approach [72]	210	90.61% @ 1%	6.81 ± 0.00%
Deep Secure Encoding [90]	256	93.22% @ 2.61%	1.39 ± 0.20%
Our method (pre-trained)	256	97.87% @ 0.00%	0.20 ± 0.13%
Our method (untrained)	256	99.90% @ 2.75%	2.03 ± 0.05%

compromised, a new code is generated and assigned to the user as a target label in calculating new adversarial perturbations and stored in the database. These perturbations, are uncorrelated with those generated earlier, as the initial perturbations are different.

4.8.2 Face verification performance with cancellable auxiliary data

Although the main objective of the approach is to protect the face templates, it is essential to ensure that the security measures do not affect the matching performance. Table 4.5 shows the performance of the proposed scheme against PIE and FEI datasets [3].
Table 4.6 shows comparison of our approach with a similar approach in the literature [90]. This approach achieved 99.9% accuracy on PIE dataset, which is about 7% jump, at relatively same FAR, also the jump in the accuracy is 4.5% when the FAR is 0% as shown in Table 4.6. Based on the results, it can be noticed that the proposed method not only has very low FAR values even at low matching scores but also has a very low FRR or high GAR.

4.8.2.1 Re-enrollment

The proposed approach avoids complete retraining of the network for all identities when a template of the user is compromised as the perturbations are kept secret. When a template is compromised, it is sufficient to generate new perturbations for that user with the new code BC2 assigned, and the corresponding new template is stored in the database. This will not affect the helper data (perturbations) already generated for other users in any way.

In case any store template is compromised, the target binary code of the user is replaced with a new randomly generated binary code and computes the final perturbations corresponding to that new target label. To test the feasibility of the approach for re-enrollment without affecting the network parameters, different random noises are used, and computed the final perturbations for each class. The final perturbations computed in this way are uncorrelated with each other, i.e., for each initial random perturbation (random noise), a new final perturbations for the same class of feature vectors is computed. Therefore, even if a stored template is compromised, a new target binary code is generated that is uncorrelated with other binary codes, and is assigned to the users to compute corresponding perturbations without affecting the network parameters.

4.9 Conclusion

In this chapter, two methods for template protection are presented (i) Using a deep neural network to map facial feature embedding to cancellable binary templates. (ii) Using targeted adversarial noise generation, which is used as auxiliary data, and cancellable binary templates. A deep CNN is used to provide a better matching performance method. With this approach, an improvement in matching performance by 6% and reduction in EER by about four times is achieved compared to earlier approaches, while providing high template security. The current approach deals with the problem of re-enrollment when any stored templates are compromised. This approach can be extended to develop a template protection algorithm for other biometrics such as fingerprint and iris.

In this chapter, a face template protection method described on how the adversarial perturbations at the input data stage could be used as helper data and how the templates are secured using a deep neural network-based mapping. In Chapter 5, template protection mechanism that secures raw biometric data by preventing invertibility and enabling cancelability without affecting recognition performance is presented.

Security of biometric templates is crucial in protecting the biometric data from attacks. Attacks on the biometric data such as feature embeddings and raw biometric data could compromise overall security of the biometric systems. We address the issue of security facial feature embedding in the next chapter where we describe a cancelable non-invertible transform method that can prevent estimation of biometric data from the biometric codes.

Chapter 5

Face descriptor protection

In many applications today, data are drawn from a high-dimensional feature space, where the dimension d is incredibly high. A problem with high dimensional data is that many algorithms that extract higher order information (clustering, nearest-neighbors, etc.) are severely impacted by high dimensionality. Transforming data into a low-dimensional space, without changing pairwise distances significantly, called low-distortion embedding of the data. While dimensionality reduction techniques like PCA, LDA, and LLE can be used to get around this issue, many of these methods will fall short in situations where the subspace structure of the data needs to be retained, especially when creating secure and discriminating biometric templates. A procedure for finding a k-dimensional($k \ll d$) subspace, which satisfies the distance-preserving property is desired. Due to its low computing cost and the guarantees it offers, Random projections has significantly increased in favor recently The Johnson-Lindenstrauss (JL) lemma [131], is the foundation of different uses of Random projections for dimensionality reduction. In particular, it has been demonstrated that, under specific circumstances, Random Projections preserves linear separability and manifold structure, when the data is linearly separable and lies on a low dimensional compact manifold respectively. To address the security and privacy issues mentioned in Section 2.2, there is a need to generate renewable and revocable biometric templates.

This chapter is organized as follows. Section 5.1 provides introduction to Random Projections, and how these projections are computed without compromising on the inter-class margins. Section 5.2 provides the details of the proposed feature embedding protection method. In Section 5.3.1, the results and security analysis of the proposed method are described. Section 5.4 provides the conclusion of the chapter.

5.1 Random projections

In addition to the DNN based template protection, Random projections provide additional layer of non-invertibility at feature embedding level.In this section, the non-invertibility of

Random Projections (RP) for biometric data protection is studied. As shown in Figure 2.9, reconstruction attacks on feature embedding can estimate the facial features, which can compromise on the security of biometric data. In this thesis, Random projections for improving the non-invertibility of the face template protection mechanism are used as an additional mechanism to DNN-based template protection methods described in earlier Section 4.1. Unlike principal component analysis where the principal components depend on the input data, with Random Projections there is very minimal correlation between the input and output data. Also, reconstructed input of the PCA is much similar to the inputs used, but in the case of random projection, the reconstructed data is not close to the input as it is a lossy process. Random projections prevent reconstruction of feature vectors from templates, as shown in Figure 5.1 and 5.2.

The following justifies the usage of the Random Projections method:

1. Directions of projections are independent of the data, and Random Projections do not require all the face data at once, unlike PCA.

2. The Random Projections matrix offers additional security because it is unrelated to the user and his/her face data

3. Random projections make the 4096-dimensional facial feature vector less redundant, making it possible to transfer the reduced dimensionality feature vector to a binary code more accurately and robustly. It also functions as a cancellable transform.

4. The user-assigned Random Projections matrix can be reassigned in the when it is compromised.

5. It also satisfies a optimal biometric template protection scheme's revocability and diversity requirements.

6. Each enrolled user may have their Random Projections matrix revoked and then reassigned by an application.

In this scheme, along with a random binary code of length K, each enrolling user assigned a Random Projections matrix during the enrollment step. This Random Projections matrix generation process can use helper data generated from the user biometric data during enrollment process. This avoids remembering seed to generate Random Projections. The original d-dimensional data are projected on to a k-dimensional ($k << d$) subspace passing through the origin using the Random Projections matrix of dimension kxd. The Random Projections matrix's columns are all of the same length. These matrices are not visible to the user, are used internally, and are securely saved as auxiliary data. The chosen number for k is 1599, which is obtained empirically and the value of d is 4096. The binary code is randomly generated with high entropy. To make brute force attacks computationally infeasible, binary codes with length $K = 256$ bits are used. Neither the user nor the user's face images have any relationship to the binary code generation process. These binary codes are used internally for training deep mapping network during enrollment as in section 4.2. After the enrollement, the binary codes are neither revealed to the user nor stored without protection. The user's protected face template is represented by the cryptographic hash (SHA3) of the binary code assigned to that user, and is stored in the database for future matching purposes.

5.2 Face data protection using random projections

A cancelable biometric system must support a huge number of users i.e. the algorithms that discriminates people based on the biometric should be able to extract discriminative features over a large set, and when a cancelable template is compromised, it shall allow re-enrollment with a new set of cancelable templates. The proposed approach [4] supports a large number of biometric users, and the application can issue a new binary codes and a new projection matrices to its currently enrolled users when a protected template or the random projection matrix is compromised and can enroll new users as well. The revocability and diversity requirements of a perfect biometric template protection strategy are thus addressed by this approach.

Generating a cancelable template, starts with obtaining face embedding. The process for obtaining a face embedding is similar to that one described in Section 4.2. The face embedding computed by the VGGFace network is of 4096 dimension, and the dimensionality of this 4096-dimensional face embedding is reduced using random projection. During both enrollment and verification phases, 4096-dimensional face embeddings of the user's face image is projected on to a Random Projections matrix of size 1599x4096 assigned to the user resulting in 1599-dimensional vector post the projection.

During the enrollment phase, a series of fully connected layers (fc8-fc12) are employed to reduce intra-user variances to acquire a reliable mapping of the reduced dimensionality feature vector to the given binary code of length K-bits. For effective mapping, the number of neurons in each fully connected layer are gradually decreased to minimize the loss. Since many bits of the network output are '1's, the final layer of the network uses the sigmoid activation function to learn this mapping. Binary cross entropy (BCE) is used as the loss function for training the fully connected layers, freezing the other layers. Hash (SHA3-512) of the K-bit binary code is the template used for matching against the database templates.

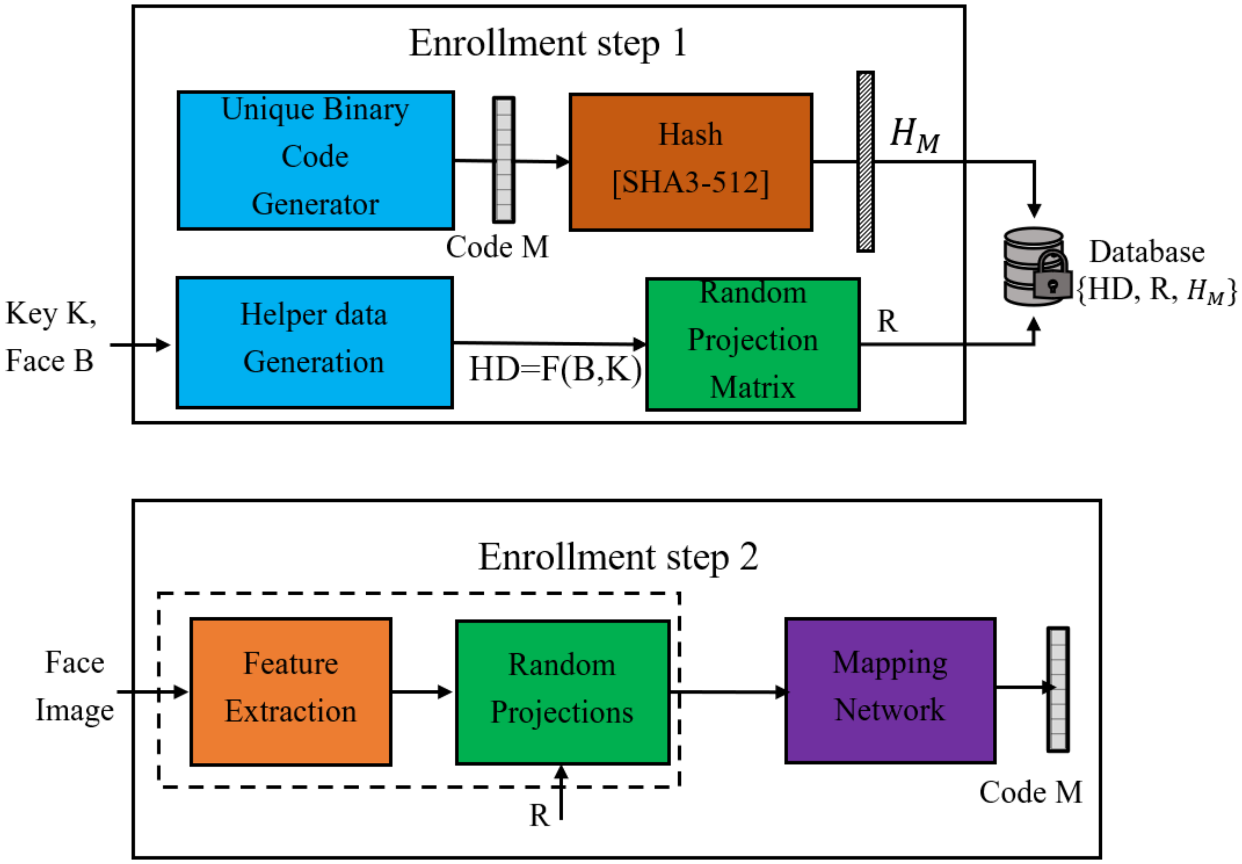


Figure 5.1: Enrollment process of the proposed face data protection method [4].

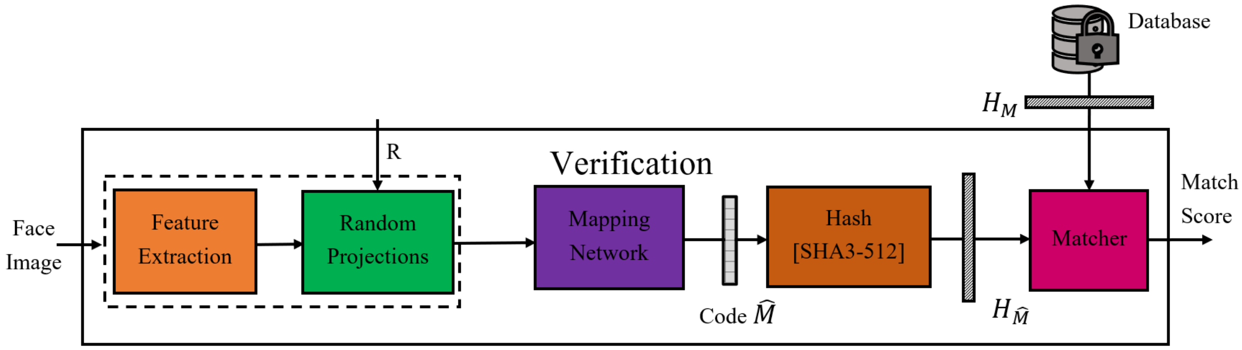


Figure 5.2: Verification process of the proposed face data protection method [4].

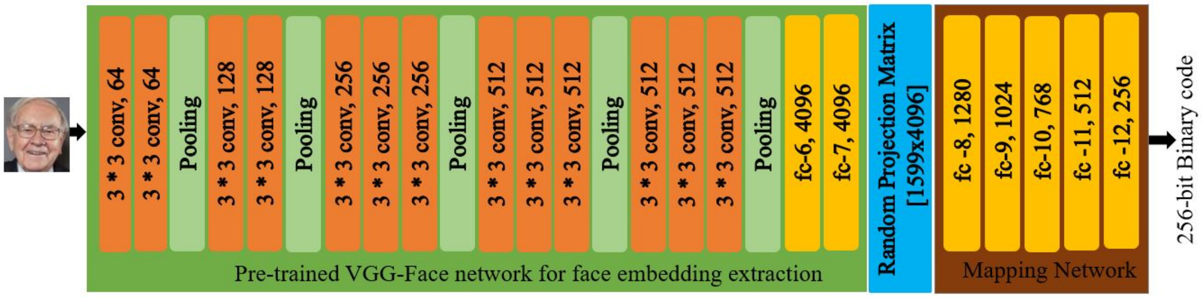


Figure 5.3: DNN Architecture of the face data protection method with *Random Projections*

Figure 5.3 shows the DNN architecture with *Random Projections* to map the face images to a 256 bit binary code template. In this, the input face feature embedding is subjected to dimensionality reduction using Random Projections. To reduce the dimensionality, the Random Projections matrix of size (1599x4096) is assigned to the user. The remaining steps of the process are similar to those that were covered in the earlier Section 4.1. Dropout, with 0.25 probability of discarding hidden activations applied, to reduce over-fitting. This network of fully connected layers minimizes the intra-user variations.

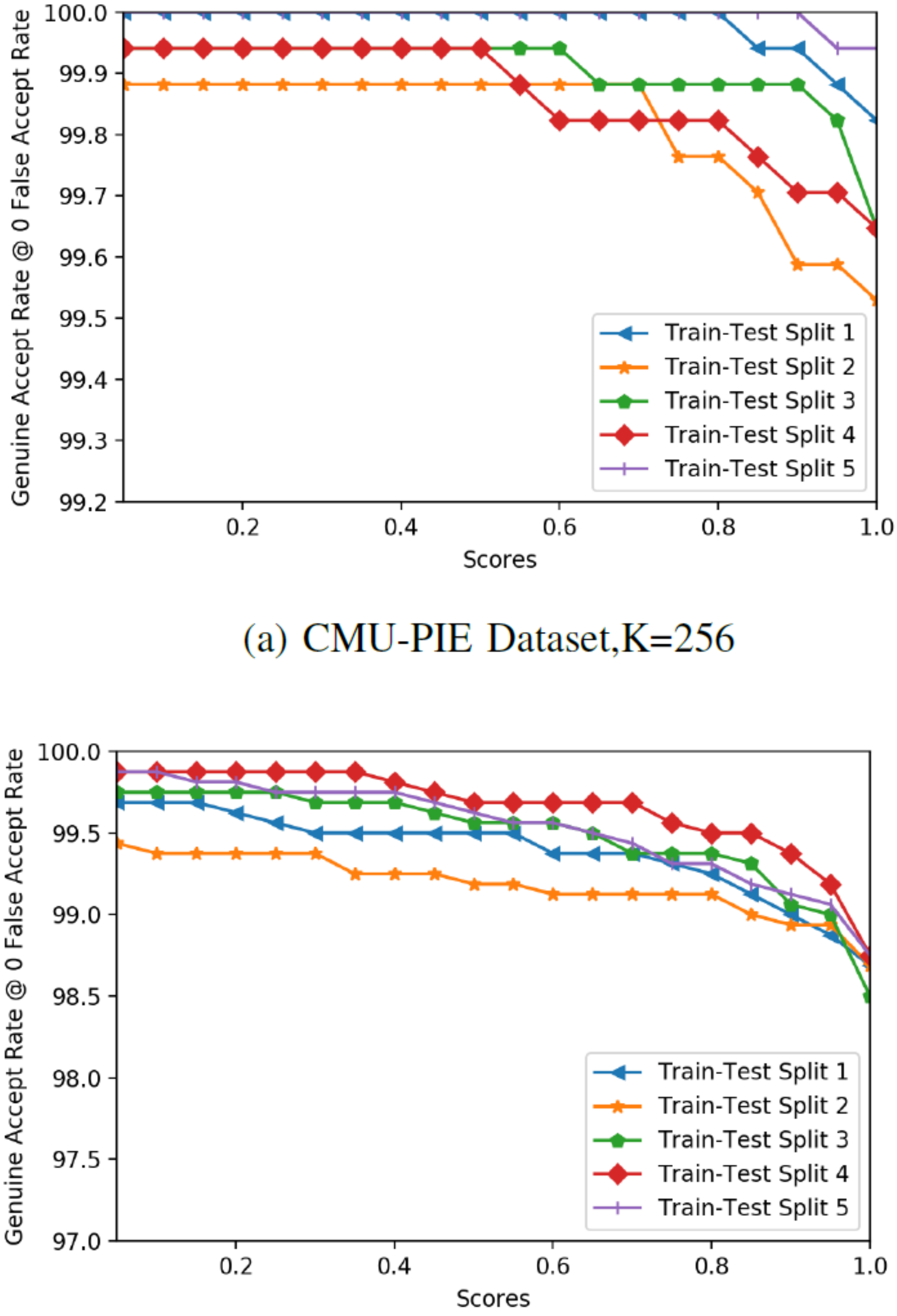
Table 5.1: Face verification performance comparison on PIE and FEI datasets

Dataset	**Enrollment Type**	**K**	**GAR@0%FAR**	**EER**
PIE	One shot	256	99.95% @ 0.04%	0.02%
	Multi-Shot	256	99.98% @ 0.02%	0.01%
FEI	One shot	256	99.73% @ 0.16%	0.14%
	Multi-Shot	256	99.84% @ 0.15%	0.08%

5.3 Results and discussion

The face recognition performance of the proposed method [4] with multi-shot enrollment is shown in Figure 5.4.

The experimental results for *Random Projections* based template protection are shown in Table 5.1 and Table 5.2. From the simulation results, notably the proposed method exhibits zero FAR values even at very low matching scores but also very low FRR values (or high GAR values) even at matching scores as high as 1. For instance, with the CMU-PIE,dataset, with multi-shot enrollment, the proposed method achieves 99.98%GAR@0%FAR at a matching score of 1 for K = 256. This is in sharp contrast to [90] with a multi-shot enrollment of users.



(a) CMU-PIE Dataset,K=256

(b) FEI Dataset,K=256

Figure 5.4: Face matching performance of Random Projections biometric protection method with multi-shot enrollment; Genuine Accept Rate (GAR) @ 0 False Accept Rate (FAR) with respect to matching score for CMU-PIE and FEI datasets

Table 5.2: Face verification performance comparison on CMU-PIE dataset

Method	Enrollment	K	GAR@FAR	EER
Hybrid method [72]	Multi-shot	210	90.61%@1%FAR	6.81%
BDA [121]	Multi-Shot	76	96.38%@1%FAR	-
MEB-Encoding [90]	Multi-Shot	256	93.22%@0%FAR	1.39%
Our method	Multi-Shot	256	99.98%@0%FAR	0.01%

5.3.1 Security analysis

In this face data protection mechanism, non-invertibility of the template to prevent reconstruction of raw biometric or the biometric features is improved through Random Projections, as projecting the feature embeddings on to a Random Projections matrix is equivalent to many-to-one mapping. Hence, it is difficult to arrive at the input of the Random Projections matrix from its output as these projections are lossy. In addition to the security features displayed by the DNN based template protection approach, Random Projections makes it even more difficult for the attacker to estimate biometrics data from the lower dimensional feature vectors or face templates due to its inherent properties. In the case of attacker having access to the stolen protected face template, represented by the cryptographic hash of the unique binary code assigned to the user only, the attacker would not be able to extract any biometric information but the binary code. As mentioned earlier, the brute force attack to reveal the binary code is computationally infeasible. In the case where the attacker has access to both the stolen protected face template and the DNN model, the attacker would perform attacks (like a dictionary attack using a large set of face images) to exploit the FAR of the system. However, the near zero FAR values of the proposed method indicate that it is resistant to such attacks.

5.4 Conclusion

In this chapter, a mechanism to prevent extracting face data i.e. the feature embeddings from the template and that can provide cancelability is proposed. Random projections are fast at reducing dimensionality, unlike PCA, which requires to hold the entire dataset in the memory requiring a higher amount of computing resources. Random projections perform well at higher dimensions too. A major advantage of Random projections is that it regenerates a new projection matrix whenever a face template is compromised or a when a new user is enrolled. As the face embeddings are projected into a lower dimensional subspace, and it is *NP* hard to recover the input feature embedding from post the random projection. The proposed approach provides about 99.98% accuracy. The proposed approach achieves zero FAR values even at very low matching scores and very low FRR values (or high GAR values).

It is to be noted that zero FAR achieved is a result of limited dataset size in the current experimental setting.

Securing biometric data in a centralized setting where the enrollment happens at once on a central computing node is itself a large problem to solve. However, due to the privacy regulations proposed by several countries, the cost of compromising the personal data is prohibitive. Hence, researchers adopted federated learning as a mechanism to offload the responsibility of data storage by learning the models on the client devices or nodes themselves. In the next chapter 6 we will explore what is *Federate Learning* and how it can be used in biometrics systems and the corresponding security implications due to the heterogeneous learning process. Finally, we give pointers to the issues to be addressed in Federated Face recognition.

Chapter 6

Secure federated learning

Traditional centralized machine learning employs a data pipeline that makes use of a central server train it on the data collected from many and host the trained model and to make predictions. The drawback of this architecture is that it compromises privacy by gathering the required data from local devices and sensors to a central location for training the models. Federated learning (FL), in contrast, is a method that downloads the partially trained model on to the client device and updates the model using local data on the device, safeguarding anonymity. The devices then transmit these locally learned models back to the main server for aggregation, and then a single consolidated and improved global model is obtained. Federated averaging (FedAvg) [132] is a popular aggregation method for aggregating locally computed updates. This chapter is organized as follows. In Section 6.1, introduction to federated learning (FL) is provided. In Section 6.2, a brief overview of the security issues and attacks on FL is provided. Section 6.3 describes a study on comparing federated learning and multiparty computation is provided. In Section 6.4, application of federated learning to protect the privacy of the users in recommender systems is provided. Section 6.5 briefly explores the literature in federated face recognition and some attacks on these systems. Conclusion of the chapter is provided in Section 6.6.

6.1 Introduction

Federated learning permits more intelligent models, lessens latency, uses less power, and maintains privacy. In FL, the server initially pre-trains the global model using central training data that is made publicly available. Following that, it chooses K clients from among N clients to distribute the global model's parameters depending on their resource information [133]. Additionally, each chosen client updates the global model by training it with unique local data. The server then compiles all the model updates and to arrive at the global model, as seen in Figure 6.1. Federated Average (FedAvg) [132] is commonly used method to ag-

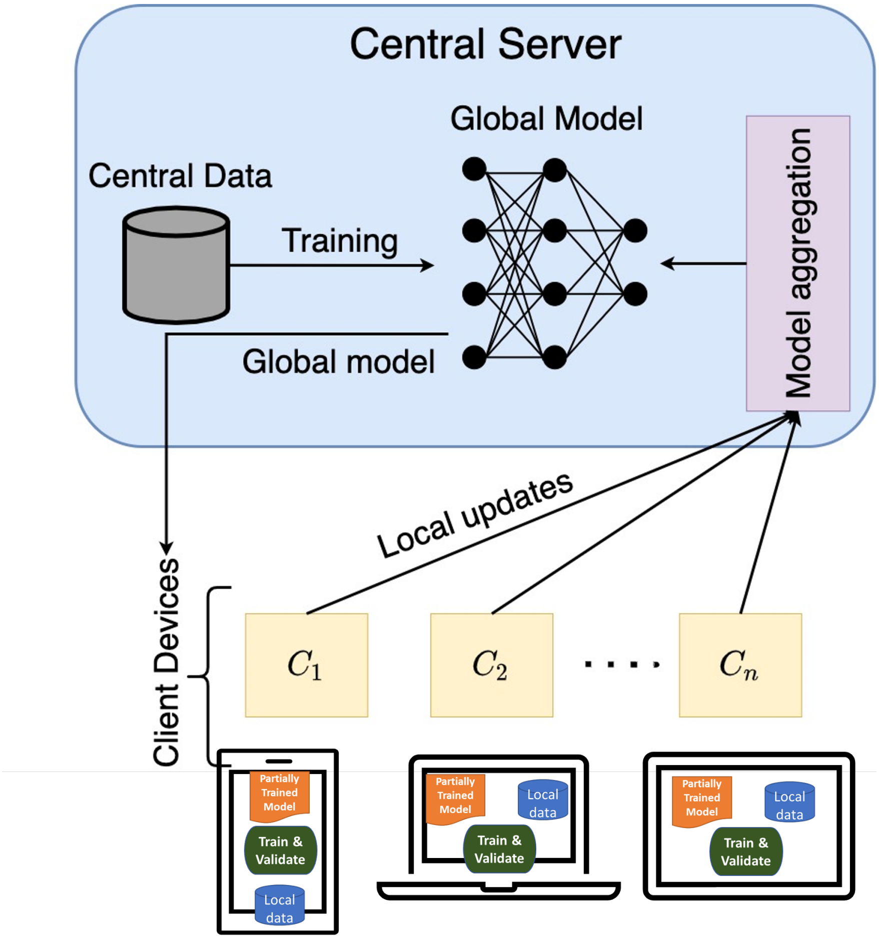
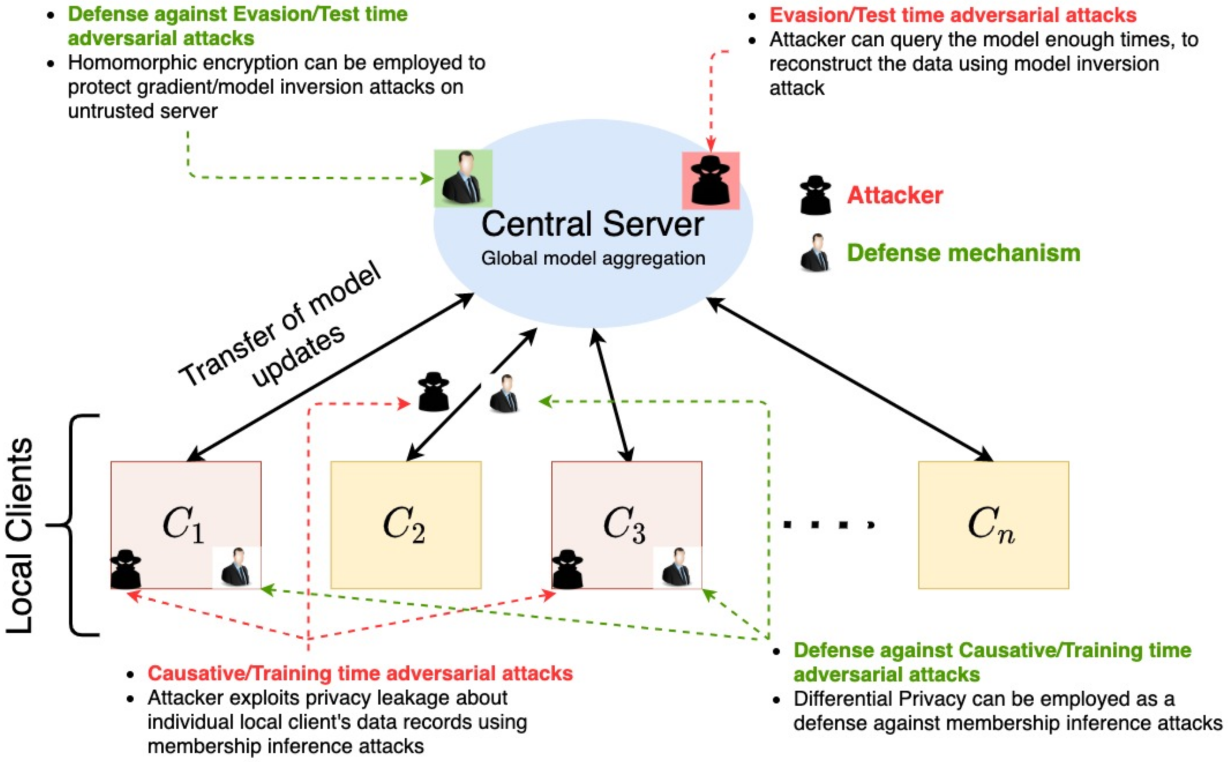


Figure 6.1: A typical federated learning architecture

gregate locally computed updates to arrive at a global model. This process is repeated until the global model reaches convergence.

Because of the distributed nature of federated learning, data scientists can successfully train a global model using model weights received from decentralized devices, as illustrated in Figure 6.1. This suggests that federated learning does not involve the interchange of any private data, although the same model is globally shared across client devices. By avoiding transfer of the data to a server, federated learning protects the confidentiality and privacy of the data in contrast to traditional machine learning, which uses a centralized data repository.

Despite all aforementioned advantages of federated learning, there are several security and privacy challenges [134] that need to be addressed: (i) Trade-off between efficiency and privacy, (ii) Privacy risks associate with the server, (iii) Selection of clients among registered



Defense against Evasion/Test time adversarial attacks
Homomorphic encryption can be employed to protect gradient/model inversion attacks on untrusted server
Evasion/Test time adversarial attacks
Attacker can query the model enough times, to reconstruct the data using model inversion attack
Central Server
Global model aggregation
Attacker
Defense mechanism
Transfer of model updates
Local Clients
C_1
C_2
C_3
C_n
Causative/Training time adversarial attacks
Attacker exploits privacy leakage about individual local client's data records using membership inference attacks
Defense against Causative/Training time adversarial attacks
Differential Privacy can be employed as a defense against membership inference attacks

Figure 6.2: Overview of privacy and security threats of federated learning. There are two types of attacks: causative (training time) evasion (test time) attacks. Membership inference attacks and model/gradient inversion attacks exploit privacy leakage.

clients, (iv) System and Statistical heterogeneity, (v) Poisoning attacks (vi) Aggregator turning malicious.

6.2 Security of federated learning systems

Multi-agent collaboration in Federated Learning is an exciting development in deep neural networks (DNN) [135]. Federated learning allows offloading or distribute a computationally demanding training task to its clients. The clients (or agents) operate like a conventional distributed system by sending frequent changes to the local model to the central server. Following the gathering of these local updates, the server changes the overall shared model and relays the weights to the clients. As discussed in earlier chapters, adversarial attacks compromise the security and performance of the deep learning models. An overview of the possible attacks on a federated learning system is illustrated in Figure 6.2. It is highly impossible to verify the authenticity and trustworthiness of a client's update. Confidence decrease, mis-classification, and targeted mis-classification are the objectives of adversarial

attacks on neural networks. The attacks can be separated into evasion/exploratory attacks and poisoning/causative attacks (i.e., attacks during training time) (i.e., test time attacks).
Threat models: In a Federated Learning system, either the aggregator or the client can be malicious and try to subverting the system.
Malicious server: Ideally the server aggregating the updates from individual clients/agents is expected to be a genuine or a trusted third party. However, it may not be the case when the application in question is sensitive. A malicious server can use the updates received from one or more clients and try to inferring the data from this. Also, a malicious server may try sending a manipulated model to the clients so that the update it receives in that round could represent the sensitive data well and the job of inferring from the updates becomes easier. This process may be repeated several times to extract good quality data.
Malicious clients: While the clients do not have a direct relationship with other clients's data, the aggregated model could well provide some clues about them. A malicious client can also add poisonous samples to the training process to subvert the learning process leading to a substandard model. Also, some clients could also collude to divert the learning process and introduce back doors to the model to be exploited later.

6.2.1 Attacks on federated Learning

As shown in the Table 6.1, broadly there are three different types of attacks that affect the federate learning systems, the most (i) membership inference attacks. (ii) model inversion attacks. (iii) adversarial attacks.

Membership inference and model inversion attacks: Federated Learning technique was proposed to secure deep learning models' data confidentiality and privacy, and only model parameter sharing between a central server and linked client devices is permitted in FL. While objective of this arrangement is to ensure the privacy of the user's data, there is still a risk of leaking the data. Model inversion attacks [136, 137] and membership inference attacks [138–140] exploit these model parameters leading to leakage of private information, posing a threat to data privacy. Hence, the model querying capability is a serious weakness that may be further fixed with the use of differential privacy and secure aggregation.

Adversarial Attacks: In this, the attacker intends to corrupt the entire federated learning process by compromising the benign learning nodes with poisonous data in the form of altered features or false labels [141, 142]. The attacker can send malicious updates to the central server and control the learning process. These poisoned updates when used in the aggregation, cause drastic degradation of the overall performance. Also, because of DNN's egregious mis-predictions, even with slight perturbations, there has been a great deal of research interest in creating novel adversarial perturbations [99] and creating defense mechanisms for it [143]. Researchers have started looking at privacy-preserving Federated Learning from an adversarial setting perspective because of this interest. Table 6.1 contains a list of

Table 6.1: Related works on FL privacy & security threats

FL attack	**Model(s)**	**Dataset(s)**	**Attacker**	**Attack purpose**
Information leakage in FL [137]	GAN, CNN	MNIST, AT&T dataset of faces	Client	Membership Inference Attack
Deep Leakage from Gradients [138]	CNN	MNIST, CIFAR-10, SVHN and LFW	Client	Membership Inference Attack
Model poisoning attack using boosting [141]	CNN	Fashion-MNIST, UCI Adult Census dataset	Client	Poisoning local model
Model poisoning attack [142]	CNN	Fashion-MNIST	Client	Poisoning local model
Comprehensive Privacy Analysis of Deep Learning [145]	ResNet, DenseNet	Texas100, Purchase100, CIFAR100	Client, Server	Membership Inference Attack
Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning [136]	CNN	AT&T , MNIST, CIFAR100	Server	Model Inversion Attack
Inverting Gradients - How easy is it to break privacy in federated learning? [139]	CNN	MNIST, CIFAR100	Server	Membership Inference Attack
Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning [137]	CNN	MNIST, AT&T	Client	Model Inversion Attack

associated works. Given that the client's local data and training process are inaccessible by the global server [144], there is a significant likelihood of poisoning attacks in FL.

As the data in face recognition systems are sensitive, preventing these attacks on FL based face recognition systems is essential. An attacker through *Poisoning attacks* can manipulate the global model to behave in a targeted manner, either to produce incorrect predictions or to corrupt the global model. Similarly membership inference attacks try to estimate if a given sample exists in the database that is used in the training process, risking the privacy of the corresponding individual. Model inversion attacks in face recognition(see Section 2.2) can estimate the face data of the other clients from the model the client receives from the server. We plan to study the effects of these attacks on the federated face recognition system.

Figure 6.3 show a high level security architecture for federated learning. In this, Differential privacy (DP) is used to protect the privacy of the client updates. Multi-party computing is used to protect privacy of the individual client updates by performing a central computing task without knowing each other updates. Homomorphic encryption (HE) inference enables clients and the aggregator performing the tasks in an encrypted domain without exposing either data or model or model updates. Public Key Encryption(PKE) is used to securely transmit the data in both the directions between the server and the client.

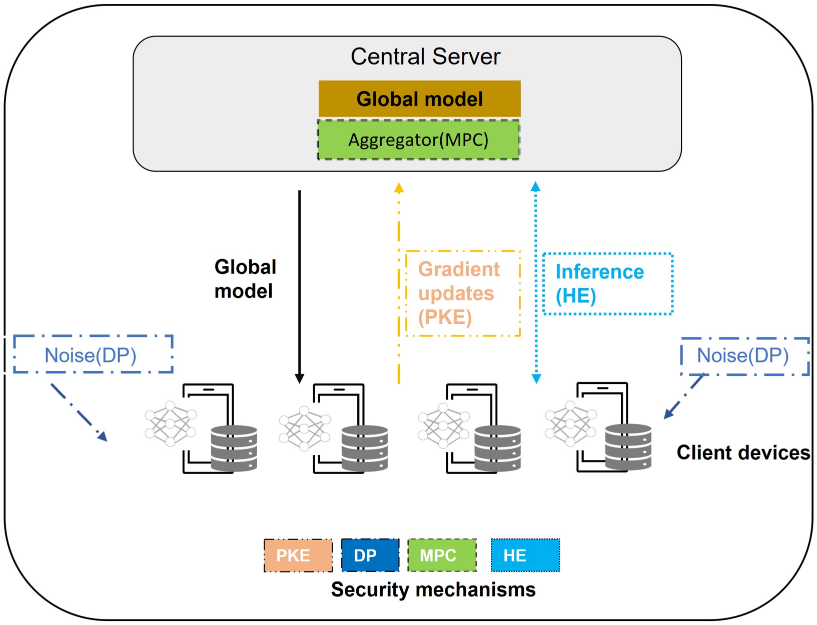


Figure 6.3: A high level security architecture for federated learning systems

These techniques can be appropriately combined to achieve the goal of security and privacy in federated learning systems.

6.3 Federated learning and multiparty computation

Multiparty computation (MPC) allows multiple parties to compute a common function without having to share their inputs publicly. Instead, each party generate something called as shares for their inputs and distributes a subset of this shares to the other parties such that, no other party can recreate the actual input as shown in Figure 6.4. This is done for every input and by each client, hence the communication cost is high for MPC protocols. In our experiment, we have considered 3 parties, where each party has a subset of MNIST dataset. Before the training begins, each party generates shares of their inputs and distributes the subset of shares with other parties. During the training phase, the addition operations can be done locally by each client using the local inputs. However, for multiplication, one round of communication is required to complete the multiplication. Hence, the communication cost increases with every multiplication in the computation. All the clients have were created in the same standalone system. Each pair of clients has an SSL-key shared for secure communication. Experimental parameters include (i) number of clients - 3 (ii) Dataset - MNIST (iii) Split - IID (homogeneous) (iv) Network - 3-layer fully connected (v) Epochs - 10

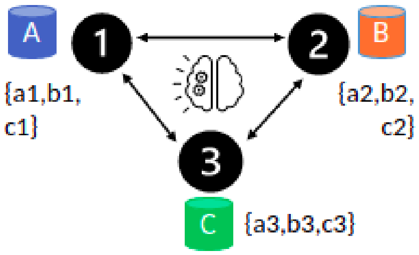


Figure 6.4: A typical multi-party computation process

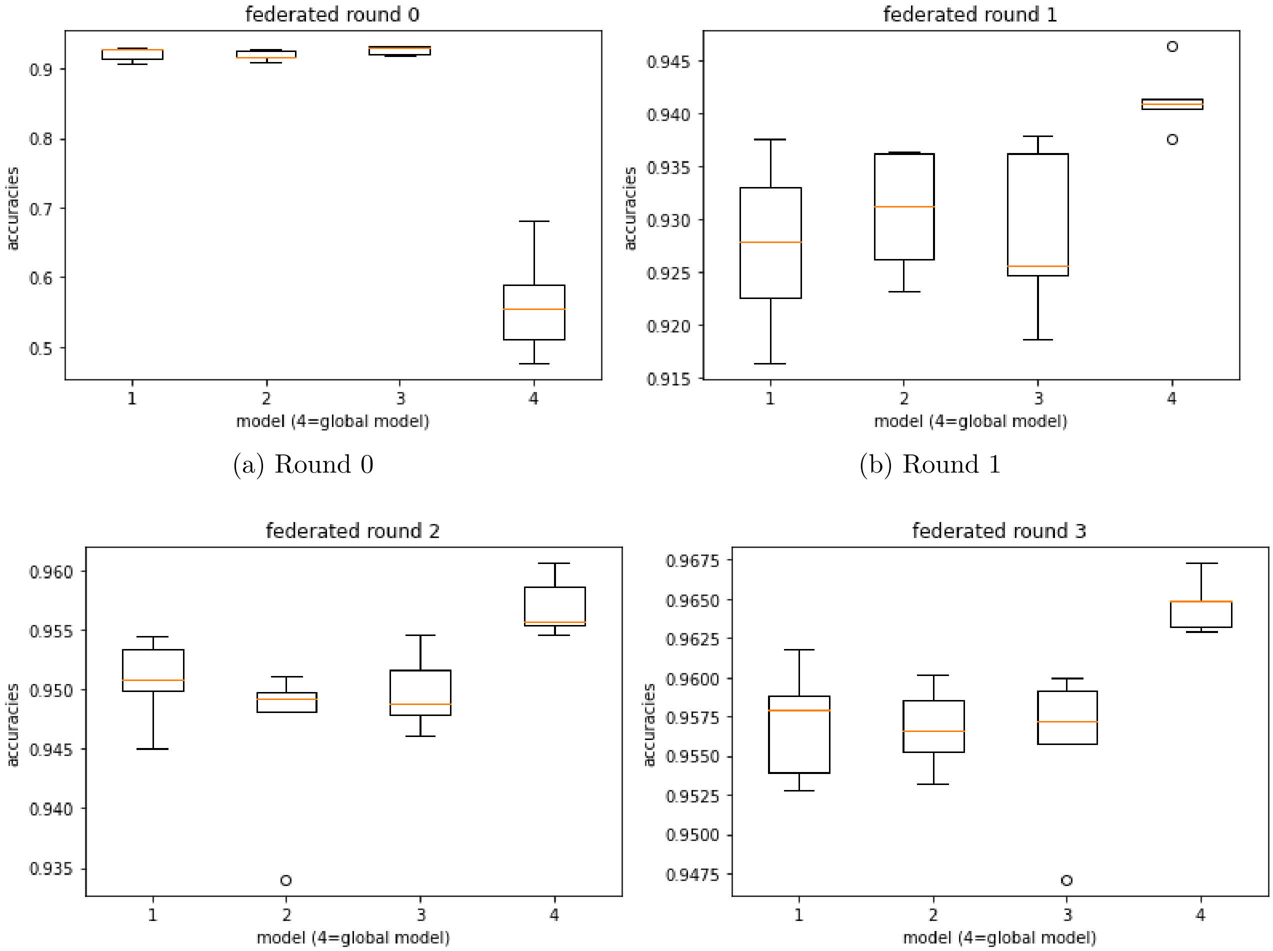
We conducted few experiments on the MNIST dataset for comparing and contrasting the approaches of private learning, which are federated learning (FL) and multi-party computations (MPC) with its centralized counterpart. In these experiments, MNIST dataset is used with homogeneous data split, and for centralized DL, we used the dataset with a typical train test split. A 3-layer Fully connected network is used for the classification purposes. In federated learning and MPC based training, we used the dataset from the centralized data to shard it into three stratified random partitions with their individual training and testing subsets. For fair comparison of results, we kept the number of times the parameters of the network updates the same across FL, MPC and centralized model experiments, by varying the number of federated rounds, epochs and mini-batch size. The network architecture remained the same across the experiments and we did 5-fold cross-validation to check variance of the results. The results we got were intuitive.

Table 6.2: Comparison of centralized Learning, federated Learning, and Muliti-party computations

Approach	**Accuracy**	**Train time**	**Threat Model**	**Communication Cost**
Centralized Model	97.67	15 seconds	Honest server	0
Federated learning (FL)	94.13	3 minutes	Curious Clients, Honest server	Model_size * num_parties * num_fl_rounds
Multi-Party computation	97.67	2.25 hours	Curious clients, (No server)	Dataset_size * num_parties^2 * num_muls

As shown in Figure 6.5 the 0^{th} federated round, the three local models (1, 2, 3 in graph) exhibited higher accuracies compared to the global model (4). Once this global model is

!htbp



(a) Round 0

(b) Round 1

(c) Round 2

(d) Round 3

Figure 6.5: Federated learning: performance study on MNIST

sent to the clients again, the clients are now holding a global model that was once also contributed by other clients as well. Hence we were expecting the accuracy of the global model to be higher than the initial global model. Moreover, we were also expecting the benefits of federated learning to reflect in the results of the first federated round (note that we started with 0). The benefit being that the global model will exhibit better accuracy on the global test set compared to the other local models alone. As can be observed, the individual local models gained a lot in accuracy compared to their accuracy in the 0^{th} round, but what's more significant is the improvement in the accuracy of the global model. Not only it is much better than the accuracy of round-0, but also the global model that we got in round-1 is having a higher accuracy against the global test set than any local model. This is why we need collaborative learning in a situation where an individual client has less data. It is always better to combine efforts with multiple clients with similar datasets. Since a direct approach where the datasets are concatenated is restricted, despite being ideal, federated learning is the best bet here. In the next round, we can still see the trend being followed where the global model is having a much higher test accuracy against a global test set. Following this, we can see a steady increase in the accuracies of both the local and global models. Vanilla federated learning is unideal when the server is compromised or if the server is curious. Attacks such as model inversion can be employed by a curious server to infer details about the client data using the local models. This is where we need secure aggregation and FL algorithms that are robust to more complex threat models involving a curious or malicious server.

Overall comparison of FL, MP and centralized settings is shown in Table 6.2. It can be seen from the table, that MPC matching the accuracy of the centralized setting but has higher communication and computational costs. Where as for federated learning, there is a little reduction in the performance while the computational cost is relatively very low.

6.4 Studies on federated recommender system

Online streaming services and shopping services make extensive use of recommendation algorithms to direct users to pertinent things, which would be exceedingly challenging to perform without them. To comprehend a user's long-term preferences, both the collaborative and content-based filtering methods often rely on past user-item interactions. To provide more pertinent recommendations, a combination of a user's more long-term preferences and current or short-term preferences should be taken into account. Thus, session-based recommendation algorithms that strongly consider the user's most recent interactions rather than just her prior preferences were developed. Priyanka et al. proposed a session-based recommendation approach, Normalized Item and Session Representation (NISER) [146] was proposed for handling popularity bias. It should be emphasized that all recommender systems utilizing the aforementioned algorithms were created using user-provided private data. This data concentration creates grave security and privacy concerns. In addition to being more susceptible to hacking and other types of data theft, centralized data also lends power to whoever has the access to the server. To provide users better, more relevant suggestions while also protecting the confidentiality of their data, we adapt the NISER [146] algorithm to the federated setting.

As in any recommender system, the K items with the highest scores constitute the top-K recommendation list. We update the global model $\mathcal{F}_g$ using popular federated averaging [132].

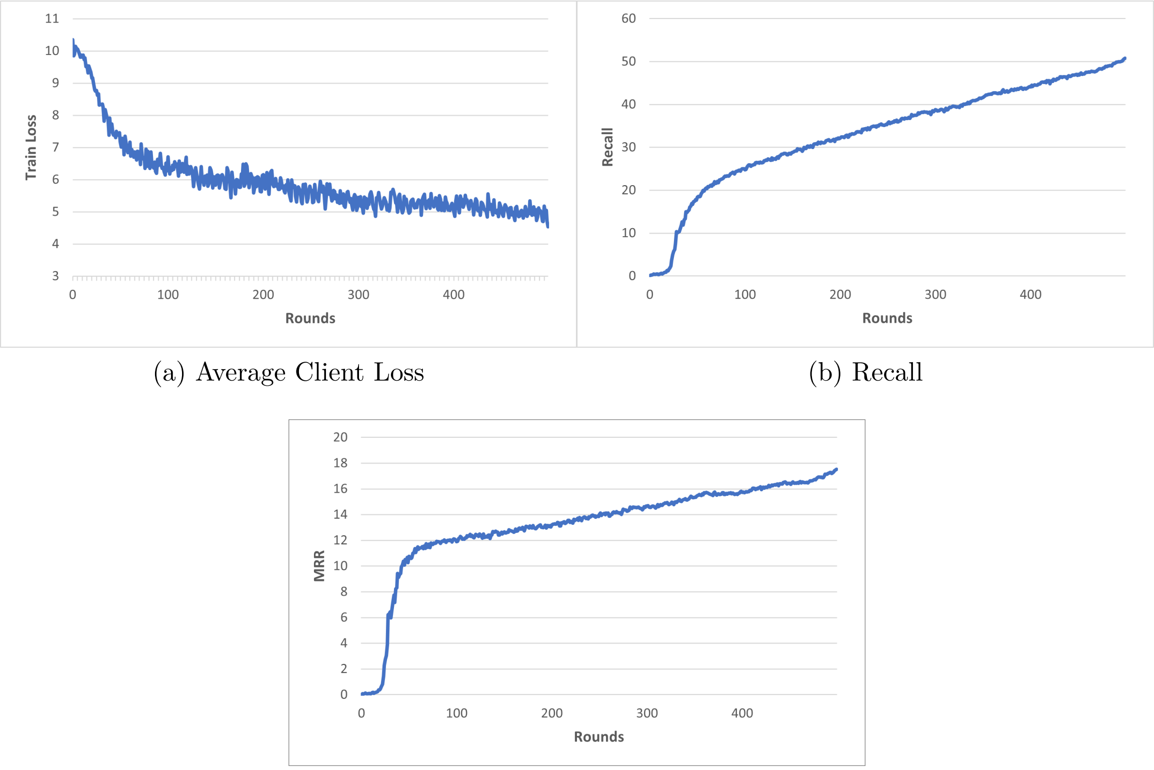
Initial global model($\mathcal{F}_g = \theta_0$) is initialized with random values and shared with a randomly chosen $N \cdot K$ clients for the local training. At each client, the sessions are recorded and item embeddings ($\mathcal{I}_s$) for all sessions (S_k). Locally, these item embeddings are updated using a graph neural network from the nodes and vertices obtained from the $\mathcal{I}_f$. This gives a combined embedding of normalized item representation, session representation, and position embedding. This updated item embedding is then used to obtain the relevance score for net clicked item i_k computed as,

$$\hat{y}_k = \frac{exp(\sigma \tilde{i}_k^T \tilde{s})}{\sum_{j=1}^{m} exp(\sigma \tilde{i}_j^T \tilde{s})} \tag{6.1}$$

where m is the total number of items present. Local training of the model and subsequent updates of the global model by aggregating the model updates will continue to go on until the global model has achieved convergence.

We evaluate the federated model on a publicly available benchmark dataset: Diginetica, transactional data from the CIKM Cup 2016 challenge. This dataset includes 43K items over 700K train sessions and 60859 test sessions. An average session lasts 5.12. We used two evaluation metrics to evaluate the global model: Recall@K and Mean Reciprocal Rank(MRR@K) as in NISER [146] with K=20. The percentage of test cases with the desired item ranking in the top K items is known as Recall@K. MRR@K is the mean of the reciprocal ranks of the desired item in the recommendation list. The item is at the top of the list of recommendations, is indicated by the high MRR score.

The central model was trained with item embeddings of size 100, learning rate of 0.001, Adam optimizer and dropout was set to 0.1. 10% of the train set was used for validation set. For federated model, we simulate the experiments with 100 clients for 500 rounds. At client side, to train the local model we use $epochs = 3$, learning rate is set as 0.001, dropout is 0.1 and SGD optimizer with momentum set to 0.9. The gradients from the clients are aggregated using federated averaging [132] at the server.



(a) Average Client Loss

(b) Recall

(c) Mean Reciprocal Rank

Figure 6.6: Federated NISER model [5] performance: (a) the average client loss, (b) recall@20 and (c) mean reciprocal rank (MRR@20).

Table 6.3: Central vs. federated model: A comparison between central and federated model on Diginetica dataset.

Model	**Recall@20**	**MRR@20**
Central	52.63	18.27
Federated	51.2	17.56

6.4.1 Results and discussion

This section compares the effectiveness of federated suggestions to those made by the central version of the model. We present the average client loss(Figure 6.6a) across rounds and Recall@20 and MRR@20 across 500 rounds is shown in, Figure 6.6b and Figure 6.6c, respectively for the updated global model. As we can see in Table 6.3, recommendations in centralized and federated environments are equivalent with no performance degradation. However, the federated model does not require the data to be present at a server as it gets aggregated from the parameters obtained from the client.

6.5 Federated face recognition

Traditionally face verification models work in a centralized manner. But, due to privacy concerns and regulations organizations using face as a biometric in access control and other services prevented from collecting and storing sensitive private data. Similarly, several law enforcement agencies gather biometric data for criminals, which includes face. These data are used to train models to identify criminals in surveillance videos or live feeds. However, tracking criminals in different jurisdiction is affected as the criminal's data may not be available with them but the laws may prevent movement of sensitive data from one jurisdiction to another. To alleviate both the problems of privacy and regulations several researchers proposed federated face recognition approaches [147–149]. In this federated setting, each client has access to the face images of his/her own and class embeddings are used to learn a shared global model and the central server for aggregates the partial updates received from the clients. However, these class embeddings can be used by the attacker to retrieve sensitive biometric data leading to privacy and security issues.

In [147] a common feature extractor (backbone) and a client specific classifier is used. Each client had a replica of the feature extractor. It was found that locally applying global momentum is better than the server only-momentum approaches. Hence, the global momentum for the feature extractor(backbone) and each classifier is maintained separately. After several local steps, the client sends the local moments (classifier) updates to the server for aggregation. The server first validates the updates for finding better weightings during training to achieve improved model performance. The central aggregator tries to identify optimal updates to be used using a validator, which validates the model updates using private validation data. Similarly, FedFace [148] uses the face images present on various clients to develop an accurate and generalizable facial recognition model, where the face photos kept at each client are a mobile device, including just the owner's face photographs, and each client is a mobile device (one identity per client). In this, both the updates and class embeddings are communicated to the server. Finally, a spreadout regularizer, which takes inspiration from the FedAwS algorithm [150], is used to ensure that the class embeddings are evenly spaced out in the embedding space and do not collapse into a single point. The server optimizes the spreadout regularizer after collecting the class embeddings from all clients in each communication round.

6.6 Conclusion

While the deep learning systems are gradually adopting the federated structure to fulfill the obligation towards the regulations and at the same time opening a new threat surface for the attackers to launch attacks to retrieve sensitive private data. Also, research in the area of federated face recognition is at the nascent stage. Hence a study on the attacks on federated face recognition systems is the need of the hour. We provide a review of some approaches in the literature on federated face recognition. In this chapter, we also, provided a study conducted on a federated recommender system and could achieve performance close to a centralized model. To validate which of the private learning approaches is suitable for face verification, we conducted a study on MNIST data. We also, did a comparison study of federated learning and multiparty computation and showcased the trade-off between privacy, computational cost and communication cost. While MPC is better suitable for strict privacy measures in face recognition systems, due to its prohibitive computational and communication costs it can only be used in high-risk applications. We plan to study a federated learning based face recognition, study backdoor attacks and find mitigation for these attacks.

In the chapters 3, 4, 5, 6 we discussed about the security aspects of facial biometrics. Various aspects of template security, feature embedding security and security against adversarial perturbations and the corresponding proposed methods were studied. In the next chapter, we discuss our work on a facial feature descriptor that can be used to recognize people effectively and require less computational power. The facial feature descriptor with SVM and PLDA classifiers is described. Also, a gender classification approach, with the proposed facial descriptor is described in the next chapter.

Chapter 7

A light weight facial feature descriptor

Face recognition is now being used in automatic login, access control, automatic attention monitoring, visual search, and photo tagging, etc. Facial recognition uses biometrics to map facial features and help verify identity through salient features. This feature set includes the geometry of facial features such as eyes, nose, mouth, and their relationship along with texture and edge information. This then creates, what is called a 'facial signature'. This facial signature is stored as a template, which is compared to a database of known faces. We proposed a novel Local Binary Pattern for face recognition purposes. Several facial descriptors are proposed in the literature but they are computationally intensive and may not be suitable in real applications. In this thesis, we propose a simple to compute a facial feature descriptor for face recognition purposes. The chapter is organized as follows. Section 7.1 describes the proposed LBP operator called cross-local binary patterns (XLBP). In Section 7.2, cross local Radon binary patterns (XLRBP) are described using a spatial pyramid structure. Section 7.3 discusses the face recognition approach using XLRBP descriptor. Section 7.4 provides the details of the proposed Gabor filter based LBP operator called cross local Gabor binary patterns (XLGBP). In Section 7.5, a face verification approach using XLGBP and probabilistic linear discriminant analysis is provided. Section 7.6 discusses a gender detection approach using the proposed cross local binary patterns and its performance results are presented. Section 7.7 provides conclusion of this chapter on LBP-based feature descriptors.

7.1 Cross local binary patterns (XLBP)

Texture representation and analysis is an important area of research for computer vision scientists. Several techniques for discriminating texture patterns exists in the literature. Initial methods are based on signal processing techniques and statistical methods.

The earlier methods are computationally complex and at the same time they are unable to perform enough for use in real world or practical applications. Ojala et al. [29, 55] proposed a method that is discriminative and, and can compute local texture features descriptors called Local binary patterns (LBP). Later several researchers adopted LBP and its extensions for various applications in pattern recognition. We present a discussion of various adaptations of LBP in later chapters.

Local binary patterns are highly discriminative and yet easy to compute. A generic LBP operates a block of image pixels and traverses the entire image to cover all the pixels, with each as the center of the block. The LBP operator compares the neighborhood of each center pixel and the comparison result is represented by bits of a binary number. Ojala et al. proposed the original LBP operator on a 3x3 block in which each 8-neighborhood pixel is thresholded (compared) with the center pixel to produce an 8-bit binary value, as shown in Equation 7.1 and 7.2. Each binary digit is weighted by its position in the binary pattern to arrive a decimal value. So, a total of 256 levels are obtained to represent the relative values around the center pixel within a 3x3 block.

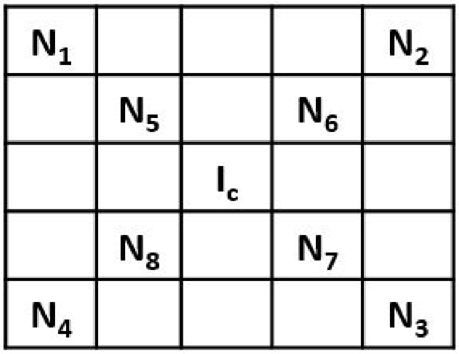
Let the number of neighborhood pixels is P and the radius of the neighborhood is R, then the notation of the LBP operator can be denoted as LBP_R^P. The LBP operator produces 2^P different output values corresponding to the 2^P different binary patterns that can be formed by P neighborhood pixels. The LBP operator can be described as in Figure 7.1. A local binary pattern is called uniform, if it contains a most two bitwise transition from 0 to 1 or vice versa, when the binary string is circular. Decimal form of a binary string for the pixel I_c is calculated by the following equation.

$$LBP_R^P(I_c) = \sum_{n=0}^{P-1} s(I_n - I_c)2^n \tag{7.1}$$

where I_n and I_c are the values of neighbourhood and center pixels respectively. The threshold function is given as

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{7.2}$$

We propose a variant of the LBP operator, which we call Cross Local Binary pattern(XLBP). This is a (P=8, R=2) operator for texture description. This descriptor considers diagonal pixels in four directions, as shown in Figure 7.1. In the case of large illumination variations our approach exhibit a better recognition rate. In this approach, the face image is divided into several sub-parts, and individual feature histograms for each sub-part are computed. Matching these individual LBP feature histograms hierarchically improves the robustness of the algorithm against illumination changes. In the traditional LBP approach, if the radius increases, the number of neighborhood pixels increase which causes the large number of bins in the feature histogram. Also, by increasing the radius, the inner pixel variations get neglected. To overcome these drawbacks, we proposed enhancement to the LBP operator by considering only diagonal pixels in all four directions. Simulations for finding the



$$\begin{aligned}
&\text{XLBP} = \\
&2^0 * s(N_1\text{-}I_c)^2 + \\
&2^1 * s(N_2\text{-}I_c)^2 + \\
&2^2 * s(N_3\text{-}I_c)^2 + \\
&2^3 * s(N_4\text{-}I_c)^2 + \\
&2^4 * s(N_5\text{-}I_c)^2 + \\
&2^5 * s(N_6\text{-}I_c)^2 + \\
&2^6 * s(N_7\text{-}I_c)^2 + \\
&2^7 * s(N_8\text{-}I_c)^2
\end{aligned}$$

Figure 7.1: Cross-local binary patterns with R=2, and P=8

effects of compression on LBP is done by considering the various LBPs and their robustness to various compression formats. From our simulations, we observe that this XLBP preserves its values for various compression levels. This is especially useful when we use video based face recognition.

7.2 Cross local radon binary patterns (XLRBP)

Radon transform is a useful tool to capture shape information due to its inherent properties. The radon transform of an image is the projection of the image along the tracing lines [56] and is based on two parameters. One is distance from the origin s, and another is the angle from the reference axis θ. The Radon transform of an image is computed as

$$R_f\left(s,\theta\right) = \int_l f(x,y)dl \tag{7.3}$$

where all points in the line l satisfy the following equation

$$x\sin(\theta) - y\cos(\theta) - s = 0. \tag{7.4}$$

Therefore, the radon transform of an image can be written as

$$R_f\left(s,\theta\right) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(x,y\right)\delta\left[s - x\sin\left(\theta\right) - y\cos\left(\theta\right)\right]dx\ dy \tag{7.5}$$

where $\delta(.)$ is Dirac delta function. As shown in Figure 7.2 Radon transform is a useful tool to capture shape features due to its inherent properties.

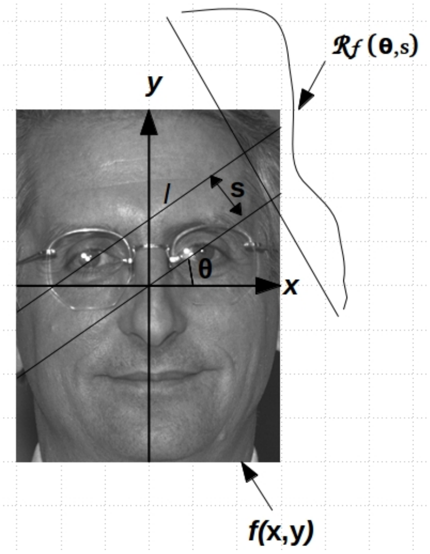


Figure 7.2: Computing Radon transform on a face image

A new face descriptor which is a combination of both the Radon transform and Local Binary Patterns is proposed [7]. Here, the Radon transform captures the shape information while LBP operator models the texture information. After that by using pyramid representation of LRBP, we can capture both the coarse and fine level characteristics of the face image. This descriptor is called as *Cross Local Radon binary Patterns*, and is a combination of *Cross-local binary patterns* and *Radon transform*. By using this method, we can encode the shape and texture of the objects in an image effectively, which is invariant to other modalities such as pose and illumination. Given an image $I(x, y)$, then the Cross local Radon binary transformed image is computed by following equation,

$$XLRBP_R^P\left(I_c^{R_f}\right) = \sum_{n=0}^{P-1} s\left(I_n^{R_f} - I_c^{R_f} - T\right) 2^n, \quad T \geq 0, \tag{7.6}$$

where the superscript R denotes Radon transformed image, and $s(.)$ is thresholding function.

$$s(x) = \begin{cases} 1 & \text{for} \quad x \geq 0 \\ 0 & \text{for} \quad x < 0 \end{cases} \tag{7.7}$$

7.2.1 Spatial pyramid representation

To model the face shape in both fine and coarse levels, XLRBP is computed using the spatial pyramid technique [151] as shown in Figures 7.3 and 7.4. Consider a spatial pyramid with L levels, an image at level $l = 0, \cdots L-1$ is divided into 2^l spatial grids along each axis direction. A spatial pyramid with L levels of an image f is represented by $Sp_f = \{sp_f^l, 0 \leq$

$l \leq L-1\}$. where in each level sp_f^l contains 2^{2l} spatial grids ($sg_f^{l,i}$), and can be represented as $sp_f^l = \{sg_f^{l,i}, 1 \leq i \leq 2^{2l}\}$.

To compute the feature representation for the entire face, each spatial grid $sg_f^{l,i}$ is transformed into Radon space and then XLBP operator is applied. These two steps are jointly described as $XLRBP_R^P(sg_f^{l,i})$. Histogram of each spatial grid $(h_{XLRBP_R^P(sg_f^{l,i})})$ is computed, and then all the 2^{2l} histograms at level l are combined to form a concatenated histogram $H_{f,l}$.

$$H_{f,l} = h_{XLRBP_R^P(sg_f^{l,i})}, 1 \leq i \leq 2^{2l}, 0 \leq l \leq L-1 \tag{7.8}$$

Pyramid representation of XLRBP is represented by $H_{f,L} = \{H_{f,l}, 0 \leq l \leq L-1\}$.

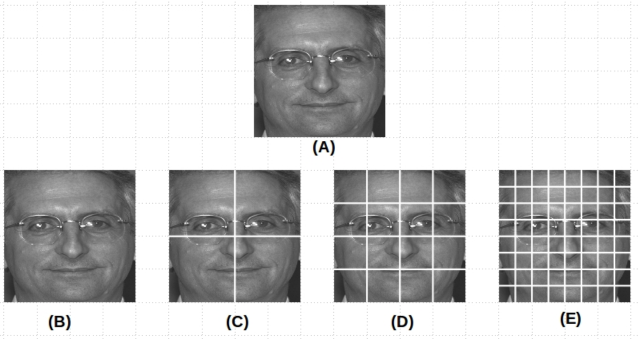


Figure 7.3: Spatial Pyramid structure

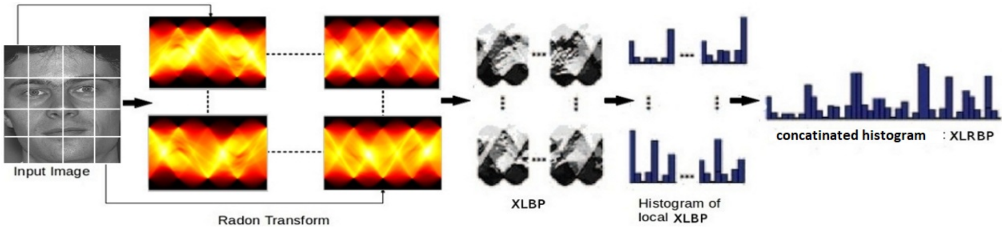


Figure 7.4: XLRBP Feature extraction from a face image [6, 7] (best viewed in color)

7.2.2 Pyramid matching kernel

For matching two face images described by the LBP spatial Pyramid descriptor, Pyramid matching kernel is used to compare two histograms $(LRBPs)$ $H_{p,L}$ and $H_{s,L}$ by using the distance measure given as,

$$d(H_{p,L}, H_{s,L}) = \frac{D(H_{p,0}, H_{s,0})}{2^L} + \sum_{l=1}^{L} \frac{D(H_{p,l}, H_{s,l})}{2^{L-l+1}}. \tag{7.9}$$

Here $D(.)$ is any histogram distance measure. This distance assigns higher weights to the matching image.

7.2.3 Distance measures

Some of the popular distance measures used in pattern recognition for similarity estimation of histograms are as follows:

Correlation distance:

$$d(H_1, H_2) = \frac{\sum_i \left(H_1(i) - \bar{H_1}\right)\left(H_2(i) - \bar{H_2}\right)}{\sqrt{\sum_i \left(H_1(i) - \bar{H_1}\right)^2 \sum_i \left(H_2(i) - \bar{H_2}\right)^2}}. \tag{7.10}$$

Chi-square distance:

$$d(H_1, H_2) = \sum_i \frac{\left(H_1(i) - H_2(i)\right)^2}{H_1(i)}. \tag{7.11}$$

Manhattan distance:

$$d(H_1, H_2) = \sum_i \left|H_1(i) - H_2(i)\right|. \tag{7.12}$$

Bhattacharya distance:

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{\bar{H}_1 \bar{H}_2 N^2} \sum_i \sqrt{H_1(i) H_2(i)}}. \tag{7.13}$$

where

$$\bar{H}_k = \frac{1}{N} \sum_j H_k(j).$$

and N is the number of histogram bins.

7.3 Face recognition using XLRBP

In this method we use Radon transform to capture high-level facial shape information. A variant of LBP (cross-LBP or XLBP) is used to describe the Radon transform output. Histogram of the XLBP values form the feature vector. Using the spatial pyramid technique, both local and global characteristics of facial shape can be modelled. In this process, the facial image is divided into equal parts in both directions at each level, resulting in 4, 16, 64 blocks for levels 1, 2, and 3 respectively. Pyramid matching kernel is used to compare the histograms of train and test faces.

7.3.1 Studies on face recognition performance

In the proposed method [6, 7], Radon transform is applied to capture the high-level facial shape information, and the XLBP operator is used to describe the Radon transform output. Histogram of the XLBP values forms the feature representation for that NxN block of the image. The concatenation of the histograms of the entire image may not give better performance, as the radon transform is applied on fixed size, it may represent global and local feature information. To overcome this limitation, a spatial pyramid technique [151] is used, which can model both local and global characteristics of facial shape. In the spatial pyramid construction process, the facial image is divided into equal parts in both directions at each level, resulting in 4, 16, 64 blocks for levels 1, 2, and 3, respectively.

Non-uniform illumination affects the LBP negatively, as each neighborhood pixel is compared with the center pixel. If the illumination variation is uniform, then the effect on the LBP computation is minimal. Since the LBP is computed on a small block of NxN pixels, variations in the pixel intensities would be minimal and the feature descriptor computed on uniform illumination variations does not get affected. Also, even for partially illuminated faces, the LBP operator can extract better feature representations. For partially illuminated images, using block-based estimation, and matching of histograms can achieve better recognition rates in these cases, as only a few blocks and their LBPs get impacted, but not all.

For the performance evaluation, concatenated XLRBP histograms are computed on the training images and stored as the feature vectors in the database. Given a test face image, XLRBP features are computed similarly and compared with all the stored ones using a distance metric. A thresholding operation is applied on these distances,and if the distance is above a designated threshold the test face is considered to be matched with the corresponding training face and the person is identified.

To evaluate the effectiveness of the proposed XLRBP approach under different lighting conditions, we conducted experiments on FERET and YaleB databases separately. The FERET database contained 900 images of 900 persons. These images were taken at normalized illumination conditions, and all the images were cropped to same size 320 × 256 pixels, such that the face is located at the center of the image. The studies conducted below were on the frontal face. In XLBP, R=2 and P=8 is considered (and in Radon transform we considered $\theta = 0 : 180$, and in spatial pyramid we considered 4 levels ($L = 4$). The Performance of the XLRBP operator is evaluated for face recognition with Bhattacharya distance as a measure on the FERET database. Experiments were conducted on various datasets and feature representations for illustrating the performance of the proposed XLBP operator.

In one experiment, the LRBP approach is compared with EGBM, LBP, and PCA algorithms. The result of this experiment is illustrated in Table 7.1. It can be observed from the table that, LRBP approach provides better results than the other three algorithms. The LRBP approach is efficient in extracting texture information compared with other three algorithms. The closest method to the LRBP method is EGBM in terms of performance.

Table 7.1: Face recognition performance comparison of LRBP against other methods on FERET dataset (Fa, Fb poses)

Algorithm	**Fa**	**Fb**	**Recognition Rate**
PCA	917	917	90.32
EGBM	917	917	94.27
LBP	917	917	88.96
LRBP	917	917	95.31

Also, XLRBP is compared against other LBP methods for face recognition, and the simulation results are shown in Table 7.2. As shown in the table, the proposed face recognition approach with the XLRBP operator performs better than the earlier LBP operators.

Table 7.2: Face recognition performance of XLRBP with Bhattacharya distance on FERET dataset (Fa, Fb poses)

Algorithm	**Fa**	**Fb**	**Recognition Rate**
LBP	900	900	88.96
LRBP	900	900	95.31
XLRBP	900	900	98.89

In another experiment, the XLRBP descriptor method has been evaluated on the YaleB database both with and without partitioning of the face image into half, as faces are symmetric in the frontal pose. In the YaleB dataset, 1520 images of 38 subjects with 45 different illumination variations in a frontal pose were considered. One image per subject is considered for training, and the remaining images were considered for testing. Partitioning the face could be helpful when the faces are partially illuminated. The results of this experiment are given in Table 7.3. Partitioning of the faces will reduce the number of computations by half, but this is suitable only for frontal faces. As shown in the table, the XLRBP approach performs better than the LRBP (plain LBP operator applied on Radon transform of the image). The recognition rate further increased after partitioning the facial image. Cross local Radon binary patterns effectively extract information in Radon space, when compared with local Radon binary patterns. Studies are conducted using cubic SVM as a classifier on the computed XLRBP facial feature descriptor on FEI, and are compared with the standard

LBP operator. The results are given in Table 7.4. A performance improvement of about 4% achieved with the proposed XLRBP operator. Also, studies were conducted by replacing the Radon transform with Gabor filters, and applying the XLBP operator on the Gabor filter response. The results are given in Table 7.5. In this case also the improvement in performance is around 5%.

Table 7.3: Face recognition performance of XLRBP on YALEB dataset

Method	**Train samples**	**Test samples**	**Recognition Rate**
LBP	38	1520	83.86
LRBP	38	1520	88.57
XLRBP	38	1520	90.20
XLRBP(partitions)	38	1520	93.76

In addition to the XLBP operator, a Radon transform based feature representation is proposed for face recognition. From the results, it is evident that the proposed feature representation performs better even for gender detection, compared to the vanilla LBP operators.

Table 7.4: Face verification performance of XLRBP on FEI dataset with Cubic-SVM classifier

Feature	**Training samples**	**Test samples**	**Recognition Rate(%)**
LBP	200	200	93
XLRBP	200	200	97

While the face recognition performance is an important issue, aspects of the security of the face recognition systems need to be taken care of. Hackers can obtain unauthorized access to data and systems, compromising the security and privacy of the individuals.

Table 7.5: Face verification performance with Gabor filter and XLBP using Bhattacharya distance metric on FEI dataset

Feature	**Training samples**	**Test samples**	**Recognition Rate (%)**
LBP	200	200	93
XLRBP	200	200	98

7.4 Cross local Gabor binary patterns (XLGBP)

Spatial frequencies and their orientations are important characteristics of texture in images. Spectral decomposition methods such as Fourier analyze the frequency characteristics of images. Fourier transform has been the most commonly used tool to study a signal's frequency properties. However, it is hard to tell when the signal of a certain frequency happens, i.e., the information about time is lost. Given the fact that the frequency contents of the majority of signals in the real world change with time, it is far more useful to characterize the signal in time and frequency domains simultaneously. A Gabor filter is obtained by modulating a Sinusoid with a Gaussian. Each Gabor filter therefore responds to some frequency, but only in a localized part of the signal. Instead of comparing the signal to complex sinusoidal functions, a natural way of representing a signal in time and frequency simultaneously is to compare the signal with elementary functions that are concentrated in both the time and frequency domains. Let $g(x, y, \theta, \phi)$ be the function defining a Gabor filter centered at the origin, with θ as the spatial frequency and ϕ as the orientation. We can view Gabor filters as

$$g(x, y, \theta, \phi) = exp(-\frac{x^2 + y^2}{\sigma^2})\ exp(2\pi\theta i(xcos\phi + ysin\phi)) \tag{7.14}$$

The response of a Gabor filter to an image can be obtained using 2D convolution operation. Let $I(x, y)$ denote the image, and $I_G(x, y, \theta, \phi)$ denote the response of a Gabor filter with frequency θ and orientation ϕ to an image at point (x, y) on the image plane. $G(.)$ is obtained as

$$I_G(x, y, \theta, \phi) = \int\int I(p, q) g(x - p, y - q, \theta, \phi) dp\ dq. \tag{7.15}$$

Cross Local Gabor Binary Patterns are a combination of XLBP and Gabor filters. Using this method, we can encode the shape of the objects in an image effectively, which is invariant to most of the modalities. Given the image $I(x, y)$, then the Cross Local Gabor Binary image is computed by following equation,

$$XLRBP_R^P\left(I_c^{G_f}\right) = \sum_{n=0}^{P-1} s\left(I_n^{G_f} - I_c^{G_f} - T\right) 2^n, \quad T \geq 0 \tag{7.16}$$

where the superscript G denotes Gabor filtered image, and $s(.)$ is thresholding function.

$$s(x) = \begin{cases} 1, & \text{for} \quad x \geq 0 \\ 0, & \text{for} \quad x < 0 \end{cases} \tag{7.17}$$

7.5 Probabilistic linear discriminant analysis on XLGBP descriptor

Due to quality deterioration, a broad range of changes in pose, occlusion, and expression changes, face identification on unconstrained face photos is still a difficult issue. Cross Local Gabor Binary Patterns (XLGBP), a novel face descriptor that extracts shape and texture both at coarse and fine levels, is designed to address these issues. XLGBP is a combination of Cross Local Binary Patterns (XLBP) [6] and Gabor filters [152]. The best Gaussian kernels for measuring local spatial frequencies at various scales and orientations are called Gabor filters. A modified variant of the conventional Local Binary Patterns (LBP) [53], XLBP, as discussed in the preceding Section 7.1, can extract the texture at both coarse and fine levels. Combining Gabor and XLBP allows one to capture the local intensity distribution with spatial information since Gabor filters are robust against minor translations and XLBP is robust against local intensity fluctuations and rotations of the pictures. The descriptor is hence resistant to changes in illumination, pose, and noise. In a nutshell, XLGBP is a method for representing textures in a multi-scale, multi-oriented spatial histogram. The image is filtered using Gabor Wavelets, and then the Cross Local Binary Patterns (XLBP) operator is used to calculate the feature histogram. To find the matching faces in the gallery database, Probabilistic Linear Discriminant Analysis (PLDA) [64] is used as a discriminator. The intra-class and inter-class variance are both modelled as multi-dimensional Gaussian in the probabilistic variant of fisher-faces known as PLDA. Since the multi-dimensional Gaussian has the highest level of discriminating, PLDA is appropriate for tasks requiring class recognition. Before modeling them, the feature vector's dimensionality is reduced using PLDA Kernel Principal Component Analysis (KPCA).

7.5.1 PLDA

Linear discriminant analysis (LDA) uses a multidimensional Gaussian to represent both inter-class and intra-class variations. It seeks the directions in the feature space where feature vectors have the greatest discriminating ability. As a result, it is the best suited for recognition tasks. It is assumed that the training data consists of J images of each of I individuals. Let $\mathbf{X}_{ij}$ is j^{th} image of the i^{th} individual, then the model data is generated by the process:

$$\mathbf{x}_{ij} = \mu + \mathbf{F}\mathbf{h}_i + \mathbf{G}\mathbf{w}_{ij} + \epsilon_{ij}, \tag{7.18}$$

where $\mu, \mathbf{F}\mathbf{h}_i, \mathbf{G}\mathbf{w}_{i,j}$ and ϵ_{ij} are model parameters to be estimated. This model comprises of two parts: (i) the signal component $\mu + \mathbf{F}\mathbf{h}_i$, which depends on the identity of the person but not the specific image (depends on i but not on j), that describes the inter-class variation.

(ii) the noise component $\mathbf{Gw}_{ij} + \epsilon_{ij}$, represents the intra-class noise, which is different from image to image even those belonging to the same individual. The term μ represents the overall mean of the training dataset. The columns of the matrix $\mathbf{F}$ contain a basis for the inter-class subspace, and the term $\mathbf{h}_i$ represents the position in that subspace. The matrix $\mathbf{G}$ contains a basis for the intra-class subspace, and $\mathbf{w}_{ij}$ represents the position in this subspace. Remaining unexplained data variations is explained by the residual noise term ϵ_{ij}, which is defined as Gaussian with diagonal covariance Σ.

7.5.1.1 Training

The training process estimates the model parameters $\theta = \{\mu, \mathbf{Fh}_i, \mathbf{Gw}_{ij}, \Sigma\}$ such that the true positives are most likely. It becomes easy if the values of latent variables $\mathbf{h}_i$ and $\mathbf{w}_{ij}$ are known. Similarly it would also become easy to estimate $\mathbf{h}_i$ and $\mathbf{w}_{ij}$ for a given μ. For solving this equation 7.18 popular EM (Expectation-maximization) algorithm is used, which estimates two sets of parameters in such a way that the likelihood is guaranteed to increase at each iteration. This algorithm has two steps: in Expectation or E-Step, a full posterior distribution over the latent variables $\mathbf{h}_i$ and $\mathbf{w}_{ij}$ for fixed parameter values is estimated, while in the Maximization or M-Step, the point estimates of the parameters $\theta = \{\mu, \mathbf{Fh}_i, \mathbf{Gw}_{ij}, \epsilon_{ij}\}$ are optimized.

7.5.1.2 Face Recognition

A model $\mathbf{M}$ can be represented as the relationship between the underlying identity variables, $\mathbf{h}_i$ and the data. For R identities R models will be learnt. During recognition, if there exists R different models $\mathbf{M}_1 \ldots, \mathbf{M}_R$, then the likelihood of the data with these models is compared. If two or more faces belong to the same person, then the corresponding identity variable $\mathbf{h}_i$ is same for both the faces. If two faces belong to different people they identity variables will be different for both. For the q^{th} model, calculate the likelihood term $P(\mathbf{x}/\mathrm{M}_q)$, where $\mathbf{x}$ is all the observed data. The posterior probability of the image for which the model is correct using Bayes' rule is given below:

$$P(\mathrm{M}_q|\mathbf{x}) = \frac{P(\mathbf{x}|\mathrm{M}_q)P(\mathrm{M}_q)}{\sum_{r=0}^{R} P(\mathbf{x}|\mathrm{M}_r)P(\mathrm{M}_r)} \tag{7.19}$$

7.5.2 Approach

The suggested method's general framework is built on a Cross Local Binary Pattern histogram sequence, which is calculated using the following process. (i) To create numerous Gabor magnitude pictures in the frequency domain, an input face image is normalized and convolved with 40 Gabor wavelets (5-scales and 8-orientations). (ii) Using Cross Local Binary

Patterns, each Gabor Magnitude Image is converted into binary patterns (XLBP). (iii) Each XLGBP map pattern picture is broken into non-overlapping rectangle areas of varying sizes, and histograms are generated for each block. (iv) The XLGBP histograms of all LGBP photos are concatenated to generate the final histogram sequence as the face image's feature descriptor. (v) The dimension of the feature vector is reduced using Kernel-PCA to enable it suitable for training, recognition process, and storing. (vi) PLDA is used for training on gallery images and probe images are used in the classification. The following subsections will describe the process in detail and are illustrated in Figure 7.5.

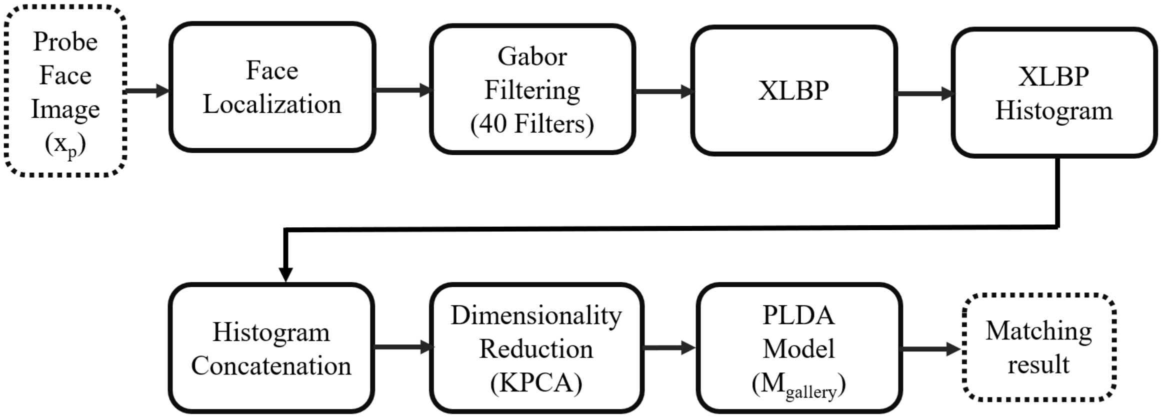


Figure 7.5: Functional diagram of XLGBP and PLDA based face recognition approach [8]

7.5.3 Face Representation Using XLGBP

The Gabor representation of a face image is derived by convolving the Gabor filters with face image as shown in Figure 7.6. Let $I(x, y)$ be the face image, the Gabor representation can be obtained using following equation:

$$G_{\Psi I}(x, y, \mu, \nu) = I(x, y) * \Psi_{\mu,\nu}(z), \tag{7.20}$$

where $\mu = 0, 1, \ldots, 7$ is orientation and $\nu = 0, \ldots, 4$ are the scale factors of the Gabor filter. Here, only the magnitude of the Gabor images is considered. The magnitude values of the Gabor images changes very slowly with displacement, so they can be further encoded. The performance of LBP increases when it is used with pre-processing filters. The Gabor images magnitude values are encoded with the Cross Local Binary Patterns operator. The convolution of the face image with 40 Gabor filters is carried out, as shown in Figure 7.6. Consider g is one of the 40 Gabor images, then XLGBP pattern images can be computed to obtain XLGBP maps using

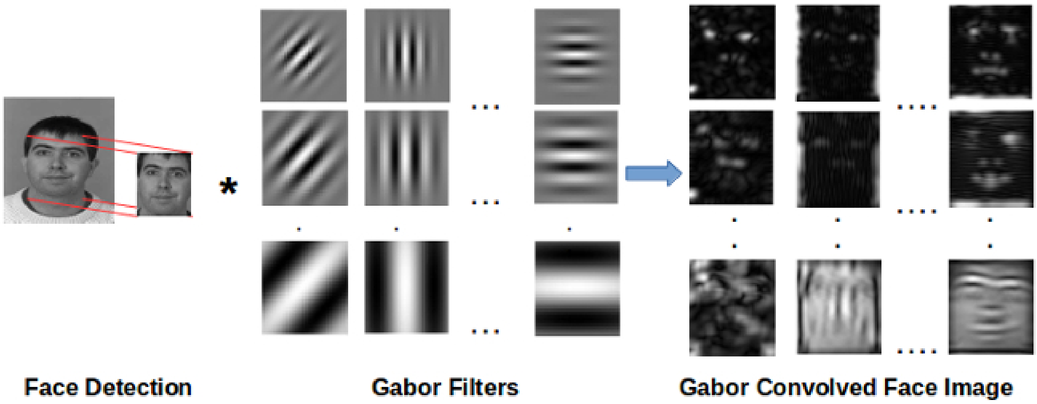


Figure 7.6: Convolution of face image with 40 Gabor filters (5-scales * 8 - orientations)

$$XLGBP_R^P\left(I_c^g\right) = \sum_{n=0}^{P-1} s\left(I_n^g - I_c^g - T\right)2^n, T \geq 0 \tag{7.21}$$

where $s(.)$ is thresholding function and g is the Gabor transform.

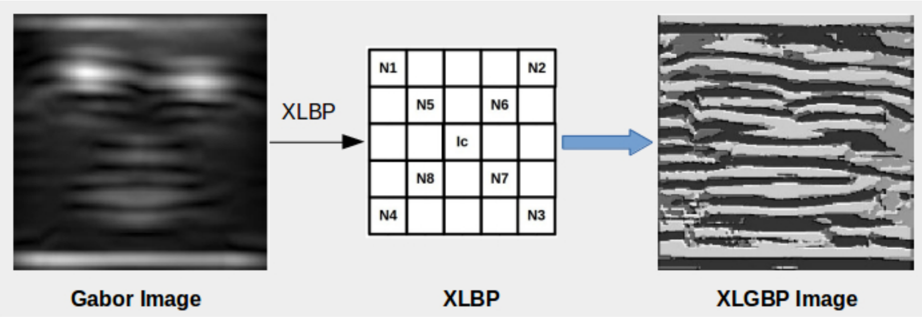


Figure 7.7: Texture extraction on Gabor face. (XLBP with radius R=2 and neighborhood P=8)

Some facial expression and illumination changes are specific to some regions in the face. To summarize the region properly, local feature histograms of the XLGBP maps are used, as shown in Figure 7.7. This process is carried out by dividing the XLGBP image into multiple non-overlapping regions spatially. The histograms computed on all these non-overlapping regions are concatenated to form a single vector sequence to represent the facial features of a single person as shown in Figure 7.8. Histogram of each spatially divided region ($sg_{\mu,\nu,m} : 1 \leq m \leq no\ of\ regions$) of an XLGBP image f is computed by

$$h_{XLGBP(sg_{\mu,\nu,m}),j} = \sum_{x,y} I\left(XLGBP(sg_{\mu,\nu,m})(x,y) = j\right), \tag{7.22}$$

where i is the i^{th} region of the XLGBP image f, j is the j^{th} gray level and

$$s(x) = \begin{cases} 1, & \text{A is True} \\ 0, & \text{A is False} \end{cases} \tag{7.23}$$

If each XLGBP map image is divided into m regions, the overall concatenated histogram sequence is

$$\mathbf{H} = \{h_{0,0,1}, \ldots, h_{0,0,m}, \ldots, h_{7,4,1}, \ldots, h_{7,4,m}\} \tag{7.24}$$

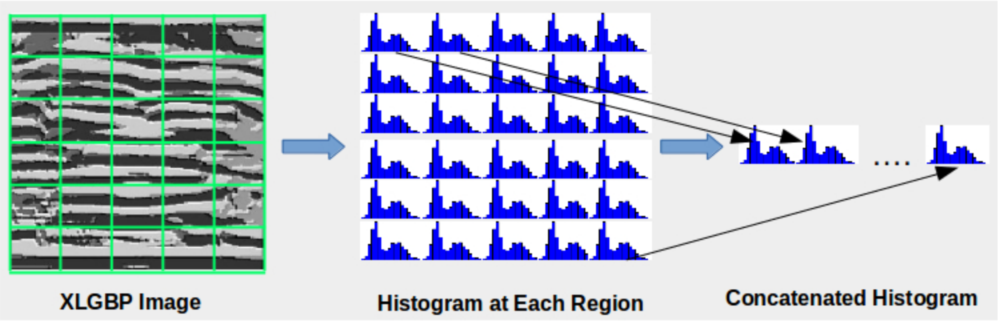


Figure 7.8: XLGBP histogram patterns concatenation shown for one XLGBP image. This process is followed for all 40 XLGBP images for complete histogram feature vector

7.5.4 Training and classification

Consider a face image of size 192 × 160 is spatially divided into non-overlapped regions of size 32 × 32, and the number of bins in each histogram would be 8. Then the length of the feature vector is 9600 ($\frac{(192\times160)\times40}{(32\times32)\times8}$), which is relatively large for both processing and storing. Kernel PCA k-PCA was used for dimensionality reduction. To train PLDA, these low dimensional feature vectors are used, and a generative model is computed with model parameters $\theta = \{\mu, \mathbf{F}, \mathbf{G}, \Sigma\}$ to maximize the inter-class difference and minimize the intra-class difference. The recognition process is carried out in the following manner. Consider two gallery faces $\mathbf{x}_1$ and $\mathbf{x}_2$, which belong to two different persons, and a probe face $\mathbf{x}_p$. In the training, two models are generated $\mathbf{M}_1$ and $\mathbf{M}_2$. If the probe image $\mathbf{x}_p$ matches with the model $\mathbf{M}_1$, then it will share the latent variable $\mathbf{h}_1$, and the gallery image $\mathbf{x}_2$ has its own identity variable. Similarly if the probe image matches with model $\mathbf{M}_2$, then it will share the identity variable $\mathbf{h}_2$. As $\mathbf{x}_1$ and $\mathbf{x}_2$ are independent, the likelihood model of the data under $\mathbf{M}_1$ can be written as

$$P\left(\mathbf{x}_{1,2,p}|\mathrm{M}_1\right) = P\left(\mathbf{x}_{1,p}|\mathrm{M}_1\right) P\left(\mathbf{x}_2|\mathrm{M}_1\right) \tag{7.25}$$

In the case of verification, if the probe image $\mathbf{x}_p$ matches with the model $\mathbf{M}_1$, then $\mathbf{x}_1$ and $\mathbf{x}_p$ will share same identity variables. Otherwise they will share different identity variables.

If a probe input image is given, in the PLDA-based approach, the gallery model with the highest probability would be considered as a matched face. Even though a good recognition rate is obtained with this approach, the false positive rate is high.

To find the correct threshold for separating faces having higher probability scores,we used the complete gallery set for validation and found the matching probabilities of the images of the same individual and different individuals, and fitted two separate Gaussian distributions. By conducting experiments on different datasets, it is observed that the variance of the Gaussians of different faces is smaller than the Gaussian of the same faces. Hence, the threshold is fixed around the mean of the probability distribution of the same individual. From these observations, the threshold for probability for better classification is computed as below:

$$\mathbf{T} = \mu_{true} - 2 * \sigma_{true}, \tag{7.26}$$

where μ_{true} and σ_{true} are mean and standard deviation of the probability distribution of the true positives.

7.5.5 Results and discussion

The proposed approach includes two important steps: (i) computing XLGBP feature vectors, in which 192 × 160 size face image is convolved with 40 different Gabor wavelets (5 scales and 8 orientations). XLBP(2,8) operator applied on these convolved images for texture extraction. Later each XLGBP map image is divided into 32 × 32 size non-overlapping regions. To arrive at the XLGBP histogram feature vector, the histograms of all regions are concatenated sequentially to (ii) The feature vectors are normalized and dimensionality reduced. Through statistical analysis, an optimal threshold value is estimated for classification with reduced false positives. In this, we conducted two types of experiments on this method. The first experiment considers frontal faces with different illumination conditions and the second one involved images with pose changes.

The first experiment was carried out on color FERET [153] and Extended YaleB [154] datasets. We considered 900 images of the FERET dataset from each Fa and Fb (frontal pose)categories with single image per subject, while in the YaleB dataset, we considered 1520 images of 38 subjects (frontal pose) with 45 different illumination variations. Experimental results on frontal pose faces are given in Table 7.6 and Table 7.7.

In the second experiment, we used the GrayFERET dataset, and an internally collected dataset for testing pose invariance. Our internal dataset contained 4910 images of 19 subjects with several poses. GrayFERET contained 2200 images of 200 persons with 11 pose variations. The experimental results on pose invariant face recognition are as given in Tables

Table 7.6: Face recognition performance comparison on FERET dataset (frontal only) [8]

Method	**Fa**	**Fb**	**Recognition Rate**
LBP [55]	900	900	88.96
LRBP [7]	900	900	95.31
XLRBP [6]	900	900	98.89
LGBP [155]	900	900	97.33
XLGBP	900	900	99
XLGBP+PLDA	900	900	100

Table 7.7: Performance comparison on YaleB dataset (Frontal images with different illumination conditions)

Method	**Train**	**Test**	**Recognition Rate**
LBP [55]	38	1520	83.86
LRBP [7]	38	1520	88.57
XLRBP [6]	38	1520	90.20
LGBP [155]	38	1520	91.40
XLGBP	38	1520	93.03
XLGBP+PLDA	38	1520	96.7

7.8 and 7.9.

Table 7.8: Performance comparison on GreyFERET dataset (11 poses per each of 200 persons)

Method	**Recognition Rate**
LBP [55]	83.86
DCP [59]	95.57
LRBP [7]	94.41
XLRBP [6]	94.54
LGBP [155]	93.69
XLGBP	95.32
XLGBP+PLDA	98.09

7.6 Gender detection using XLBP descriptor

Local binary patterns have been used for gender classification and it can be easily computed by using neighboring pixels and a centre pixel with in a NxN window. For the gender

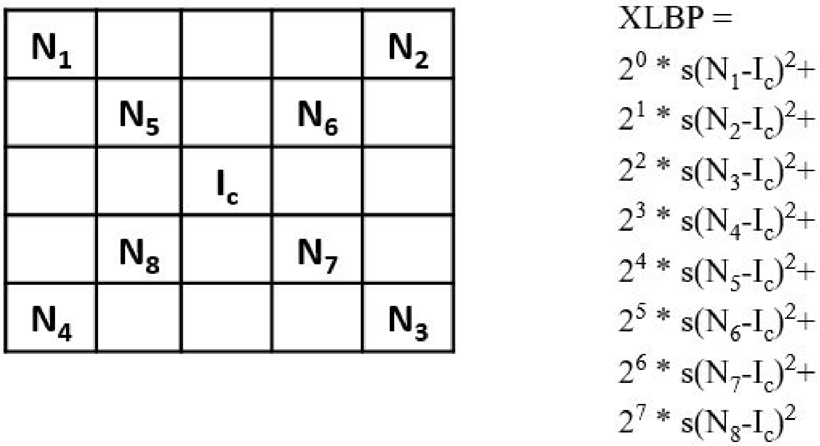
Table 7.9: Performance comparison on our own Dataset (4910 images of 19 persons)

Method	**Recognition Rate**
LBP [55]	92.87
DCP [59]	95.72
LRBP [7]	94.42
XLRBP [6]	94.70
LGBP [155]	97.35
XLGBP	97.75
XLGBP+PLDA	98.07

classification, local binary patterns represent the texture information of a given face. In the literature LBP has been used for gender classification, but the performance can still be improved. To improve the performance over other LBP-based methods, we propose to use XLBP, which contains a modified neighborhood of the original LBP, spatial pyramid representation, and can effectively discriminate facial texture, which is important for gender classification. The modified neighborhood of LBP is now computed by considering eight pixels in the diagonal position as instead of all surrounding pixels within the window, as shown in Figure 7.9. This avoids the additional computation required for interpolation. The modified neighborhood pixels are not in uniform locations as in the traditional LBP. The locations of the neighborhood chosen intelligently for representing the texture with less computation. Our hypothesis is that the shapes contain more line structures in diagonal directions Performance of the modified neighborhood LBP exhibited improvement over the LBP [29], due to its modified neighborhood and ability of LBP to discriminate the face for gender classification better than the traditional LBP.

Block diagram of the proposed gender classification is shown in Figure 7.10. For the gender classification, the face is divided into non-overlapping blocks of size 25 x 25, and the proposed XLBP is computed on each block. Facial feature descriptor for gender classification is arrived by concatenating the histograms of XLBPs of each block. These concatenated histograms can represent both at local level and coarse level texture information. A linear SVM classifier is trained on facial descriptors obtained from each face image in the training set of labeled faces (male and female). The trained SVM was used for gender classification during the test time.

In this work, we combined face datasets of FERET [153] and FEI [123] together for gender classification which contains 600 male faces and 600 female faces. Faces are normalized into 200x200 resolution, and each face is divided into sub-faces with a window size of 25x25. The proposed modified LBP operator (XLBP) is applied on each sub-face, and the histogram is obtained. The histograms of all modified LBP sub-faces are concatenated to form a single feature descriptor for gender classification, which describes the texture of the given face.



$$\begin{aligned}\text{XLBP} = {} & 2^0 * s(N_1 - I_c)^2 + \\ & 2^1 * s(N_2 - I_c)^2 + \\ & 2^2 * s(N_3 - I_c)^2 + \\ & 2^3 * s(N_4 - I_c)^2 + \\ & 2^4 * s(N_5 - I_c)^2 + \\ & 2^5 * s(N_6 - I_c)^2 + \\ & 2^6 * s(N_7 - I_c)^2 + \\ & 2^7 * s(N_8 - I_c)^2\end{aligned}$$

Figure 7.9: XLBP (R=2,P=8) computation on a face image

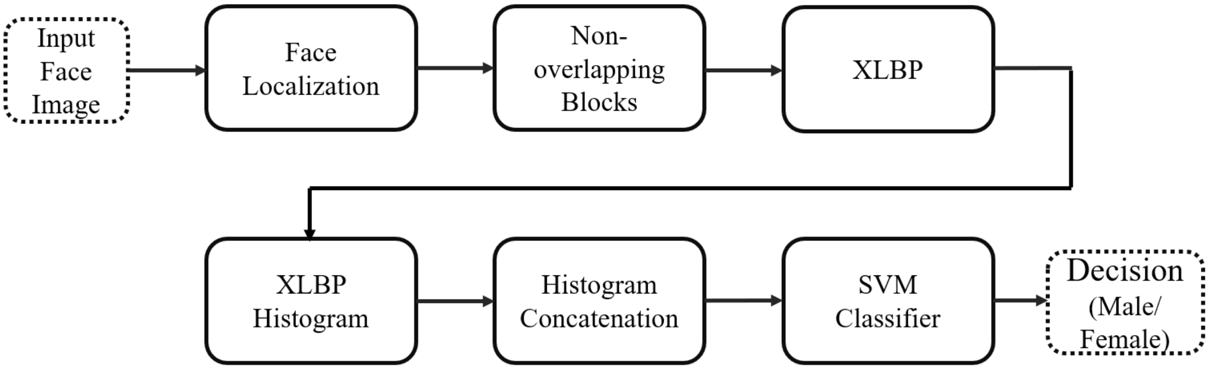


Figure 7.10: Gender classification using XLBP facial features [9]

We trained a linear SVM classifier with labels combined face dataset, and testing has been performed by k-fold cross validation. We also implemented existing techniques based on standard LBP with radius R ($R = 2$).

7.6.1 Results and discussion

In this we present the experimental results of the proposed XLBP operator and the traditional LBP. The results show that the performance of the modified LBP (XLBP) is better than the traditional LBP for gender classification. The LBP and xLBP were compared in three ways. In all the cases, the results show that the modified LBP performs better than the traditional LBP for gender classification with much less computations.

In the first experiment, LBP (2,8) and XLBP (2,8) were compared with k-fold cross validation by varying the k value. Gender recognition rate increases as k increases as shown

Table 7.10: Comparison of gender detection between traditional LBP and proposed XLBP with leave-one-out method

Feature and Classifier	**Training and Testing**	**Gender recognition rate**
Traditional LBP (2,8) + SVM (linear)	Leave-one-out method	95.42
Proposed LBP (2,8) + SVM (linear)	Leave-one-out method	96.17

Table 7.11: Gender detection comparison between traditional LBP and proposed XLBP with different block sizes

Feature and Classifier	**Gender recognition rates with 40-fold cross validation**		
	Block size 10 x 10	**Block size 25 x 25**	**Block size 50 x 50**
Traditional LBP (2,8) + SVM (linear)	93.5	94.16	92
Proposed LBP (2,8) + SVM (linear)	94.66	95	94.16

in Figure 7.11. In the second experiment, LBP (2,8) and XLBP (2,8) were compared with leave one out testing method as given in Table 7.10. In this method, 1199 faces out of 1200 faces are used for training and remaining one face for testing. This is repeated for 1200 times with each face being used for testing. In the third experiment, we compared LBP (2,8) and XLBP (2,8) are compared with the various window sizes, and high gender recognition rates were obtained with a window size of 25 x 25 as given in Table 7.11. Figure 7.12 illustrates the performance of gender detection on the images from the FERET database.

It has been shown that, the proposed variant local binary pattern operator outperforms the other LBP methods for face recognition without any increase in the computational requirement. In addition to the XLBP operator, a Radon transform based feature representation for better face recognition is proposed. From the results it is evident that, the proposed

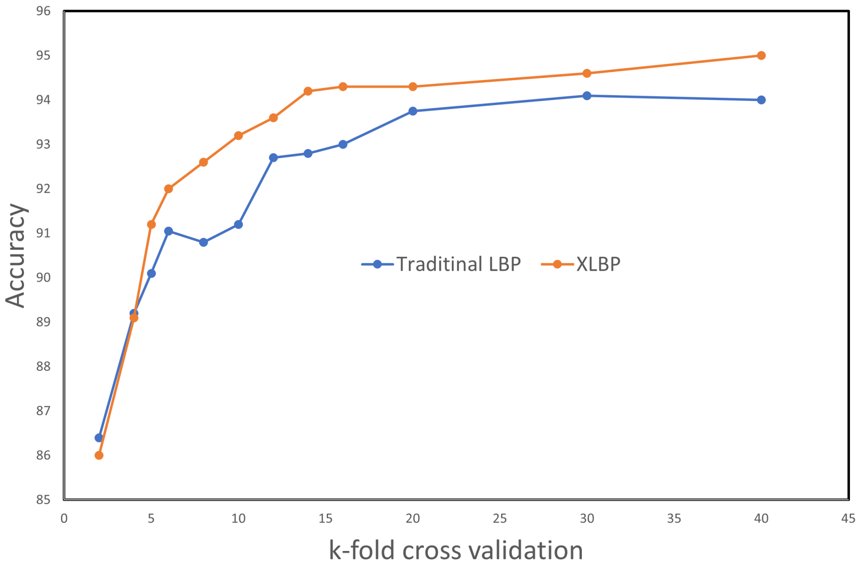


Figure 7.11: Comparison of gender detection performance with traditional LBP and proposed XLBP LBP with various k values [9]

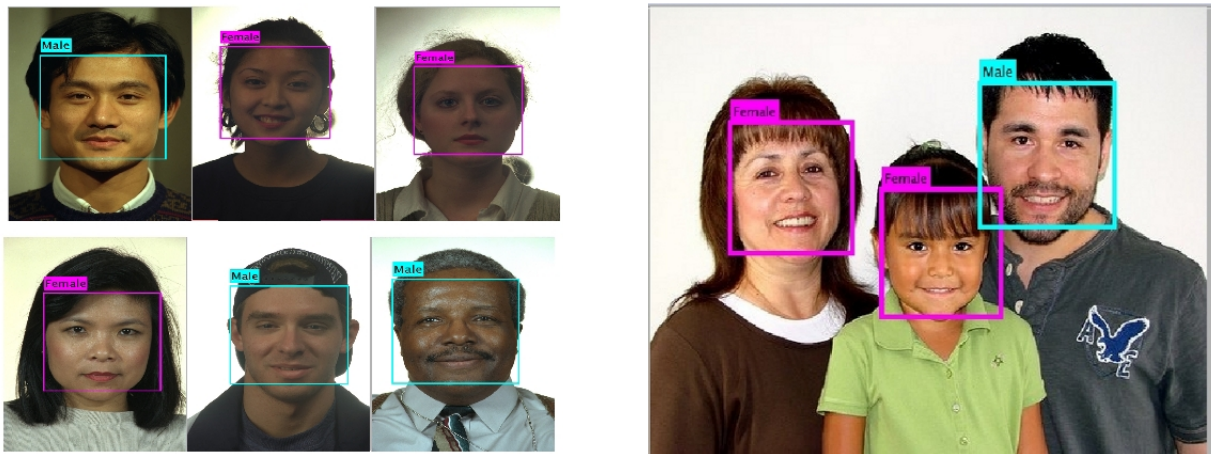


Figure 7.12: Gender detection example results on FERET dataset (best viewed in color)

feature representation performs better, for gender detection in addition to face recognition compared to the vanilla LBP operators.

7.7 Conclusion

Effective feature descriptor is a key factor for the performance of any feature-based face recognition methods. In this chapter, a novel face descriptor XLBP and its variants XLRBP and XLGBP are presented. First, we defined the XLBP by varying the neighborhood of LBP and found that a 'X' shaped neighborhood performs better in terms of representation. Then we followed the spatial pyramid approach to describe the face by concatenating XLBP of non-overlapping blocks. Similarly, we applied Radon transform for representing the face due to its ability to represent the face in different orientations to arrive XLRBP.

Similarly, we applied Gabor transform on the face images and applied XLBP on Gabor magnitude to develop XLGBP. These features overcome the limitations of face recognition under illumination and pose variations. This method encodes the local Gabor magnitude differences between neighborhood pixels, representing the shape attributes of a given face image. To compute the feature vector of the image, the spatial histograms at each scale and orientation are concatenated and the resultant feature descriptor contains both the structure information and texture information.

Also, we presented PLDA which is a probabilistic discriminator and allows the non-linearity by capturing non-linear relationship between different poses of the same person. We conducted different experiments on both frontal faces and images with pose variations. The experimental results show that this method performs better than other traditional LBP-based methods with Bhattacharya distance. The performance is even better when probabilistic linear discriminant analysis is used for classification. We also described a gender classification approach based on the proposed XLGBP descriptor and shown that, the proposed descriptor performs well compared to traditional LBP.

In the previous chapters, we discussed various studies on template protection, feature embedding protection, protection against adversarial perturbations for face recognition. Also, in this chapter we discussed a computationally light face feature descriptor. In the next Chapter, 8, conclusions of the studies conducted in this thesis is provided along with future research directions to take this work forward.

Chapter 8

Conclusions

8.1 Summary

Rapid digital transformation due to COVID-19 increased the security and privacy issues related to digital services, as many people come online and a large percentage of them are first-time users. Biometrics is used as one of the authenticating factors for gaining access to services. However, these biometric systems have several issues with respect to their security and privacy. Attacks on biometrics can happen at different stages of a biometrics system. Unlike passwords, biometrics are unique and can not be replaced like passwords or tokens. Hence, approaches for protecting biometric systems at various stages are the need of the hour. Particularly in this thesis is we focused on face biometric.

Numerous face verification methods have been proposed in the literature, most of which focus solely on improving the performance. However, adversarial attacks degrade the performance of the face recognition model and lead to security issues. To address this, we proposed a method for countering the adversarial attack through modular Siamese networks (MSN). This approach provides both robustness as good explainability. We believe that pursuing this direction is essential for developing more trustworthy systems. Due to the critical nature of the biometrics systems, there is very less room for mistakes. Incorporating interpretability to the system itself could allow us to handle errors. In Chapter 3, we presented a novel technique to learn latent representations of high-level facial feature-specific representations. We have demonstrated that the proposed MSN based face verification system is resistance to adversarial examples generated by FGSM, FFGSM, and DeepFool attacks. Additionally, we extended the finding of feature-specific representations to retrieve facial images closely matching the query image. This approach can retrieve faces even with a partial face (eyes, nose, mouth etc.) as a query. Retrieval of faces with partial face query is critical in applications such as surveillance, where the face is only partly visible and available for recognition.

In this thesis, we also presented two methods for template protection in Chapter 4 (i) using a deep neural network to map the facial feature embedding to cancelable binary templates, and (ii) using targeted adversarial noise generation, which is used as auxiliary data and cancelable binary templates. These methods provide better recognition performance while providing high template security. We presented how the adversarial perturbations at the input data stage can be used for face template protection, and how the templates are secured using a deep neural network based mapping. Also, the problem of re-enrollment in template protection methods is addressed by using two sets of mapping codes and adversarial perturbations.

In Chapter 5, we presented a mechanism to prevent extracting face data from the feature embeddings while providing cancelability. This approach is based on random projections, which is a popular dimensionality reduction method. Random projections perform well at higher dimensions too. In this approach, the face embeddings are projected onto a lower dimensional subspace, and it is NP hard to recover the input feature embedding. A major advantage of random projections is that we can regenerate any projection matrix whenever a face template is compromised or a when a new user is enrolled. The proposed approach achieves both the performance and security of the biometric data due to non-invertibility of the Random projection matrix.

We also proposed a face descriptor, which is light weight in terms of computation, while effectively representing the facial features. In Chapter 7, we presented this novel face descriptor called XLBP and its variants XLRBP and XLGBP. We applied Radon transform and Gabor transform on the face images and further applied XLBP on Gabor magnitude to come up with XLRBP and XLGBP, respectively. The proposed LBPs are represented by spatial pyramid for further classification or comparison. These features overcome the limitations of face recognition under illumination and pose variations. Also, we presented PLDA, which is a probabilistic discriminator, and allows non-linearity in determining the non-linear relationship among different poses of the same person. We conducted experiments on both frontal faces and images with pose variations. The experimental result show that this method performs better than other LBP-based methods with Bhattacharya distance, and even better when PLDA is used for classification.

Despite several advantages of federated learning, several security and privacy challenges still exist. This includes a malicious user injecting poisoning updates into model aggregation, leading to sub-optimal models. Especially in federated face recognition, poison samples could create impersonation attacks and need to be addressed. In Chapter 6, We introduce federated learning and various possible security attacks on these models. We studied various aspects of federated learning and multiparty computation to understand and proceed with federated 'face' learning and for conducting studies on poisoning attacks that could create a backdoor for attackers in the system.

8.2 Main contributions

1. Developed a robust face verification method with inherent interpretability using modular Siamese networks.

2. Proposed a DNN-based face template protection method, and achieved better performance.

3. Proposed a face template protection method based on adversarial perturbations and achieved better performance and template security, while eliminating the need for re-enrolment.

4. Developed a random projection based face embedding protection method, and achieved good matching performance as well as security of face embeddings.

5. Explored approaches for federated face recognition and attacks on them.

6. Proposed XLBP-based face descriptors that achieve state of the art performance compared to the method in a similar category.

8.3 Scope for future work

1. The current approaches use a CNN-based neural network for generating secure templates. However, recently deep hashing methods being used to generate the hash of the input data directly. These hashes can be used to generate projections that can handle intra-class variations for template protection.

2. Similarly, Modular Siamese networks based content retrieval tasks can be extended to retrieve images using deep hash techniques.

3. Poisoning attacks on federated face recognition systems could cause several security issues, and these need to be studied and countered using appropriate approaches.

4. Modular Siamese network has the ability to represent individual latent representations effective. One of the major causes of bias in deep learning models is their inability to learn some features from the data. We strongly believe using a modular Siamese network, we can identify the causes of bias and improve fairness.

List of Publications

Journals

1. **Chalamala S.R.**, Kummari N. K., Singh A. K., Saibewar A., and Chalavadi K. M., ”Federated learning to comply with data protection regulations”, CSI Transactions on ICT, Spinger, Vol 10, pp. 47–64. 2022.

Conferences

1. Dammu P.P., **Chalamala S.R.**, Singh A.K. and Yegnanarayana B., ”Interpretable and Robust Face Verification”, in the Proceedings of CIKM Workshops, CEUR-WS, Vol-3052, 2021.

2. A. K. Jindal, I. Shaik, V. Vasudha, **S. R. Chalamala**, R. Ma and S. Lodha, ”Secure and Privacy Preserving Method for Biometric Template Protection using Fully Homomorphic Encryption” in the Proceedings of 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE , pp. 1127-1134, 2020.

3. S. K. Jami, **S. R. Chalamala** and A. K. Jindal, ”Biometric Template Protection Through Adversarial Learning”, in the Proceedings of International Conference on Consumer Electronics (ICCE), IEEE, pp. 1-6, 2019.

4. A. K. Jindal, **S. Rao Chalamala** and S. K. Jami, ”Securing Face Templates using Deep Convolutional Neural Network and Random Projection”, in the Proceedings of International Conference on Consumer Electronics (ICCE), IEEE, pp. 1-6, 2019.

5. A. K. Jindal, **S. Chalamala** and S. K. Jami, ”Face Template Protection Using Deep Convolutional Neural Network”, in the Proceedings of CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, pp. 575-5758, 2018.

6. Jami S.K., **Chalamala S.R.**, and Kakkirala, K.R.,”Cross Local Gabor Binary Pattern Descriptor with Probabilistic Linear Discriminant Analysis for Pose-Invariant Face

Recognition”, in the Proceedings of International Conference on Computer Modelling and Simulation (UKSim-AMSS), IEEE, pp. 39-44, 2017.

7. **S. R. Chalamala**, S. K. Jami and Yegnanarayana B., ”Enhanced face recognition using Cross Local Radon Binary Patterns,” in the Proceedings of International Conference on Consumer Electronics (ICCE), IEEE, pp. 481-484, 2015.

8. **S. R. Chalamala** and K. R. Kakkirala, ”Local Binary Patterns for Digital Image Watermarking”, in the Proceedings of International Conference on Artificial Intelligence Modelling and Simulation (AIMS), IEEE, pp. 159-162, 2015.

9. B. Gudla, **S. R. Chalamala** and S. K. Jami, ”Local Binary Patterns for Gender Classification”, in the Proceedings of International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), IEEE, pp. 19-22, 2015.

10. **S. R. Chalamala**, K. R. Kakkirala and J. S. Kumar, ”Face recognition using spatial pyramid matching and LRBP”, in the Proceedings of International Colloquium on Signal Processing and its Applications (ICSPA), IEEE, pp. 67-70, 2014.

Bibliography

[1] Preetam Prabhu Srikar Dammu, Srinivasa Rao Chalamala, Ajeet Kumar Singh, and Bayya Yegnanarayana. Interpretable and robust face verification. *ACM International Conference on Information and Knowledge Management (CIKM) Workshops*, 2021.

[2] Arun Kumar Jindal, Srinivasa Rao Chalamala, and Santosh Kumar Jami. Face template protection using deep convolutional neural network. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 575–5758, 2018.

[3] Santosh Kumar Jami, Srinivasa Rao Chalamala, and Arun Kumar Jindal. Biometric template protection through adversarial learning. *International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2019.

[4] Arun Kumar Jindal, Srinivasa Rao Chalamala, and Santosh Kumar Jami. Securing face templates using deep convolutional neural network and random projection. *IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2019.

[5] Srinivasa Rao Chalamala, Naveen Kumar Kummari, Ajeet Kumar Singh, Aditya Saibewar, and Krishna Mohan Chalavadi. Federated learning to comply with data protection regulations. *CSI Transactions on ICT*, 10:47–60, 2022.

[6] Srinivasa Rao Chalamala, Santosh Kumar Jami, and Bayya Yegnanarayana. Enhanced face recognition using cross local radon binary patterns. *IEEE International Conference on Consumer Electronics (ICCE)*, pages 481–484, 2015.

[7] Srinivasa Rao Chalamala, Krishna Rao Kakkirala, and Jami Santosh Kumar. Face recognition using spatial pyramid matching and lrbp. *IEEE 10th International Colloquium on Signal Processing and its Applications*, pages 67–70, 2014.

[8] Santosh Kumar Jami, Srinivasa Rao Chalamala, and Krishna Rao Kakkirala. Cross local gabor binary pattern descriptor with probabilistic linear discriminant analysis for pose-invariant face recognition. *2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim)*, pages 39–44, 2017.

[9] Balakrishna Gudla, Srinivasa Rao Chalamala, and Santosh Kumar Jami. Local binary patterns for gender classification. *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, pages 19–22, 2015.

[10] GDPR. General data protection regulation (gdpr), 2018. https://gdpr-info.eu/.

[11] CCPA. General data protection regulation (gdpr), 2018. http://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5.

[12] NITI AAYOG. Principles for responsible ai, 2021. https://www.niti.gov.in/sites/default/files/2021-02/Responsible-AI-22022021.pdf.

[13] NITI AAYOG. Principles for responsible ai, 2021. https://www.niti.gov.in/sites/default/files/2021-08/Part2-Responsible-AI-12082021.pdf.

[14] Anil K Jain, A A Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:4–20, 2004.

[15] Anil K Jain and Karthik Nandakumar. Multibiometric systems: fusion strategies and template security. 2008.

[16] Abhishek Nagar, Karthik Nandakumar, and Anil K Jain. Multibiometric cryptosystems based on feature-level fusion. *IEEE Transactions on Information Forensics and Security*, 7:255–268, 2012.

[17] Abhishek Sharma, Murad Al Haj, Jonghyun Choi, Larry S. Davis, and David W. Jacobs. Robust pose invariant face recognition using coupled latent space discriminant analysis. *Computer Vision and Image Understanding*, 116:1095–1110, 11 2012.

[18] Xi Yin and Xiaoming Liu. Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Transactions on Image Processing*, 27:964–975, 2018.

[19] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38:188–194, 1 2016.

[20] Jan Rupnik and John Shawe-Taylor. Multi-view canonical correlation analysis. *Conference on data mining and data warehouses (SiKDD 2010)*, pages 1–4, 2010.

[21] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. *International conference on machine learning*, pages 1247–1255, 2013.

[22] Abhishek Sharma, Abhishek Kumar, Hal Daumé, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2160–2167, 2012.

[23] David Roi Hardoon, Sándor Szedmák, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16:2639–2664, 2004.

[24] Jangsun Baek and Min-Soo Kim. Face recognition using partial least squares components. *Pattern Recognition*, 37:1303–1306, 2004.

[25] Annan Li, S Shan, and Wen Gao. Coupled bias–variance tradeoff for cross-pose face recognition. *IEEE Transactions on Image Processing*, 21:305–315, 2012.

[26] Xiujuan Chai, S Shan, Xilin Chen, and Wen Gao. Locally linear regression for pose-invariant face recognition. *IEEE Transactions on Image Processing*, 16:1716–1725, 2007.

[27] Ahmed Bilal Ashraf, Simon Lucey, and Tsuhan Chen. Learning patch correspondences for improved viewpoint invariant face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[28] G LoweDavid. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (ICCV)*, 2004.

[29] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24:971–987, 7 2002.

[30] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. *International Conference on Image Processing (ICIP)*, 1:129–132 vol.1, 1997.

[31] Simon Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. *IEEE International Conference on Computer Vision (CVPR)*, pages 1–8, 2007.

[32] Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, F Zhao, Jayashree Karlekar, Sugiri Pranata, Shengmei Shen, Junliang Xing, Shuicheng Yan, and Jiashi Feng. Towards pose invariant face recognition in the wild. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2207–2216, 2018.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[34] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. *British Machine Vision Conference (BMVC)*, 2015.

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E Reed, Dragomir Anguelov, D Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[36] Kaiming He, X Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 07-12-June:815–823, 10 2015.

[38] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. *International Conference on Machine Learning (ICML)*, pages 507–516, 2016.

[39] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746, 2017.

[40] Yutian Li, Feng Gao, Zhijian Ou, and Jiasong Sun. Angular softmax loss for end-to-end speaker verification. *International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 190–194, 2018.

[41] H. Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jin Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5265–5274, 2018.

[42] Ross B Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[43] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 39:1137–1149, 2015.

[44] Jiankang Deng, J Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *ArXiv*, abs/1905.00641, 2019.

[45] Paul A Viola and Michael J Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:I–I, 2001.

[46] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[47] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:886–893 vol. 1, 2005.

[48] Thorsten Joachims, Thomas Hofmann, Yisong Yue, and Chun-Nam John Yu. Predicting structured objects with support vector machines. *Communications of the ACM*, 52:97 – 104, 2009.

[49] W Liu, Dragomir Anguelov, D Erhan, Christian Szegedy, Scott E Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. *European Conference on Computer Vision (ECCV)*, 2016.

[50] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23:1499–1503, 2016.

[51] Guoying Zhao, Timo Ahonen, Jiri Matas, and Matti Pietikäinen. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing*, 21:1465–1477, 2012.

[52] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. *Lecture Notes in Computer Science*, 3021:469–481, 2004.

[53] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28:2037–2041, 2006.

[54] XueMei Zhao and ChengBing Wei. A real-time face recognition system based on the improved lbph algorithm. *IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pages 72–76, 2017.

[55] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28:2037–2041, 2006.

[56] Dattatray V. Jadhav and Raghunath S. Holambe. Feature extraction using radon and wavelet transforms with application to face recognition. *Neurocomputing*, 72:1951–1959, 3 2009.

[57] Ville Ojansivu and Janne Heikkilä. Blur insensitive texture classification using local phase quantization. *Lecture Notes in Computer Science*, 5099 LNCS:236–243, 2008.

[58] Timo Ahonen, Esa Rahtu, Ville Ojansivu, and Janne Heikkilä. Recognition of blurred faces using local phase quantization. *19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[59] Changxing Ding, Jonghyun Choi, Dacheng Tao, and Larry S Davis. Multi-directional multi-level dual-cross patterns for robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38:518–531, 2016.

[60] Juho Kannala and Esa Rahtu. Bsif: Binarized statistical image features. *21st International Conference on Pattern Recognition (ICPR)*, pages 1363–1366, 2012.

[61] Matthew A Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[62] Kamran Etemad and Ramalingam Chellappa. Discriminant analysis for recognition of human face images. *Journal of The Optical Society of America A-optics Image Science and Vision*, 14:1724–1733, 1997.

[63] Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13:1450–1464, 11 2002.

[64] Simon J D Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[65] Simon J.D. Prince and James H. Elder. Tied factor analysis for face recognition across large pose changes. *British Machine Vision Conference (BMVC)*, pages 889–898, 2006.

[66] Akshat Agrawal. A review for face recognition using gabor wavelet transform. *Pattern Analysis and Applications*, 2017.

[67] Nalini K Ratha, Jonathan H Connell, and Ruud M Bolle. An analysis of minutiae matching strength. *AVBPA*, 2001.

[68] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *Proceedings of the ACM Conference on Computer and Communications Security*, 24-28-October-2016:1528–1540, 10 2016.

[69] Marcos Martinez-Diaz, Julian Fierrez, Javier Galbally, and Javier Ortega-Garcia. An evaluation of indirect attacks and countermeasures in fingerprint verification systems. *Pattern Recognition Letters*, 32:1643–1651, 9 2011.

[70] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures, 2015.

[71] Ziqi Yang, Ee-Chien Chang, and Zhenkai Liang. Adversarial neural network inversion via auxiliary knowledge alignment. *ArXiv*, abs/1902.08552, 2019.

[72] Y C Feng, P C Yuen, and A K Jain. A hybrid approach for generating secure and discriminating face template. *IEEE Transactions on Information Forensics and Security (TIFS)*, 5:103–117, 2010.

[73] Anil K Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric template security. *EURASIP J. Adv. Signal Process*, 2008, 1 2008.

[74] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. Springer Science and Business Media, 2009.

[75] Salil Prabhakar, Sharath Pankanti, and Anil K Jain. Biometric recognition: Security and privacy concerns. *IEEE Security and Privacy (S&P)*, 1:33–42, 3 2003.

[76] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *Advances in Neural Information Processing Systems*, 3:1988–1996, 2014.

[77] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE/CVF Conference on Computer Vision andPattern Recognition (CVPR)*, pages 4685–4694, 2019.

[78] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6746, 2017.

[79] Yu Liu, Hongyang Li, and Xiaogang Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *ArXiv*, abs/1710.00870, 2017.

[80] Mikhail Aleksandrovich Pautov, Grigorii Melnikov, Edgar Kaziakhmedov, Klim Kireev, and Aleksandr Petiushko. On adversarial patches: Real-world attack on arcface-100 face recognition system. *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pages 0391–0396, 2019.

[81] Avishek Joey Bose and Parham Aarabi. Adversarial attacks on face detectors using neural net based constrained optimization. *International Workshop on Multimedia Signal Processing (MMSP)*, 5 2018.

[82] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. *36th International Conference onMachine Learning*, 97:1310–1320, 7 2019.

[83] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in Neural Information Processing Systems 31*, pages 4939–4948, 2018.

[84] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

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

[86] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography 2006 38:2*, 38:237–257, 2 2006.

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

[88] N K Ratha, S Chikkerur, J H Connell, and R. M Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29:561–572, 2007.

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

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

[91] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision (ICCV)*, 128:336–359, 2019.

[92] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018.

[93] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10, 2015.

[94] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ArXiv*, abs/1703.01365, 2017.

[95] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *ArXiv*, abs/1705.07874, 2017.

[96] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “why should i trust you?”: Explaining the predictions of any classifier. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[97] Haoran Jiang and Dan Zeng. Explainable face recognition based on accurate facial compositions. *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1503–1512, 2021.

[98] Jonathan R Williford, Brandon B May, and Jeffrey Byrne. Explainable face recognition. *European Conference on Computer Vision (ECCV)*, 2020.

[99] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations, (ICLR)*, 12 2014.

[100] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23:828–841, 2019.

[101] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy (S&P)*, pages 39–57, 6 2017.

[102] Albrecht Schmidt and Zuhair Bandar. Modularity - a concept for new neural network architectures. *IASTED International Conference on Computer Systems and Applications, Irbid, Jordan, 1998*, 1998.

[103] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[104] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. *International Conference on Machine Learning Workshop (ICMLW)*, 2, 2015.

[105] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. *International Workshop on Similarity-Based Pattern Recognition*, 2014.

[106] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:539–546 vol. 1, 2005.

[107] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–412, 2017.

[108] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016-December:1335–1344, 12 2016.

[109] X Hou, L Shen, K Sun, and G Qiu. Deep feature consistent variational autoencoder. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141, 2017.

[110] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. *13th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 67–74, 2018.

[111] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 12 2014.

[112] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst*, 10 2007.

[113] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2574–2582, 2016.

[114] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[115] Fei Zuo, Bokai Yang, Xiaopeng Li, and Qiang Zeng. Exploiting the inherent limitation of l0 adversarial examples. *International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*, pages 293–307, 2019.

[116] Mandar Kulkarni and Aria Abubakar. Siamese networks for generating adversarial examples. *arXiv preprint arXiv:1805.01431*, 2018.

[117] Gasser Auda and Mohamed Kamel. Modular neural networks: a survey. *International Journal of Neural Systems*, 9:129–151, 1999.

[118] A. Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. *Very Large Data Bases Conference*, 1999.

[119] Y C Feng, P C Yuen, and A K Jain. A hybrid approach for generating secure and discriminating face template. *IEEE transactions on information forensics and security (TIFS)*, 5:103 –114, 2010.

[120] A K Jain, K Nandakumar, and A Nagar. Biometric template security. *EURASIP Journal on advances in signal processing*, 2008:1–17, 2008.

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

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

[123] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28:902–913, 6 2010.

[124] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations Workshop (ICLRW)*, 2017.

[125] N Papernot, P McDaniel, S Jha, M. Fredrikson, Z B Celik, and A Swami. The limitations of deep learning in adversarial settings. *IEEE European Symposium on Security and Privacy (EuroSIP)*, pages 372–387, 2016.

[126] Abhishek Nagar, Karthik Nandakumar, and Anil K Jain. Biometric template transformation: a security analysis. *Media Forensics and Security II*, 7541:237–251, 2010.

[127] C Szegedy, W Zaremba, I Sutskever, J. Bruna, D Erhan, I J Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014.

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

[129] P. Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16:295–306, 4 1998.

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

[131] William B Johnson. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984.

[132] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera Arcas. Communication-efficient learning of deep networks from decentralized data. *AISTATS*, pages 1273–1282,, 2017.

[133] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. *IEEE International Conference on Communications (ICC)*, pages 1–7, 2019.

[134] Jingjing Chen, Liangming Pan, Zhipeng Wei, Xiang Wang, Chong-Wah Ngo, and Tat-Seng Chua. Zero-shot ingredient recognition by multi-relational graph convolutional network. *AAAI Conference on Artificial Intelligence*, pages 10542–10550, 2020.

[135] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *International Conference on Neural Information Processing Systems(NeurIPS)*, pages 4427–4437, 2017.

[136] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. *IEEE Conference on Computer Communications*, pages 2512–2520, 2019.

[137] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. *ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618, 2017.

[138] Ligeng Zhu and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32:17–31, 2019.

[139] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[140] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *IEEE Symposium on Security and Privacy (S&P)*, 5 2019.

[141] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. *International Conference on Machine Learning*, pages 634–643, 2019.

[142] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Model poisoning attacks in federated learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[143] L I U Ximeng, X I E Lehui, WANG Yaopeng, and L I Xuru. Adversarial attacks and defenses in deep learning. *Chinese Journal of Network and Information Security*, 6:346–360, 2020.

[144] Jamie Hayes and Olga Ohrimenko. Contamination attacks and mitigation in multi-party machine learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[145] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. *IEEE Symposium on Security and Privacy (S&P)*, pages 739–753, 5 2019.

[146] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam M Shroff. Niser: Normalized item and session representations to handle popularity bias. *arXiv: Information Retrieval*, abs/2105.02501, 2019.

[147] Fan Bai, Jiaxiang Wu, Pengcheng Shen, Shaoxin Li, and Shuigeng Zhou. Federated face recognition. *ArXiv*, abs/2105.02501, 2021.

[148] Divyansh Aggarwal, Jiayu Zhou, and Anil K Jain. Fedface: Collaborative learning of face recognition model. *IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2021.

[149] Haoyuan Gao Lingyun Liu Yifan Zhang. Fedfv: federated face verification via equivalent class embeddings. *Multimedia Systems*, pages 1–11, 2022.

[150] Felix X Yu, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Federated learning with only positive labels. *International Conference on Machine Learning (ICML)*, pages 10946–10956, 2020.

[151] S Lazebnik, C Schmid, and J Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:2169–2178, 2006.

[152] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18:959–971, 1996.

[153] P J Phillips, Hyeonjoon Moon, S. A Rizvi, and P J Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22:1090–1104, 2000.

[154] A S Georghiades, P N Belhumeur, and D J Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23:643–660, 2001.

[155] Zhen Lei, Shengcai Liao, Ran He, Matti Pietikainen, and Stan Z Li. Gabor volume based local binary pattern for face representation and recognition. *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008.