Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Engineering
(33^{rd} cycle)

Deep learning techniques for biometric authentication and robust classification

Arslan Ali

* * * * * *

Supervisors
Prof. Enrico Magli
Prof. Tiziano Bianchi

Doctoral Examination Committee:
Prof. X

Politecnico di Torino
May 25, 2021

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . .

Arslan Ali
Turin, May 25, 2021

Abstract

Despite intensive research in the last years, the biometric authentication problem is still not fully solved; thus, it is not deployed in critical sectors. In this dissertation, we first consider a generic biometric authentication scenario in which the framework can be applied to any biometric trait, i.e., face, fingerprint, iris, and so on. The features extracted from biometric traits are mapped onto a latent space such that authorized and unauthorized users follow simple and well-behaved distributions. We show that, by learning a regularized mapping instead of a classification boundary, higher performance and improved robustness is achieved. Secondly, a deep unconstrained face verification scenario is considered. In the proposed approach, no specific metric on facial features is imposed; instead, the decision space is shaped by learning a latent representation in which matching and non-matching pairs are mapped onto clearly separated and well-behaved target distributions.

The second part of the dissertation focuses on a robust and accurate classification system using deep learning. In recent years, deep learning has shown outstanding performance in several applications, including image classification. However, deep classifiers are known to be highly vulnerable to adversarial attacks, in that a minor perturbation of the input can easily lead to an error. Providing robustness to adversarial attacks is a very challenging task, especially in problems involving a large number of classes, as it typically comes at the expense of a reduced accuracy. We propose a novel approach for training deep, robust multiclass classifiers that provide adversarial robustness while at the same time achieving or even surpassing the classification accuracy of state-of-the-art methods.

Acknowledgements

This work results from the research cooperation with Sony R&D Center Europe Stuttgart Laboratory 1.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Biometric authentication

Today, we live in a digital era – everything is digital – every kind of content – pictures, videos, documents, cards, and so on. Digital technologies have completely changed how we live our lives; everything is just a few taps away. The population of digital citizens is growing with the advancements of digital society. We can do everything online, from grocery shopping to home entertainment. Phones, tablets, and laptops stay with us because within them lies our life's story. Indeed digital technology is making life convenient, but at the same time, a serious question arises; is your digital data safe? If you lose them, what will be the cost?

As the digital world is progressing, the risk of unauthorized access to sensitive data is also increasing, making security even more important. Digital citizens rely more and more on digital devices to store their data, making them a lucrative target for adversaries. Digital adversaries quickly adapt and integrate with the new technologies and are looking for new ways to exploit them. A data breach can lead to myriad of consequences. The adversary can steal sensitive data such as credit cards, money, trade secrets, contact information, and the list goes on.

User Authentication is the process of verifying an entity's identity to determine whether someone or something is what they claim to be. Authentication is the gateway to access valuable data. Weak authentication security can enable adversaries to gain access to sensitive data. The objective of reliable authentication is to ensure that only legitimate users can access a certain system. Such systems include secure access to cell phones, laptops, tablets, ATMs, and buildings.

Unlike machine authentication that uses automated processes, user authentication involves authorizing logins using personal credentials. Personal credentials can be chosen according to three main authentication factors: knowledge, possession, and inherence [1]. The traditional version of user authentication involves a simple knowledge factor like password. However, password verification, has many issues since people tend to reuse the same passwords or write them down, which easily leads to credential stealing, or to forget them, which hinders their usability [2]. With tech companies expanding, it becomes essential to concentrate on user-authentication to validate and secure their identity. Organizations are increasingly opting for advanced user authentication techniques to safeguard and secure customer databases more comprehensively.

Biometric features hold a unique place when it comes to authentication and security applications. Unlike token-based features such as keys and ID cards, they cannot get lost and unlike passwords, they can not be forgotten. The progress of biometric authentication technology has been substantial over the past two decades; advanced authentication methods such as biometrics are finally becoming a reality with deep learning. It is seen as one of the most effective and safe methods of individual authentication. Biometric technology is described as the automatic techniques of recognizing or confirming a living person's identity, mainly centered on a physiological or behavioral trait. Concerning user authentication one of the key advantages of biometric technology is its ability to eliminate the need to use passwords at all [2]. It is experiencing rising acceptance worldwide due to its usability, investment benefits, and future potential. Biometric technology is not new, but in recent years its utilization has become progressively prevalent. The reason for the success of biometric technologies relies on the many advantages they offer over conventional methods. The advantages that biometrics provides are that the information is distinctive for every person, it cannot be lost, and it can be utilized as a technique for individual identification. Biometric technologies can be used not only for user authentication but also to provide privacy or data discretion, authorization or access control, data integrity, and non-repudiation. Despite increased security, efficiency, and convenience, biometric authentication also has disadvantages: Biometric databases can still be hacked and, if stolen, cannot be replaced, biometric devices like facial recognition systems can limit privacy for users, False rejects and false accepts can still occur preventing select users from accessing systems.

Since error rates in biometric systems are shown to have dropped at an exponential rate, one would assume that biometric authentication is becoming a largely solved problem. Unfortunately, this is not the case, as many challenges still remain. The reduction in error rates shown is for biometric traits captured in a controlled environment with cooperative subjects. As an example concerning faces, authentication performance significantly deteriorates when variations in facial pose, facial expression, and illumination (collectively known as PIE) are introduced [2]. Other factors, such as image quality (e.g., resolution, compression, blur), time-lapse, facial aging, and occlusion, also contribute to face recognition errors [3].

User authentication, in general, is a classification problem. The second part of this dissertation focuses on a robust and accurate classification system using deep learning.

1.2 Robust and accurate classification

Over the course of the last few years, deep learning has shown outstanding performance in several applications, including image classification. Neural networks have applications in multiple fields ranging from scientific to industrial products such as biometric authentication and safeguards in the automobile industry. The performance of these techniques has reached similar or even greater accuracy levels than humans in multiple and complex visual and classification tasks [4, 5, 6, 7]. More recently, deep neural networks have also shown remarkable performance at learning complex mappings for image translation, and segmentation [8, 9, 10, 11].

However, the ever-growing use of neural networks in our society raises serious concerns in the matter of *security*, as they can be targeted by malevolent *adversaries*. Several obstacles still hinder their use in fields where security is essential, or if neural networks are used for safety protocols for verification programs such as systems for autonomous driving, biometric authentication, and medical diagnostics [12, 13, 14, 15].

One of the most severe threats to deep learning is represented by *adversarial perturbations*, a collection of methods that are designed to interfere with neural networks input data in order to produce undesired outputs, shift the expected outcome, or more in general, cause algorithm malfunctions and performance reductions. This happens in the face of modifications that are very difficult to detect, to the extent that they are often undetectable to the human eye; human performance usually does not suffer from these infinitesimal modifications. Malicious attackers could exploit such vulnerabilities to cause malfunctions in systems, and the attack would be very hard to detect. One common way of providing defense against such perturbations is to use adversarial samples in the training phase as a particular form of data augmentation to improve robustness [16, 17, 18]. However, such adversarial training does not prevent adversaries from tampering with the final classification stage [19]; it has been proven that universal adversarial perturbations can be crafted to induce the wrong classification with high probability independently of the used dataset [20]. Further, adversarial training is a very time-consuming process, and it comes at the cost of reducing the classification accuracy.

As the number of classes in a classification system increases, providing robustness to adversarial attacks get more challenging as it typically comes at the expense of reduced accuracy. Without robust classification, neural networks can not be deployed confidently in applications where security is essential. As the integration of neural networks in contemporary society grows, they become even more subject to malicious adversaries actions. As the deep learning practitioners are laying attention to develop methods providing robustness to adversarial attacks, at the same time, new techniques are also being developed to craft more successful adversarial attacks [21]. Although many countermeasures have been proposed, an effective defense mechanism against the broad spectrum of adversarial perturbations and maintaining state-of-the-art accuracy is not available yet.

1.3 Contributions

The contributions of this dissertation are two-folds:

1. **Biometric authentication**

   - **Generic biometric authentication:** We demonstrate that mapping the biometric traits (any modality) onto well-behaved target distributions (close to the desired target) leads to higher performance and improved robustness. The mapping of biometric traits onto target distributions can be achieved either through an adversarial game Jensen-Shannon (JS) divergence or statistical approaches like Kullback-Leibler (KL) divergence. Simple and well-behaved distributions enable to employ tunable decision boundaries to make a decision. Extensive experiments on publicly available datasets of faces and fingerprints confirm the superiority of proposed frameworks over existing methods.

     (a) A. Ali, *et al.*, "Authnet: Biometric Authentication Through Adversarial Learning." In: *IEEE 29th International Workshop on Machine Learning for Signal Processing* (MLSP), IEEE. 2019, pp. 1-6.

     (b) A. Ali, *et al.*, "Adversarial Learning of Mappings Onto Regularized Spaces for Biometric Authentication." In: *IEEE Access*, v. 8, 2020, pp. 149316-149331.

     (c) A. Ali, *et al.*, "Learning mappings onto regularized latent spaces for biometric authentication." In: *IEEE 21st International Workshop on Multimedia Signal Processing*, (MMSP), IEEE. 2019, pp. 1-6.

   - **Face verification:** Although the proposed method for generic biometric authentication achieves excellent performance in terms of security metrics like accuracy and is robust against adversarial perturbations, however, the framework is based on a one-vs-all classification scenario. The network needs re-training for every new user, and the trained model is data homogeneous. To solve the problem, we leverage the idea of generic biometric authentication and map it to an unconstrained face verification scenario with one-shot training. The proposed approach does not impose any specific metric on facial features; instead, it shapes the decision space by learning a latent representation in which matching and non-matching pairs are mapped onto clearly separated and well-behaved target distributions. The proposed network jointly learns the best feature representation and the best metric that follows the target distributions, to be used to discriminate face images. Specifically, we propose to use as target distributions two Gaussian distributions with different means and same variance. This choice enables a simple linear decision boundary that can be tuned to achieve the desired trade-off between false alarm and genuine acceptance rate and leads to a loss function that can be written in closed form. Extensive analysis and experimentation on ten publicly available datasets show a significant performance improvement and confirms the effectiveness and superiority of the proposed method over existing state-of-the-art methods.

   (a) A. Ali, *et al.*, " BioMetricNet: deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions." United States Patent US. United States Patent and Trademark Office.

   (b) A. Ali, *et al.*, "BioMetricNet: deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions" In: *Proceedings of the European Conference on Computer Vision* (ECCV), Springer. 2020, pp. 133-149.

   (c) A. Ali, *et al.*, "BioMetricNet: deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions" to be submitted to *IEEE Transactions on Biometrics, Behavior and Identity Science* (TBIOM), IEEE. 2020.

2. **Robust and accurate multi-class classification:** We establish that instead of maximizing the likelihood of target labels for individual samples, learning a mapping of the input classes onto target distributions in a latent space such that the classes have high inter-class separation leads to outstanding performance in terms of classification accuracy and adversarial robustness. The proposed loss, first of its kind, pushes the network to produce feature distributions yielding high inter-class separation. The mean values of the distributions are centered on the vertices of a simplex such that each class is at the same distance from every other class. We show that the regularization of the latent space based on our approach yields excellent classification accuracy and inherently provides robustness to multiple adversarial attacks, both targeted and untargeted, outperforming state-of-the-art approaches over challenging datasets.

   (a) A. Ali, *et al.*, "Beyond cross-entropy: learning highly separable feature distributions for robust and accurate classification." In: *Proceedings of the IEEE International Conference on Pattern Recognition* (ICPR), IEEE. 2020.

   (b) A. Ali, *et al.*, "Beyond cross-entropy: learning highly separable feature distributions for robust and accurate classification."submitted to *IEEE Transactions on Information Forensics and Security* (TIFS), IEEE. 2020.

1.4 Dissertation organization

The remaining of this dissertation is organized as follows:

- **Chapter 2** introduces the general concepts and provides a background on artificial neural networks and biometrics.
- **Chapter 3** presents our generic framework for biometric authentication based on adversarial neural networks. Unlike other methods, our model maps input biometric traits onto a regularized space in which well-behaved regions, learned through an adversarial game, convey the semantic meaning of authorized and unauthorized users.
- **Chapter 4** illustrates our novel model for generic biometric authentication based on deep neural networks. The proposed model learns a mapping of the input biometric traits onto a target distribution in a well-behaved space in which users can be separated employing simple and tunable boundaries using a statistical model.
- **Chapter 5** demonstrates our novel framework for deep unconstrained face verification, which learns a regularized metric to compare facial features. Differently from popular methods, the proposed approach does not impose any specific metric on facial features; instead, it shapes the decision space by learning a latent representation in which matching and non-matching pairs are mapped onto clearly separated and well-behaved target distributions. In particular, the network jointly learns the best feature representation and the best metric that follows the target distributions, to be used to discriminate face images.
- **Chapter 6** outlines our findings for robust and accurate multi-class classification. Differently from other frameworks, the proposed method learns a mapping of the input classes onto target distributions in a latent space such that the classes are linearly separable. Instead of maximizing the likelihood of target labels for individual samples, our objective function pushes the network to produce feature distributions yielding high inter-class separation. The mean values of the distributions are centered on the vertices of a simplex such that each class is at the same distance from every other class.
- **Chapter 7** summarizes our findings and presents directions for future work.
- finally, we include a glossary of acronyms with their definitions.

Chapter 2

Background: general concepts and literature review

2.1 Deep learning

The chapter introduces general concepts, basic terms, and definitions, followed by a state-of-the-art review on specific topics.

In recent years, neural networks (NNs) became increasingly popular in the fields of biometrics, computer vision, and image classification. NNs can learn how to solve different problems directly from the data without requiring hand-crafted features; their tremendous power have led to state-of-the-art performance on many complex tasks. NNs are composed of a stack of layers, each with a non-linear activation function to learn the data representation. The top layers in the hierarchy learn abstract representation compared to lower layers. The layers evolve during the training process to exploit their ability to approximate arbitrary non-linear functions [22]. During the training, these layers learn how to compute highly discriminative features that can be used to classify the input data.

Increasing demand for reliable security systems has led to an increased interest in biometrics-based authentication systems. Deep learning has completely reshaped this research; biometrics-based research has been dramatically shifted away from the holistic learning and hand-crafted based approaches towards the NNs based models. In the remainder of this chapter, the concepts related to deep learning are introduced.

Artificial neuron

A neuron is the basic unit of a neural network (Fig. 2.1). It is a simple predictive function inspired by biological neurons. An artificial neuron transforms the inputs to an output y in two steps: Affine transformation (weighted sum of input) followed by a nonlinear transform. An affine transform is written as:

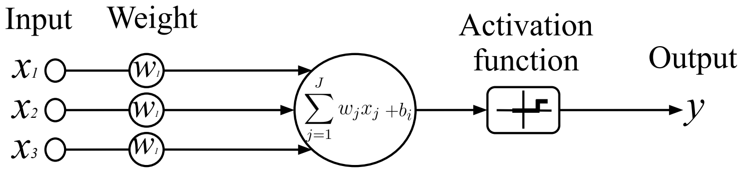


Figure 2.1: An illustration of artificial neuron.

$$z = b + \sum_{j=1}^{J} w_j x_j \tag{2.1}$$

In equation 2.1, w_j is the weight used on the features x_j, and b is the bias. Both weights and bias are parameters of a neuron. Neuron output is given by $y = f(z)$. Here f is a linear or nonlinear function known as the activation function. If f is the identity, a neuron is equivalent to linear regression model. When f is a nonlinear function a neuron is able to represent more complex regression models. For example, if f is the sigmoid or logistic function, i.e., $f(z) = 1/(1 + e^{-z})$, the neuron behaves like a logistic regression models. In this case it is important to notice that the $z \in R$ is squashed to an output $y \in [0, 1]$. For binary classification, often one desires to have an output representing the probability that the input belongs to a certain class; a neuron with a logistic activation function is usually a convenient choice.

Loss Function

Given a training set of N labeled sequences, the aim is to learn a prediction function $f(z)$ with parameters w and b that optimizes a loss function. Let x_i denotes the i^{th} training sample, y_i be the ground truth label and $\hat{y}_i$ be the prediction for i^{th} training example. A standard choice for the linear regression loss function is the square loss:

$$J_{SL}(\hat{y}_i, y_i) = (y_i - \hat{y}_i) \tag{2.2}$$

In equation 2.2, y_i is the target label, and $f(x_i)$ is the predicted label for a given input sequence x_i. The loss function for the complete training set is an average of loss function across all training examples. In the case of logistic regression having binary output labels, the loss function is called binary cross-entropy:

$$J_{CE}(\hat{y}_i, y_i) = y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{2.3}$$

In equation 2.3, y_i is between 1 and 0. Given a sequence x_i, $\hat{y}_i$ is the predicted output of the activation function and represents the conditional probability of the class labeled as 1 when x is observed.

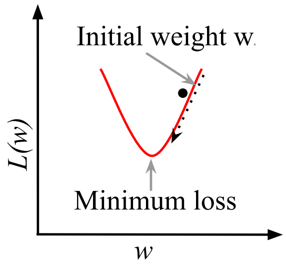


Figure 2.2: Reducing loss: gradient descent.

Learning algorithm

A learning algorithm optimizes network parameters to minimize the loss function. For a given problem, given a set of N training samples with corresponding labels (x_i, y_i) and a loss function $J(y_i, f(x_j, w))$, w is the parameter of the neuron optimized by the learning algorithm to minimize the loss function; this process is known as model training. One commonly employed algorithm is gradient descent [23]. It starts with an initial guess of weights $w(0)$, and is systematically updates them based on the training data to minimize the loss function. For a given pair (x_i, y_i), the loss function $J(w)$ is interpreted as a function that maps the weights values to the loss values. For differentiable loss functions like square distance and cross-entropy, the gradient of the loss function $J(w)$ computed at a specific value of the weights is used to determine the direction of the fastest decrease for the loss (Fig. 2.2). Repeated passes are made through the training set, iteratively updating the weights to minimize the loss function. If $w(t)$ represents the parameters at a specific iteration t, we have:

$$w(t+1) = w(t) - \eta \frac{1}{N} \sum_{i=1}^{N} \frac{\partial J\left(y_i, f(x_j, w(t))\right))}{\partial w} \tag{2.4}$$

Where η is the learning rate, it controls the update step in the opposite direction to the gradient. Training is terminated once the loss gets stable. For small η, it takes a long time for the training to converge; on the other hand, too large values on η can cause the algorithm to behave erratically since the gradients can no longer predict the change in the loss function accurately.

The training set usually includes a very large number of samples; computing gradient for each example in each iteration can be infeasible. To handle this, the training set is split into small batches of size in the range $16 - 512$ samples (referred to as minibatches); in each training iteration, the average gradient for minibatch is computed. As training progresses, gradients across the training examples are repeatedly calculated until the loss converges to a minimum value.

An epoch refers to the number of iterations to use all samples in the training set once and only once. The length of an epoch depends upon the minibatch and

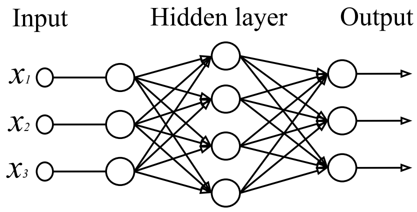


Figure 2.3: The anatomy of Feed-Forward neural network.

the training set size. The aforementioned algorithm is also known as stochastic minibatch gradient descent.

Training and test sets - Overfitting and Underfitting

The training set refers to labeled examples used to fit the parameters of the model. The model's performance on the training set is typically higher than on never seen before samples. Patterns can exist in the training sets that are simply the artifacts of the noise in measurements. If the model leverages these artifacts while training, the classifier goes astray on the new examples. Therefore, the training set's performance can be misleading; the situation is referred to as *overfitting.*

Thus a model's performance should always be evaluated on a set of images the model has not seen during training [24]. This never seen before dataset is referred to as the test dataset. The model generalization is the capability to perform well on test sets different from the training set. On the contrary, if the model is not powerful enough to capture the training data patterns, it is known as *underfitting.*

Regularization

One way to overcome the *overfitting* problem is to avoid using overly complex models; this is termed as *regularization* [24]. Penalizing the weights of the model in the loss function helps to regularize the model. A model with small weights makes it insensitive to minor changes in the input and generalizes well on the never seen before datasets. l_1 and l_2 regularization are among commonly employed methods. l_1 regularization adds l_1 norm of the weights as a penalty term and encourages the algorithm to use relatively few large weights and many small weights. Similarly, l_2 regularization uses the l_2 norm of the weights as a penalty term and encourages weights with smaller magnitudes.

Validation set

As the classification model's performance differs for the training and test sets, a subset of the dataset from the training set referred to as the validation set is set aside to make some design choices. Hyperparameters are explored and tuned

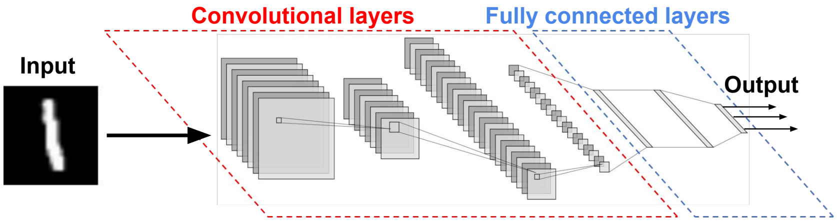


Figure 2.4: The anatomy of convolutional neural network.

on the validation set, and finally, the model's performance is reported for the test dataset.

A deep multi-layer neural network

A single neuron may work well for several purposes but is limited as it consists of a single linear transformation followed by the non-linearity. Stacking multiple layers of neurons, where each neuron takes input from previous layers dramatically increases the expressive power of the model. Each layer transforms the inputs into increasingly complex representations. The deep neural network's final layer is either a logistic or linear neuron that operates on the penultimate layer's output. Neural networks are a powerful extension of linear and logistic regression models that operate on non-linear, hierarchical transformations of the original features learned from the training data.

Feed-Forward (FF) Neural Networks

Feed-Forward neural networks or multi-layer perceptron are the most basic types of neural networks. Each node in the previous layer is connected to each node in the subsequent layer (Fig 2.3). During training, labeled examples are fed to the model, which predicts its output and updates the network's parameters by propagating the loss gradients with respect to the ground truth labels backward through the entire network. The weight values of the hidden layer h are computed using $h = f(\sum_{j=1}^{N}(w_j x_j) + b_h)$. Input feature x_j is multiplied with the weight w_j and aggregated with the bias term b_h, and finally, a non-linear activation function f is applied.

Convolutional Neural Networks (CNN)

The problem with fully connected networks is that the number of parameters grows in a exponential way with the number of layers and the size of the inputs. With CNN, we can reuse the same parameters for different locations of the input, such that the number of parameters does not depend on the size of the input.

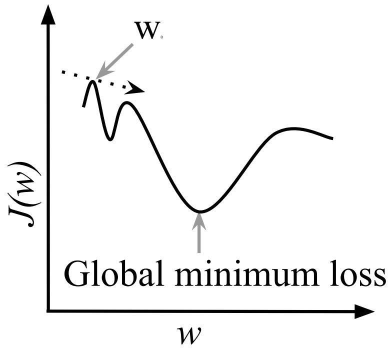


Figure 2.5: A non-convex loss function of a non-linear DNN contains multiple local minima.

CNN employ convolutional layers with neurons called filters that detect the local patterns of the raw inputs (Fig 2.4). Each filter of the CNN scans the input to find a particular pattern encoded by the filter weights. The convolutional layers are stacked such that the filters in the subsequent layers detect complex patterns on top of filters in the previous layers. Finally one or more fully connected FF layers integrate the entire patterns detected across all positions to predict an output label.

Training the network

Neural networks are generally trained with the stochastic gradient descent method. In the gradient descent algorithm, the loss gradient with respect to the parameters is calculated for the minibatches. The parameters are adjusted to reduce the loss. The process is repeated until the performance on the validation dataset no longer improves. The process of terminating the training when the validation set performance no longer improves is referred to as early stopping. Commonly employed loss functions include cross-entropy for classification and mean squared error for the regression.

Gradient descent algorithm is sensitive to the weights initialization; good weights initialization leads to higher performance, and poor weights initialization may fail the network to converge. A general rule of weights initialization is that the activation functions should have zero mean and unit variance. Network's weight initialization depends on network choice and activation functions employed. Some effective solutions for weight initialization are listed in [25, 26, 27].

For multi-layer NNs, the loss is high dimensional and non-convex with multiple hills and valleys (Fig. 2.5); consequently, the optimization gets more complicated. The learning rate is decreased over epochs with gradient descent to allow the algorithm to explore different loss function contours. Another technique that can help the network get out of local minima is the use of a momentum term. The momentum term's choice leads to large gradients over several iterations in the direction that has a consistent gradient and small gradients in the direction where gradients are changing over iterations. Learning rate and momentum are the parameters that must be optimized. Several optimizers have been proposed to make the choice easy [28, 29, 30, 31, 32]. A comprehensive analysis is presented in [23]. These optimizers adapt the learning rate for each parameter according to the magnitude of the past gradients.

Efficient computation of gradients - backpropogation algorithms

The computation of gradients of loss with respect to the network's weights is of key importance to train a NN with a stochastic gradient descent algorithm. The backpropagation algorithm [33] efficiently computes the gradients using the chain rule $\frac{\partial J}{\partial w} = \frac{\partial J}{\partial z} \cdot \frac{\partial z}{\partial w}$. The backpropagation algorithm starts from the output layers, and back through intermediate layers propagates to the input layer. The gradients of all individual parameters of the network (weights) can be computed by employing this information. This information is utilized to update the parameters using a stochastic gradient algorithm.

Activation functions

Without employing nonlinearities, a DNN can only describe linear functions, as a linear function of a linear function is still linear. The choice of nonlinearities strongly influences the convergence properties of NNs. One commonly employed nonlinearity is logistic function. Logistic function often suffers from the vanishing gradient's problem when the input is too large or too small. Generally, logistic function is used when the output is to be constrained between 0 and 1, i.e., for binary classification tasks. Networks trained with ReLU nonlinearity $f(z) = \max(0, z)$ do not undergo this problem. ReLU is a piecewise linear thresholding function that sets all negative values to 0. Various alternatives to ReLU exist, such as Leaky ReLU [34], PReLU[35], ELUs[36], and SELUs [27], etc.

Regularization

To avoid overfitting, weight penalties like ℓ_1, ℓ_2 regularization are often applied to the loss function. These penalty terms can be applied either to the neurons' activation or the networks' weights to encourage sparse solutions. These techniques also help prevent the weights from getting stuck near zero or exploding to large values. Another very commonly employed solution is dropout. The main idea behind dropout is to randomly zero-out or drop a fraction of the neurons' activation in a layer p for every training step. At the prediction time, no dropout is employed.

Batch normalization

Batch normalization [37] ensures that while training, each layer's input has roughly the same mean and variance within minibatches. It combats the covariate shift problem when the internal parameters are updated. This is done by explicitly learning optimal values for mean and variance of the distribution and normalizing the minibatches explicitly to achieve this. Batch normalization accelerates and stabilizes the networks' training and acts as a regularizer, reducing dropout dependency. Recently SELU [27] activation's have exhibited a similar effect by naturally resulting in the layers with zero mean and unit variance. Another alternative to batch normalization is layer normalization [38]; it performs normalization based on individual training examples rather than complete batches.

2.2 Biometrics

Reliable biometric authentication is a crucial property required by a wide variety of systems. The objective of such schemes is to ensure that legitimate users solely access the protected system. Applications of biometric authentication systems include secure access to cell phones, laptops, ATMs, and buildings. Without reliable authentication methods, such systems are vulnerable to the wiles of imposters.

In traditional systems, the most common ways of restricting access include card-based authentication and passwords. These authentication methods are not fool-proof. Once the password is divulged to an unauthorized user or the card is stolen, this usually leads to security breach. Additionally, simple passwords can be guessed, hard passwords are difficult to remember, and the cards can be stolen and lost.

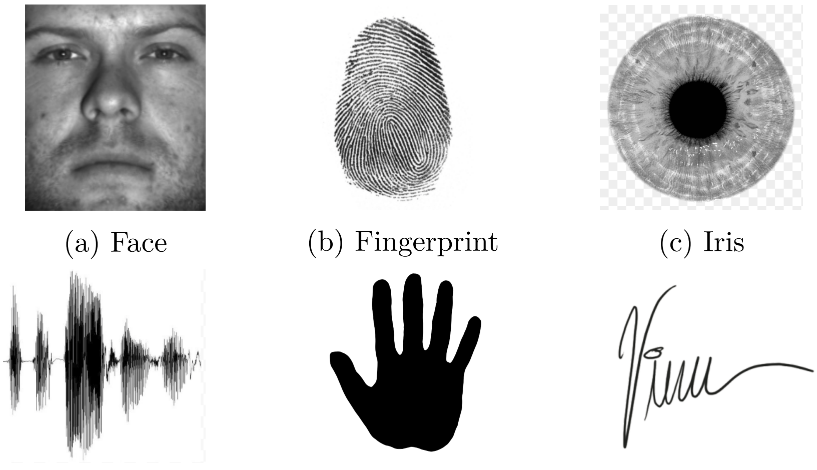
The problems beset by traditional verification systems are addressed by the evolution of biometric-based methods. The use of physical or behavioral characteristics (face, fingerprint, iris, signature, etc.) to automatically authenticate or recognize an identity are referred to as biometrics or biometric system. A biometric trait can be either physical or behavioral, which is measurable, repeatable, and distinguishable. A biometric trait must have the following four characteristics [39].

- Universality: Most individuals should have it
- Distinctiveness: It should be unique
- Permanent: It should not be varying over time
- Collectible: It should be possible to measure it quantitatively

These characteristics make the use of biometrics a reliable way of authentication. Further, the traits can not be shared, transferred, guessed, or lost like a traditional card or password-based authentication, leading to increased user convenience. Further, it leads to an improved authentication accuracy, as the system can be tuned to minimize the unauthorized accesses. Also, the cost of employing biometric systems is continually decreasing.

Verification refers to the process where the user claims an identity *(I)*, and the system provides an output decision, which is either yes or no. Conversely, in the identification process, the system automatically searches the whole database of enrolled users to find if there is a match or not. In the output, the identity of the users I is provided.

The verification problem can be modeled as follows: given an input feature vector X_Q, and the corresponding claimed identity I, it is to determine whether (I, X_Q) belongs to P_1 or P_2. P_1 shows that the requested claim is valid, i.e., the user is genuine, whereas P_2 indicates that the requested claim is false, i.e., its an



(a) Face (b) Fingerprint (c) Iris

(d) Voice (e) Hand (f) Signature

Figure 2.6: Sample biometric traits: (a) face, (b) fingerprint, (c) iris (d) voice signal (e) hand geometry (f) signature.

impostor attempt. The input feature vector X_Q is matched against X_I, which corresponds to stored identity I, to determine its category.

$$(I, X_Q) \in \begin{cases} P_1 & \text{if } \mathcal{S}\left(X_Q, X_I\right) \geq \tau \\ P_2 & \text{otherwise} \end{cases} \tag{2.5}$$

In equation 2.5, $\mathcal{S}$ determines the similarity between X_Q and X_I, where τ is the predefined threshold. Verification can be seen as a one vs. all class classification problem, where each claimed identity is classified either as P_1 or P_2 based on X_I, I, X_Q, the threshold τ, and the function $\mathcal{S}$.

On the other hand, in the identification problem, the aim is to determine the identity of a person. For a given input feature vector X_Q, the aim is to determine if the identity I_k where $k \in \{1,2,\ldots N, N+1\}$ corresponds to a specific identity enrolled in the database $I_1, I_2 \ldots I_N$. Usually a fictitious identity I_{N+1} is added to represent the case in which no identity can be determined. Thus,

$$X_Q \in \begin{cases} I_k & \text{if } \max_k \left\{\mathcal{S}\left(X_Q, X_{I_k}\right)\right\} > \tau, k = 1,2,\ldots N \\ I_{N+1} & \text{otherwise} \end{cases} \tag{2.6}$$

In equation 2.6, X_{I_k} is the template corresponding to identity I_k, where τ is the predefined threshold.

Biometric system modules

Generally, a biometric authentication system consists in the following four essential modules:

- **Sensor module** captures the user's biometric data, such as the fingerprint scanner capturing fingerprints and the camera module capturing individuals' face images.

- **Feature extraction module** processes the acquired data and extracts the most discriminative features. For example, in the case of fingerprints, the position and orientation of minutiae points are computed.
- **Matching module** matches the feature values against the stored template and generates a matching score. For example, in the fingerprint system, the matching minutiae of query and template are computed and treated as the matching score.
- **Decision-making module** makes the final decision; the claimed identity is either accepted or rejected based on the matching score in verification or identified as a user in the database for identification.

For example, in the case of fingerprint-based authentication schemes, initially, during the enrollment phase, the user places his finger F on the sensor. Feature extractor extracts the discriminative features from the sensors output F_s to generate the template F_t. The generated template F_t along with the identity of the user is stored in a database. During the verification phase, the fingerprint of the user is captured again. Template $F_{v,t}$ is generated from the captured fingerprint and is matched against the stored template F_t in the database along with the stored identity I. If the two templates are "close enough", the authentication system grants access to the user. Generally, for matching the templates, a similarity measure is used; if the templates' similarity score is lower or higher than a specific threshold T, the access is granted, else the access is denied. Conversely, during the identification, the online generated template F_i is compared with all templates in the database. If there is a match, the authentication system outputs the corresponding associated identity I of the user.

False Acceptance Rate (FAR) and False Rejection Rate (FRR) metrics are commonly employed to quantify the authentication accuracy. FAR is the probability of an imposter being accepted by the system. Contrary, FRR is the probability of a legitimate user being rejected by the user. The point where false acceptances are equal to false rejections, i.e., FAR=FRR, is denoted by Equal Error Rate (EER). Another commonly used metric is the Genuine Acceptance Rate (GAR), which shows the probability of accepting legitimate (genuine) users. It should be noted that GAR=1-FRR. The metrics mentioned above depend on the selected threshold T. The labels corresponding to the thresholds can be added, e.g., T_1 for FAR_1 and GAR_1, similarly T_2 for FAR_2 and GAR_2. By varying the thresholds $T = T_1, T_2 \ldots T_k$, multiple (k) operating points for a system can be obtained. As a result, the GAR vs. FAR plot obtained by varying the thresholds is called the

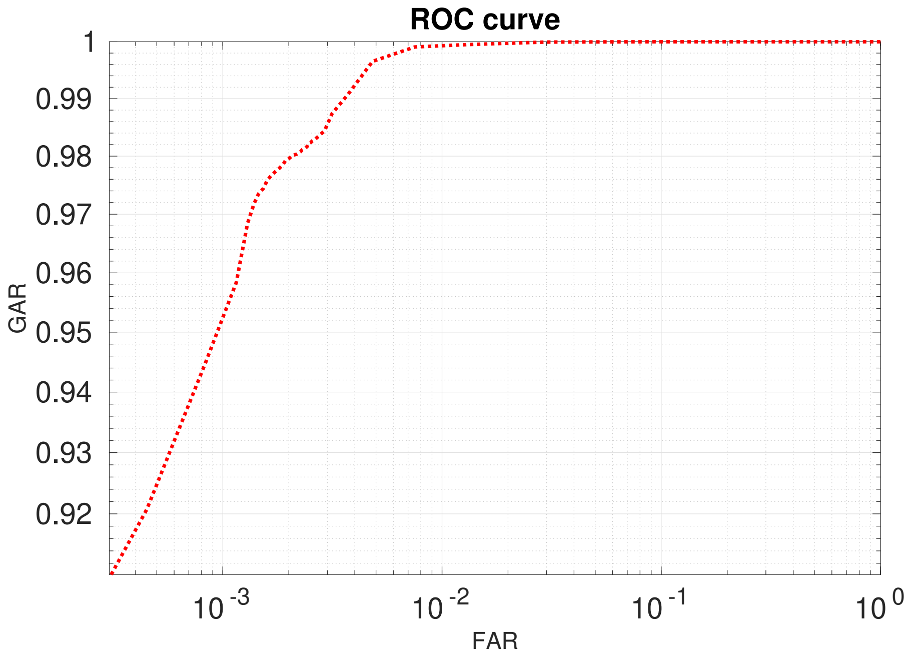


Figure 2.7: A typical ROC curve for biometric verification system

Receiver Operating Characteristics (ROC) Curve. ROC curve is very commonly used to evaluate the performance of an authentication system. Fig. 2.7 depicts the ROC curve obtained through a verification system. One of the sample operating point (GAR,FAR) = (0.95,0.001) shows the GAR corresponding to a given FAR. The designer of the authentication system analyzes the ROC curve according to the requirements, (e.g. for user convenience the GAR should be greater than 0.95, whereas to assure the security of the system, the FAR should be less than 0.001) and decides which operating point should be used.

Despite the advantages of biometric authentication systems compared to traditional card-based and password-based authentication systems, several unresolved problems are still present. A major concern is the authentication accuracy, i.e., verification and identification accuracy brought by biometrics-based systems. An increase in the authentication accuracy automatically brings an improvement in the security of the systems. Apart from authentication accuracy, many other issues should be accounted for before deploying a biometric-based authentication system.

One big concern that hinders the deployment of biometrics based authentication systems is the robustness to adversarial attacks. For traditional authentication systems, the parameters for security analysis are much more straightforward. For example, for passwords one can consider the minimum length, the frequency with which the password is updated, the set of allowed characters and have a reasonable estimate of the difficulty of guessing a password.

Similarly, the security of the card-based authentication systems can be analyzed

Biometric trait	Universality	Distinctiveness	Permanence	Collectability
Face	H	L	M	H
Fingerprint	M	H	H	M
Iris	H	H	H	M

Table 2.1: Comparison of various biometric characteristics. High, medium, and low values are denoted by H, M, and L, respectively.

based on the illegal utilization of the card. The feasibility of replicating or illegally mimicking the card's characteristics, e.g., how easy it is to replicate a signature close to one present on the database, is hard to estimate.

One generic problem within the biometrics-based system's scope is the robustness against adversarial attacks, i.e., guaranteeing the security against perturbed input traits. For example, during enrollment of fingerprint images, the samples are stored in a central or local database as a user template. During the verification, the newly generated template is compared with the stored template in the database. Verification is successful if the similarity score exceeds a pre-specified threshold.

Everything comes with a price; biometrics-based authentication systems are intrinsically more complicated than traditional authentication systems. E.g., with each acquisition, the biometric data of individuals vary by some amount contrary to true/false for a password or card-based authentication. Further, complex enhancement modules may be required to improve the quality of the biometrics traits. Further, overall biometric authentication architecture should securely access the stored biometric traits (stealing of biometric traits). This leads to several critical points that may get compromised, which are absent in traditional authentication methods [40].

Many of the currently existing applications and frameworks are tightly designed for traditional authentication systems. Integrating biometrics-based authentication systems into existing systems is also important. This is due to the independent development of biometrics-based and traditional authentication systems. Fusing biometric components in existing systems bring several problems that need to be solved, e.g., variation in biometric data with time.

2.3 Common biometric traits

As the trend shifts from traditional authentication to biometrics-based authentication, several biometrics traits have recently been used for authentication and recognition, Fig. 2.6. Examples include the face, fingerprint, iris, signature, voice, palm print, ear shape, gait, keystroke, etc. [41]. Among all, face and fingerprint are the most prominent ones used in this dissertation.

The biometric trait choice is usually directed by the target application, with its strengths and weaknesses. Tab. 2.1 summarizes the pros and cons of three

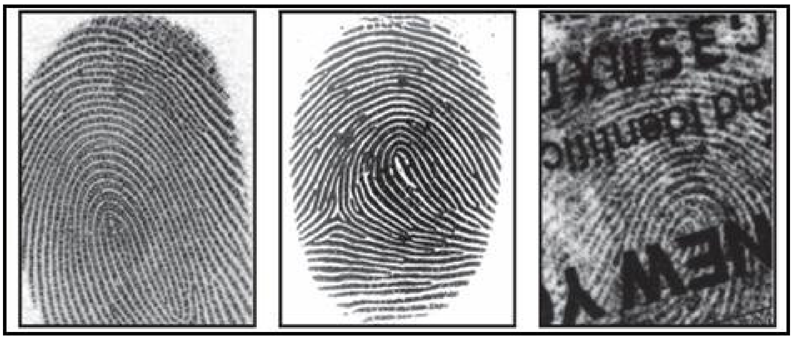


Figure 2.8: Types of biometrics: (a) rolled fingerprints, (b) plain fingerprints, (c) latent fingerprints [42]

prominent biometric traits, i.e., face, fingerprint, and iris [39]. Four prominent characteristics, i.e., universality, permanence, distinctiveness, and collectability, are analyzed. For example, it is easy to collect face images; hence, the face's collectability is high. Similarly, fingerprints are considered very distinctive; the ridge structure does not change much over the years; even after the cuts on fingers, the fingerprint pattern reappears after the healing process. On the other hand, even though it is easy to acquire fingerprints, it needs some degree of cooperation from the user (medium collectability). Additionally, with the time, in aged people, the fingerprint pattern can become less evident and make automatic authentication or recognition unstable.

The face is the most convenient trait to acquire and does not need much cooperation for acquisition. It can be acquired even from a distance with or without the user's consent. This property is highly useful in applications like video surveillance. On the other hand, recognition becomes complicated with the variations in face images like pose, lightning conditions, makeup, expressions, etc. [41]. Additionally, over time, the characteristics are not very stable, such as weight gain, aging, etc.

Iris images are complicated to acquire; sophisticated and expensive sensors are used for the acquisition. Further, it needs considerable cooperation from the subject, as many factors can influence the acquisition like partially closing eyelids, wearing lenses or eyelids, etc. Therefore iris recognition is not commonly used in handheld devices like mobile phones, tablets, laptops, etc.

The next section presents a review of the representation and matching of fingerprint and faces biometric modalities.

2.3.1 Fingerprints

The fingerprint is an impression of the friction ridge skin on the tip of the finger. Friction ridge skin presents raised ridges because their function is to grip and grasp. Fingerprints are different both across different fingers of the same person and across different persons [43]. For matching, there are essentially three kinds of fingerprints acquired:

1. *Plain*, obtained by placing a finger flat on paper or on the plate of a scanner
2. *Rolled*, its obtained by rolling a finger from nail-to-nail on a paper or fingerprint scanner
3. *Latent*, typically obtained from crime scenes, recovered from objects accidentally touched by an individual

Representation

The fingerprint structure is composed of ridges and valleys. In Fig. 2.8a, the ridges are the dark areas of the rolled fingerprint and correspond to the finger's raised ridges. Valleys correspond to the space between the ridges. In Fig. 2.8a, valleys are the bright areas. Fingerprint characteristics are categorized into three different levels from coarse to fine details: level 1 - ridge flow, level 2 - minutiae, level 3 - pores, dots, ridges, etc [44].

- **Level 1:**
  It is the most coarse level representation of the ridges. It consists of orientation and frequency map of the ridges of the fingerprint. The local orientation of a ridge at a point (x, y) is defined as the angle of the line tangent to the ridge at (x, y), and is in the range $[0, \pi)$. The local ridge frequency at a point (x, y) is the average number of ridges crossing a segment of unit length centered at (x, y) and normal to ridge orientation [44].

- **Level 2:**
  Ridges often exhibit discontinuities in several ways. The location of the discontinuity is called minutiae. The most commonly accepted discontinuities are the points where ridges end or bifurcate. A minutia is usually represented by its location, orientation, and type (bifurcation or ending). Its permanence and ease of representation make it the most commonly employed feature for fingerprint matching.

- **Level 3:**
  Level 3 constitutes the micro-level characteristics of fingerprints. Level 3 includes micro features like sweat pores, incipient ridges, dimensional ridge attributes, like width or shape of ridges. These features are usually observed in low resolution and are extremely important for latent fingerprint examiners, especially when minutiae is too small. For automatic fingerprint recognition, level 3 features are rarely used as their extraction is computationally very expensive and they are not as reliable as level 1 or level 2 features. Level 3

features are mostly used as additional evidence by forensic experts, as they often need additional information to match latent fingerprints.

Matching

Most of the published algorithms for fingerprint matching are based on the minutiae. Few of them are based on image correlation [45, 46]. In such cases, features are mostly the intensities of the pixels. For computing correlation between images, fingerprints are aligned globally or locally. Additionally, two images of fingerprint impressions might appear very different due to variations in pressure (ridge/ valley thickness, global structure, contrast, etc.), significantly affecting the correlation between two images. Often the methods used in this category are computationally very expensive.

Apart from this, there are several other feature-based methods, where features include singular points, texture information, level 3 features, etc. These methods are often used in combination with minutiae-based methods to improve performance. They are also used in cases where it is challenging to extract minutiae, as it may happen with latent fingerprints. However, non-minutiae-based features are not as distinctive as minutiae ones and can only be used in conjunction with minutiae.

As mentioned, most commonly employed fingerprint matching algorithms are based on minutiae. There are three main steps in fingerprint matching using minutiae; alignment, pairing, and score computation. Alignment estimates, the parameters that are used to transform one set of minutiae to match the same coordinates as a second set of minutiae. Pairing is the process of finding the corresponding minutiae. Finally, in score computation, a matching score is assigned to a fingerprint-based on the number of corresponding minutiae.

2.3.2 Faces

A face is composed of skull characteristics, musculature, and associated soft tissues. Usually, with age and gender, these features variate. Challenges in Face authentication include variations in pose, lightening, occlusion, expression, weight variation, cross-age, makeup, etc.

Representation

The first step in Biometric authentication is face detection. In face detection, the location of the face is determined. This process is solved using a two-class classification problem: face vs. no-face. After face detection and localization, facial features can be extracted. Usually, facial images are aligned based on eye positions and are normalized for the size and illumination before the matching process. Similarly to fingerprints, facial features are categorized into three levels [47, 44].

- **Level 1** features capture global facial characteristics like the geometry of the face. These coarse features can be extracted from the low-resolution image and can quickly distinguish between different face shapes like elongated or round [48].
- **Level 2** features consists in features corresponding to facial characteristics. Examples of Level 2 facial features include the structure of facial components like mouth, the spatial relationship between the face's components, etc. Similar to Level 2 features of fingerprints, they are the most essential features for facial recognition.
- **Level 3** features consists in It constitutes minor facial details, like moles, scars, freckles, etc.

Matching

There are three main approaches for matching facial images: (i) appearance-based, (ii) model-based, and (iii) texture-based [49]

- **Appearance-based approach:** In appearance-based techniques, the face image is mapped into a lower-dimensional subspace. The representative vectors are learned based on the training set of facial images. Examples include Principal component analysis (PCA) [50] and Linear Discriminant Analysis (LDA) [51]. In these approaches, the face image is represented in terms of the learned basis vectors, as a weighted sum of the basis vectors. Test images can be compared with the reference images by comparing the associated weights of the test images with the reference images in the database by computing the Euclidean distances between the two, which is a measure of dissimilarity between the two images. LDA and PCA's difference is that LDA incorporates the class information during the training stage (supervised learning), whereas PCA does not and is an unsupervised technique. In the PCA approach, the data is projected with an objective of maximizing the variance. In the LDA approach, the data is projected in a way that the inter-class/intra-class variance ratio is minimized.
- **Model-based approaches:** These approaches refer to the use of face models. Graph matching is an example of such approaches. In graph matching, the face is represented based on a model graph. In the model graph, the face's landmarks are associated with the graph's nodes. The graph is fitted to a face to generate a representation of the face. The graph contains a set of local descriptors (bunch) at each fiducial point to account for the variations in the

face images' neighborhood. For each fiducial point in the query face image, the local descriptor is extracted to compare the stored model's descriptor.

- **Texture based approaches:** Texture-based approaches employ local features such as Scale Invariant Feature Transform (SIFT) or Local Binary Patterns (LBP). These local features are extracted for pre-specified points, and the feature vectors are generated for these descriptors. These feature vectors are further compared to generate a similarity score. SIFT is a histogram of the gradient orientations in the neighborhood, whereas LBP represents a relationship between the neighborhood pixels' intensities.

Chapter 3

Adversarial learning of mappings onto regularized spaces for biometric authentication

In the following two chapters, we consider a generic biometric authentication scenario in which the framework can be applied to any biometric trait, i.e., face, fingerprint, iris, etc. The features extracted from biometric traits are mapped onto a latent space such that authorized and unauthorized users follow simple and well-behaved distributions. We demonstrate that mapping the biometric traits onto well-behaved target distributions leads to higher performance and improved robustness. We propose two different methods to shape the latent space according to the target distribution. The mapping of biometric traits onto target distributions can be achieved either through an adversarial game (JS-divergence) or statistical approaches like (KL-divergence). In this chapter, we propose AuthNet based on adversarial training, whereas in the next chapter, we propose RegNet, based on statistical approaches. We further show that simple and well-behaved distributions enable to employ of tunable decision boundaries to make a decision.

3.1 Introduction

Recently biometric authentication systems started drawing increasing attention, thanks to their convenience; the users are authenticated based on information they inherently own, avoiding the need to remember passwords or provide keys. The most common approach followed by such systems is template matching. The biometric traits are associated with a template that captures the trait's most discriminative features in template matching. As a result, all templates of a specific biometric trait belonging to the same user lie within a suitable distance metric. Once a suitable biometric trait that can be a face, fingerprint, or iris is captured through a suitable sensor, it is processed to obtain the corresponding template securely. During the enrollment phase, the system is prepared to grant access to the enrolled users. In the verification phase, the pre-trained system used to enroll the users is employed. New biometric traits of users requesting authentication are acquired and matched against the associated templates of the stored users. Depending upon the matching process's outcome, a user is either granted or denied access to the system.

The extracted features have to be the most discriminative ones and should be embedded in a well defined metric space to enable the template matching process. This is one of the most critical parts of a biometric authentication system and greatly affects authentication accuracy. Traditionally, handcrafted design was employed to extract the features. On the other hand, deep learning-based methods have the great advantage of learning the best features directly from the data due to their ability to learn complex mappings [52, 9] and addressing difficult classification tasks [5].

In deep learning-based approaches, the biometric authentication problem is generally addressed by learning feature embeddings in a way that the extracted template represents the most discriminative features of a specific trait in a suitable metric space. Similarly to the traditional biometric authentication systems, the discriminative features learned during training are shared among the users, and the template matching is based on the distance between the embeddings. In AuthNet, we follow a different path by relying on a classification-based approach. The neural network learns the decision boundaries that can discriminate a specific user from every other user.

Classification based approaches require per-user training; they trade off the added complexity with the improved user-specific features. It is essential to mention that the embedding-based network training process requires a considerable amount of labeled data, as the network has to learn generic features of the data class. Classification based approaches do not suffer from this issue as the network is specifically trained for that specific user for which the most discriminative features are to be learned. conversely, embedding-based approaches learn specific features of the considered class, e.g., faces, and may fail on a specific user.

In this regard, it is essential to emphasize that deep learning-based classification learns highly non-linear boundaries with complex shapes to partition the feature space [53]. In [53], it is shown that the decision boundaries significantly affect the robustness of the classifier. More particularly, it is demonstrated in [54], that most of the data points gather near the decision boundaries; as a result, two similar biometric traits may get assigned to different classes leading to an error. Furthermore, this undesirable behavior is an intrinsic property of the classifier and does not depend on the input data [54].

For the above-mentioned reasons, in AuthNet, a novel *user-specific* classification

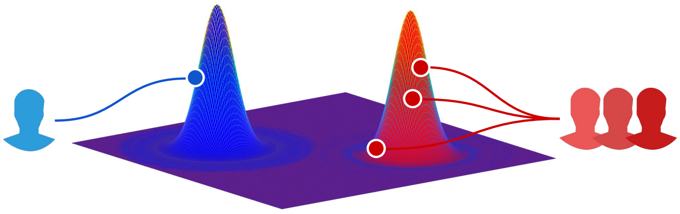


Figure 3.1: The goal of AuthNet is to map the input biometric traits onto target distributions in the latent space. Authorized users (blue) are mapped to a target distribution whose mean value is far from that of the unauthorized users (red).

strategy is proposed that does not enforce the network to learn complex classification boundaries. Rather, a network design that learns to map the input biometric traits onto a regularized and well-behaved latent space is proposed. The feature distributions are regularized, leading to tunable and straightforward decision boundaries between the classes, reducing the probability of misclassification. More particularly, the aim is to obtain "non-arbitrary" boundaries that can improve accuracy and robustness.

The first step comprises learning a compact and meaningful mapping of the input biometric traits onto well-defined distributions in the latent space. Ideally, latent space should be shaped so that the authorized and unauthorized users are clustered in two distinct and compact regions leading to regular boundaries. A decision is then made employing a linear decision boundary discriminating authorized users from everyone else. In AuthNet, to enforce proper shaping of the latent space, adversarial training is employed. Later in the chapter, we provide an in-depth discussion of how AuthNet correctly maps users misclassified by other approaches. We further motivate a higher misclassification rate by competing methods. AuthNet comes in two different flavors, based on ResNet [55] and DenseNet [56]. We provide a detailed performance comparison of the average values of the considered metrics computed independently on each user and aggregated for all users. We further provide a detailed analysis of the robustness of the approach when tested on new datasets that the network has not seen during the training and in the presence of targeted perturbations and verify how the regularization of the latent space leads to robust authentication compared to traditional classification approaches. Finally, we add a discussion on the choice of optimal system parameters.

3.2 Background

Different methods have been proposed to address the biometric authentication task over the years when dealing with different biometric traits such as faces, fingerprints, retinas, and gait. With this work, we specifically focus on the most

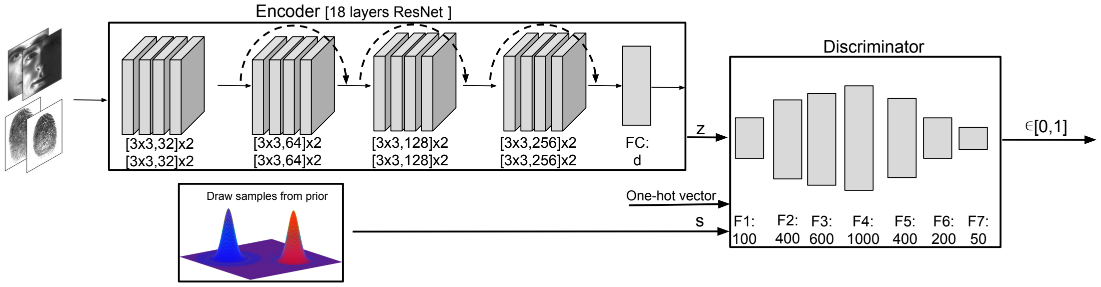


Figure 3.2: AuthNet-R architecture at enrollment phase. Training biometric traits are given as input to the encoder which consists of an 18-layered residual network followed by a fully connected layer. The output of the encoder, together with a one-hot vector and samples of the target distributions, is given as input to the discriminator which is made of 6 fully connected layers.

extensively employed biometric modalities, namely face and fingerprint.

3.2.1 Faces

Recently faces have emerged as a practical biometric modality. Formerly this trait was considered challenging due to its inherent difficulty in handling far from ideal acquisition conditions. Indeed, the traditional approaches exhibit a high variance to the poses and illumination conditions. Eigenface [50], a pioneer in this sense, is a well-known approach; the features describing the faces are obtained by projecting the test images onto space spanned by the eigenvectors computed on the training data. Fisherface [57] overcomes some of the eigenfaces' weaknesses by learning the projection operator in a supervised fashion to maximize (minimize) inter (intra) class variance, consequently allowing a higher degree of invariance to illumination changes. Other conventional approaches based on the low-dimensional representation of faces include sparse representations [58, 59], linear subspace [60, 51] and manifold [61] representations. A considerably different approach, [62, 63] follows the path of employing local features to overcome the challenges in handling the facial changes.

The most considerable improvement in biometric authentication has been achieved through deep learning-based methods. It removed the major bottlenecks encountered by conventional face authentication methods. Deep learning methods successfully solved problems which were a nightmare to traditional acquisition models, like far from ideal acquisition, different poses, illuminations, and expression conditions, see, e.g., DeepFace [64]. A well-known method in this line is FaceNet [65], which learns the input images' embeddings employing triplet loss. The network is trained so that the embeddings preserve the notion of the image similarity in terms of the ℓ_2 distance in the embedding space. However, it is common to train the network with a softmax cross-entropy loss because of the instability arising during a triplet-loss training. Nevertheless, the interclass dispersion and intraclass compactness are not guaranteed in this case. Recently ArcFace [66] introduced the additive angular margin loss to improve the discriminative power of the learned embedding while leading to a stable training process. Few other works adopt a similar strategy, e.g., [67, 68, 69, 70, 71].

The aforementioned works mainly rely on the recent trend in face recognition based on embedding computation and matching. Most of the effort is spent designing a loss function that can lead to more stable and effective embeddings.

In this respect, let us better emphasize the scope of AuthNet with respect to the latest trends in unconstrained face *recognition*. In AuthNet, the focus is mainly the biometric *authentication* problem; the target is to have minimum false acceptances while achieving high recognition accuracy. For this, it is common to assume that the user put himself in a controlled condition. The face datasets we consider are ones commonly employed for biometric authentication tasks; see [72, 73]. On the other hand, recent "in-the-wild" face datasets are better suited for evaluating recognition and clustering tasks because of the large number of users and poses. Since the number of samples per user is very limited, such datasets cannot cope well with user-specific training.

3.2.2 Fingerprints

Fingerprints are one of the earliest and most widely employed biometric traits in practical systems. Most conventional fingerprint authentication approaches are based on handcrafted features computed from minutiae, ridge, and valley patterns and rely on standard template matching in the domain. In general, they can be categorized based on the use of global or local features of fingerprints. Some of the works relying on global features include [74, 75]. Conversely, the approaches [76, 77, 78, 79, 80] are based on descriptors relying on local information of the minutiae and neighborhood. Moreover, it was shown in [81] that employing additional information such as shape context and orientation leads to an additional improvement in the performance. Recently, new approaches based on deep learning representational capabilities have been proposed, e.g., improving minutiae extraction and classification robustness. In [82, 83] CNNs are employed to extract minutiae from raw fingerprint images, in [84] stacked autoencoders are employed to classify fingerprint into the arch, left/right loop, and whorl. In [85], authors employ neural networks to filter minutiae and improve detection, whereas in [86] neural networks are used to extract minutiae on thinned fingerprint images. Recently CNN's are also used for latent fingerprint minutiae extraction [87].

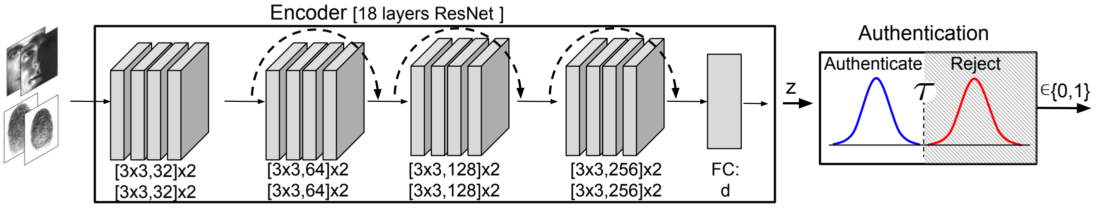


Figure 3.3: AuthNet-R architecture at authentication phase. In this phase the biometric trait of a user requesting access is given to the pre-trained encoder which will output a sample $\mathbf{z}$ coming from either $\mathbb{P}_0$ or $\mathbb{P}_1$. Then, the thresholding decision is made and a binary output (accept or reject) is returned.

3.3 Proposed method

We start by introducing and describing the components of the proposed architecture designed for biometric authentication see Fig. 3.1. AuthNet strives to learn a well-behaved representation of the input biometric traits in latent space, leading to simple tunable decision boundaries for classification. The aim is to learn a mapping from the sample in biometric space to a sample of the target probability distribution for authorized and unauthorized users. Ideally, the distance between the probability distribution of samples resulting from the mapping and the target distribution should be minimal. One widely employed approach to tackle this kind of problem is through an adversarial game.

3.3.1 Adversarial learning

Adversarial models are a prevalent approach for generative models. A foremost generative model trained through adversarial loss, the Generative Adversarial Network (GAN) [52], gained instant popularity and unlocked the path to adversarial training.

A GAN implicitly learns the probability distribution of input data such that the network can *generate* samples similar to input data. In other terms, the network learns to minimize the distance metric between the distribution of the generated samples and the real data. GAN employs JS divergence as the distance metric, which is the optimal solution for a two-player game. The idea behind adversarial learning models is to reach a minimum of a function defined as a minimax game, where two entities have adversarial goals. The equilibrium between the local optimal solutions corresponds to the global optima. The two networks, named generator and discriminator, are modeled as neural networks with the minimax game introduced in the loss function to make the two networks compete against each other during training. The discriminator aims to discriminate between the generated and the real samples correctly; contrarily, the generator should generate the samples realistic enough to fool the discriminator.

In AuthNet, as discussed in detail in the following section, the data distribution samples are mapped to a latent representation following target distributions. This can be seen as the inverse mapping of a conventional GAN, in which samples of fixed distributions are mapped to the captured distribution of the data.

3.3.2 Latent mapping

Let us now provide the details of AuthNet, whose main concept is depicted in Fig. 3.1.

Let $\mathcal{B} = \{\mathcal{B}_{a=0}, \mathcal{B}_{a=1}\}$ denote the set of all possible biometric traits and $a \in \{0,1\}$ an indicator variable such that $a = 1$ represents the authorized user and $a = 0$ represents all other unauthorized users. Furthermore, let us define as $\mathbf{x} \in \mathbb{R}^n$ a generic biometric trait in $\mathcal{B}$ and as $\mathbf{z} \in \mathbb{R}^d$ its latent representation with $d < n$. The goal is to learn an encoding function $\mathbf{z} = H(\mathbf{x})$ of the input biometric trait such that $\mathbf{z} \sim \mathbb{P}_1$ if $\mathbf{x} \in \mathcal{B}_{a=1}$ and $\mathbf{z} \sim \mathbb{P}_0$ if $\mathbf{x} \in \mathcal{B}_{a=0}$, with $\mathbb{P}_1$ and $\mathbb{P}_0$ the target distributions in the latent space. If the distributions $\mathbb{P}_1$ and $\mathbb{P}_0$ are well-behaved, a simple distance-based thresholding approach can be employed to determine whether the user with its associated biometric trait $\mathbf{x}$ is authorized or not.

Let us set $\mathbb{P}_1 = \mathcal{N}(\boldsymbol{\mu}_1, \sigma_1\mathbf{I})$ and $\mathbb{P}_0 = \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0\mathbf{I})$ to be Gaussian, this amounts to enclosing the energy of the latent representation of authorized and unauthorized users within hyperspheres whose radius depends on both d and the distribution parameters.

For the sake of simplicity and without loss of generality, we set $\boldsymbol{\mu}_1 = \mu_1\mathbb{1}$ and $\boldsymbol{\mu}_0 = \mu_0\mathbb{1}$, with $\mu_1 < \mu_0$ and $\sigma_1 = \sigma_0$. Having Gaussian distributions with equal variance, a hyperplane is the optimal decision boundary, further boiling down to a simple threshold when $\mathbf{z}$ is a scalar. This results in a very simple classifier that learns a complex mapping from a high-dimensional input space to a much simpler latent space in a way that mimics kernel-based methods.

Modes of operation: AuthNet operates in two distinct phases: enrollment and authentication. During enrollment (see Fig. 3.2), the users are registered in the system. The latent representation of authorized and unauthorized users are enforced to follow $\mathbb{P}_1$ and $\mathbb{P}_0$, based on the one-hot label vector. The authentication phase follows the enrollment phase (see Fig. 3.3); the biometric traits are projected to the latent space tested against the target distributions to determine whether the trait belongs to the authorized or unauthorized user class. For the case $d = 1$, this amounts to comparing a scalar z with a threshold: if the metric value is less than a specific threshold, i.e., $\mathbf{z} \sim \mathbb{P}_1$, the user is classified as authorized user, else the user is classified unauthorized.

3.3.3 Enrollment

The enrollment phase aims to learn an encoding function $H(\mathbf{x})$, which maps the user biometric traits onto the target distributions. An optimal $H(\mathbf{x})$ is one for which the distance between $H(\mathbf{x}) : \mathbf{x} \in \mathcal{B}_{a=1}$ and $\mathbb{P}_1$, and between $H(\mathbf{x}) : \mathbf{x} \in \mathcal{B}_{a=0}$ and $\mathbb{P}_0$ is minimum. To address the problem, we employ an adversarial loss whose optimum is reached once the JS divergence between the latent mapping and target distribution is minimized.

Fig. 3.2 depicts the AuthNet architecture during enrollment phase. It consists of two subnetworks: an encoding function $H(\mathbf{x}, \boldsymbol{\theta}_h)$ having parameters $\boldsymbol{\theta}_h$ and a discriminator $D(\mathbf{p}, \boldsymbol{\theta}_d)$ subnetwork with parameters $\boldsymbol{\theta}_d$. For the sake of readability we drop the parameters in the notation of encoding and discriminator subnetworks.

Biometric traits $\mathbf{x}$ are given as an input to the encoding function $H(\cdot)$, which outputs the encoded latent representation $\mathbf{z}$. The vector $\mathbf{p} \in \{\mathbf{s}, \mathbf{z}\}$ is provided as an input to the discriminator in an alternate fashion, either a sample from one of the target distributions $\mathbf{s}$ or the encoded latent representation $\mathbf{z}$. The vector $\mathbf{s} \in \mathbb{R}^d$ is made of randomly drawn samples from the target distributions $\mathbb{P}_1$ if $\mathbf{x} \in \mathcal{B}_{a=1}$ or $\mathbb{P}_0$ if $\mathbf{x} \in \mathcal{B}_{a=0}$, respectively. To further improve the training stability and performance, the input biometric trait label a is given to the discriminator as additional information which serves as a switch to select a "sub-discriminator" function for either authorized or unauthorized users.

The discriminator $D(\mathbf{p})$ outputs a scalar value, which can be interpreted as the probability of the given input coming from the target distribution. $D(\mathbf{p})$ returns 1 when the input is classified as target distribution, so this is the probability that the input is from the target distribution

To address the above-defined adversarial setting, the loss function we consider is given by

$$V(H, D) = \mathbb{E}_{\mathbf{s}\sim\mathbb{P}}\left[\log(D(\mathbf{s}, a))\right] + \mathbb{E}_{\mathbf{x}\sim\mathcal{B}}\left[\log(1 - D(H(\mathbf{x}), a))\right], \tag{3.1}$$

which is optimized as a mini-max two-player game according to

$$\min_{\boldsymbol{\theta}_h} \max_{\boldsymbol{\theta}_d} V(H, D), \tag{3.2}$$

where the optimization is carried over the parameters $\boldsymbol{\theta}_h$ and $\boldsymbol{\theta}_d$ in an alternate fashion.

Being an adversarial model, the specific goal of the encoding function $H(\mathbf{x})$ is to generate samples that minimize the probability of D making a correct choice, i.e., generate samples $\mathbf{z}$ which will fool the discriminator. Contrarily discriminator $D(\mathbf{p})$ task is to maximize the probability of assigning correct labels to the samples of latent representation $\mathbf{z}$ and the target distribution $\mathbf{s}$.

At the start of the learning phase, the discriminator quickly learns to distinguish latent representation $\mathbf{z}$ from the target distributions $\mathbf{s}$. Over the next few iterations, the encoder learns to generate samples closer to the target distributions. Eventually, the encoder will start to generate samples $\mathbf{z}$ which are close enough to $\mathbf{s}$ so that the discriminator cannot distinguish between them.

In the case of AuthNet, as generally done for adversarial models, these two objectives are optimized alternately: one step for the discriminator followed by one for the encoder.

3.3.4 Authentication

Once the training phase is finished, the subsequent phase is authentication. In the authentication phase, a user's biometric trait is provided as an input to the network based on which a user is either authorized or denied access.

For AuthNet, during the authentication phase, only the trained encoder network is utilized. This network computes the latent representation $\mathbf{z}$ of the input biometric trait on which the decision is made. As stated, for Gaussian distributions, a hyperplane can be used for the optimal decision, i.e., we can use the test.

$$(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T \mathbf{z} \lessgtr (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T (\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)/2. \tag{3.3}$$

For $d = 1$, this boils down to comparing the scalar $\mathbf{z}$ with a threshold $\tau = (\mu_1 + \mu_0)/2$, (see Fig. 3.3).

3.4 Training and Implementation Details

3.4.1 Network insight

Encoder sub-network

A biometric trait, either an RGB or a gray-scale image with the size depending on the dataset, is given as input to the encoder subnetwork. The choice of encoder subnetwork is a crucial task; in general, any state-of-the-art neural network that can learn good features can be employed. To prove the idea, experiments on several neural network architectures such as plain CNN, ResNet [55], and DenseNet [56] with a different number of layers were conducted. It was empirically found that for the considered datasets, ResNet-18 or DenseNet-50, followed by a fully-connected layer with an output of size d, are able to efficiently learn the latent mapping. It is crucial to notice that in the last layer of the encoder sub-network, no non-linear activation function is employed, as the output is mapped to a sample of the target distributions. Additionally, it was found that employing a large network with too many parameters like ResNet-101/152 or DenseNet-121/169 for small/ medium-sized datasets leads to slower training without any performance improvement. This motivates us to use ResNet-18 / DenseNet-50 encoder sub-networks.

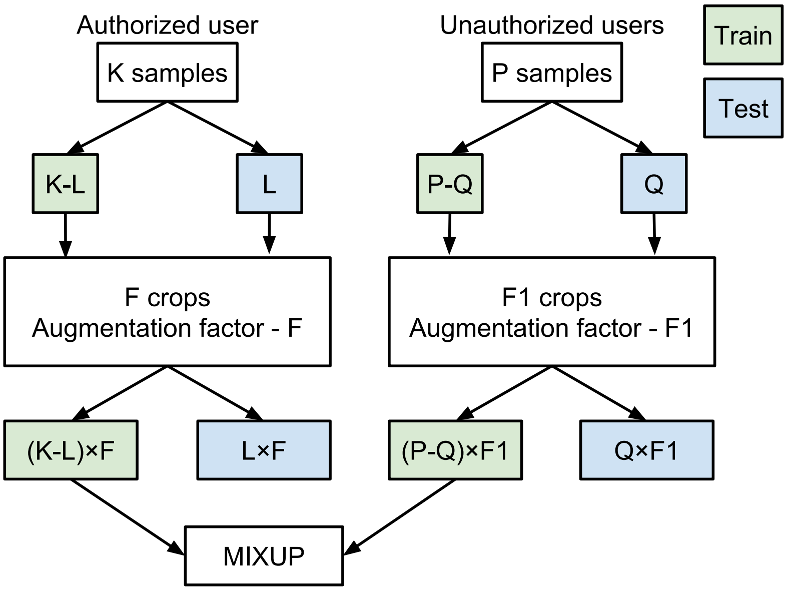


Figure 3.4: The data augmentation makes use of random crops to increase the number of input samples to a factor of F and F_1 for authorized and unauthorized users respectively. Then, positive and negative samples are mixed by means of a convex combination.

In the following sections, AuthNet with ResNet encoder sub-network is referred to as AuthNet-R and AuthNet employing DenseNet encoder as AuthNet-D.

Discriminator sub-network

The discriminator subnetwork has three main inputs:

1. Prior distribution samples
2. Output from encoder sub-network i.e. latent vector $\mathbf{z}$ having size d
3. One-hot vector a during training to tell the discriminator whether the sample is authorized or unauthorized

The discriminator consists of a fully connected network containing 8 layers having a ReLU activation function at each layer's output. The number was empirically selected so that the discriminator has enough capacity to compete with the encoder. Through empirical testing, it has been found that the chosen network size works well for different d-values and makes the encoder strong enough to compete with the chosen encoder (i.e., ResNet-18 or DenseNet-50) and, as a result, leads to stable training. The layer size depends upon the encoder sub-network structure: If the layers are not properly sized, the encoder loss might instantly drop to zero; as a

d	**GAR@10^{-3}FAR%**	**GAR@10^{-3}FAR%**
	Yale face B	**CMU Multi-PIE**
1	100	100
3	97.663	99.88
64	91.001	96.53

Table 3.1: GAR comparison of randomly selected users from Yale DB2 and CMU MultiPIE when considering different dimensionality of the latent space d. The best case is obtained for $d = 1$: highest GAR for a fixed FAR=10^{-3}.

result, stopping the training. We found that eight layers in the discriminator sub-network were enough to cope with the “capacity” (or the number of parameters) of the encoder sub-network.

The input to the discriminator sub-network is the concatenation of latent vector $\mathbf{z}$ from the encoder subnetwork and the one-hot vector a corresponding to the class the user belongs to. The first fully connected layer has an output size equal to 100 which gradually increases to a maximum of 1000; after this, the layer size gradually decreases with the final layer having an output size equal to 1 to which the sigmoid activation is applied, estimating the probability that the sample is coming from the target prior distribution.

Preprocessing and training parameters

The network is trained using Adam optimizer [88] with an iterative algorithm, as discussed in [52]. The optimization is carried stepwise, one for the encoder and one for the discriminator. Weight decay of 0.0004 and dropout of 0.7 is employed. Initially, for the first 5000 iterations, the learning rate is 0.01 and is then decreased by a factor of 10 after every 5000 iteration. In total, the network is trained for 30000 iterations. In AuthNet, the only preprocessing considered is the energy normalization of the input images

3.4.2 Data augmentation

Having a large amount of diverse data is crucial for training deep neural networks. A neural network’s performance depends upon the features learned from the data used during the training phase. In the case of AuthNet for biometric authentication, the acquisition process should be fast, and the number of acquired samples during training should be reasonable. As a result, if we consider fingerprints, we can have as few as 12 samples per user to be authenticated. Hence, an efficient augmentation strategy is required to provide the network with enough diverse data.

Besides, the aim is to have a general-purpose and versatile augmentation strategy that could work for different biometric traits.

Dataset	Method	EER%	GAR@10^{-2}FAR%	GAR@10^{-3}FAR%	Max accuracy
Face-Yale	**AuthNet-D**	**0* ± 0 (0.009)**	**100* ± 0 (100*)**	**100* ± 0 (100*)**	**100* ± 0 (99.991)**
	AuthNet-R	**0.003 ± 0.011 (0.019)**	**100* ± 0 (100*)**	**99.687 ± 1.767 (100*)**	**99.997 ± 0.011 (99.983)**
	AuthNet enc. classifier	0.013 ± 0.054 (0.040)	100* ± 0 (100*)	99.011 ± 0.031 (100*)	99.987± 0.054 (99.961)
	FaceNet	1.258 ± 2.084 (1.288)	98.793 ± 3.114 (98.707)	98.781 ± 3.150 (98.683)	98.825 ± 1.869 (99.300)
	ArcFace	0.696 ± 1.533 (0.893)	99.024 ± 2.798 (99.108)	97.762 ± 4.063 (96.630)	99.367 ± 1.434 (99.229)
Face Multi-Pie	**AuthNet-D**	**0* ± 0 (0.005)**	**100* ± 0 (100*)**	**100* ± 0 (100*)**	**100*± 0 (99.993)**
	AuthNet-R	**0* ± 0 (0.001)**	**100* ± 0 (100*)**	**100* ± 0 (100*)**	**100*± 0 (99.998)**
	AuthNet enc. classifier	0.009 ± 0.034 (0.676)	100* ± 0 (99.432)	99.886 ± 0.447 (99.693)	99.991 ± 0.034 (99.325)
	FaceNet	0.770 ± 1.080 (0.930)	98.466 ± 3.490 (99.201)	90.513 ± 21.582 (92.045)	99.368± 0.981 (99.197)
	ArcFace	1.727 ± 0.164 (1.811)	98.124 ± 0.321 (98.125)	97.897 ± 0.814 (97.272)	99.058 ± 0.103 (98.871)

Table 3.2: Performance comparison of AuthNet with respect to other face authentication schemes. Average values of the considered metrics computed independently on each user and on the aggregated scores (shown in parenthesis) are reported. We mark as 0* and 100* values below the minimum achievable precision, i.e. smaller than $4.1E{-}5$ and $5.68E{-}4$ for Yale and MultiPIE datasets respectively.

Dataset	Method	EER%	GAR@10^{-2}FAR%	GAR@10^{-3}FAR%	Max accuracy
Fingerprint	**AuthNet-D**	**0* ± 0 (0.147)**	**100* ± 0 (100*)**	**100* ± 0 (98.817)**	**100*± 0 (99.895)**
	AuthNet-R	**0.058 ± 0.217 (0.339)**	**99.955 ± 0.248 (100*)**	**99.400 ± 2.949 (98.476)**	**99.957± 0.173 (99.740)**
	AuthNet enc. classifier	0.188 ± 0.722 (0.565)	98.448 ± 8.621 (99.845)	95.148 ± 9.742 (94.384)	99.812 ± 0.722 (99.435)
	Verifinger	0.163 ± 0.697 (0.758)	99.680 ± 1.229 (99.796)	99.375 ± 2.459 (95.638)	99.902 ± 0.373 (99.398)
	Hybrid approach [81]	1.515 ± 2.651 (3.200)	95.937 ± 7.452 (95.000)	90.001 ± 19.261 (85.909)	98.868 ± 2.178 (96.906)

Table 3.3: Performance comparison of AuthNet with respect to other fingerprint authentication schemes on FVC 2006 DB2 dataset. Average values of the considered metrics computed independently on each user and on the aggregated scores (shown in parenthesis) are reported. We mark as 0* and 100* values below the minimum achievable precision, i.e. smaller than $5.5E{-}5$.

Fig. 3.4 summarizes the augmentation strategies we employ. It is based on image crops and samples mixup. Firstly, for each sample of size $m \times m$, all possible crops of size $n \times n$ are extracted. As the potential positive samples (authorized users) are considerably less than available negative samples (unauthorized users), different augmentation factors are used for each, namely F and F_1. Former refers to the augmentation factor for authorized user samples, i.e., the positive ones; the latter refers to the negative ones. Clearly for us, $F > F_1$ due to the unbalanced class sizes.

Once positive and negative training samples crops are obtained, they are mixed using a convex combination as described in [89] to obtain diverse training samples. Another side advantage of the mixup, as shown in [89] is better regularization and improved network generalization.

Given a positive and a negative sample, respectively denoted as $\mathbf{x}_{a=1}$ and $\mathbf{x}_{a=0}$, a new sample is fabricated as $\mathbf{x}'_m = \lambda \mathbf{x}_{a=1} + (1-\lambda)\mathbf{x}_{a=0}$,where $\lambda \in [0,1]$ is distributed according to a Beta distribution with parameters α and β that in our case are both fixed to 0.4. This choice of parameters leads to a distribution peaked at 0 and 1 and has the lowest probability at $\lambda = 0.5$. It is consequently producing augmented samples that are not far from the centroid of either class. To associate the labels of newly created samples, we use round(λ).

3.5 Performance analysis

AuthNet is a general-purpose network designed to work on different biometric traits seamlessly. Experiments were conducted on **faces** and **fingerprints**. It is common to assume in a biometric authentication system that the user put him/herself in controlled conditions to acquire biometric traits. In this regard, the considered datasets are among the biggest ones acquired in such conditions.

3.5.1 Datasets

For **face** authentication, we evaluate AuthNet on CMU Multi-PIE [90] and Yale Face database DB2 [91].

CMU Multi-PIE has 337 candidates with 750,000 images. The dataset is acquired over a span of five months in four different sessions, with the images having 15 viewpoints and 19 illumination conditions. It includes images with different illuminations, expressions, and poses. We consider the frontal posed images with different expressions and illuminations to highlight the robustness of the algorithm. For each user enrollment, 75% of the samples are employed for the training, and the remaining 25% are left for testing. Out of 128, 96 user samples are drawn for unauthorized users' training, and the remaining 32 user samples are left for testing. Samples are resized to $144 \times 192 \times 3$, maintaining the aspect ratio. To create more diverse samples, positive and negative users samples are combined through a mixup strategy, as discussed in Sec. 3.4.2.

The second dataset we employ is the cropped version of *extended Yale Face Database B* having the frontal pose images of 38 subjects with varying illumination conditions. For each authorized user enrollment, 75% of the samples are drawn for training, and the remaining 25% are left for testing. For unauthorized users, 31 user samples are employed for training, and 6 user samples are left for testing. Further, for augmentation we employ crops of size 184×160, as described in Sec. 3.4.2 with an augmentation factor of $F = 81$ and $F_1 = 25$ respectively. Finally, for each training batch of size b, we randomly select b samples from both authorized and unauthorized users datasets. Then, positive and negative samples are combined through mixup as explained in Sec. 3.4.2 resulting in b new samples.

For **fingerprint** authentication we employ *Fingerprint Verification Competition (FVC 2006) DB2* [92] dataset. Although old, this is still an actively used dataset [93, 94]. It constitutes of 150 users samples acquired through an optical sensor. The samples are resized to 202×149, maintaining the aspect ratio. For each authorized user enrollment, 75% of the samples are used for training, and the remaining 25% are left for testing. For unauthorized users, 124 user samples are used for training, and 25 user samples are left for testing. Finally, the dataset is augmented by $F = 289$ and $F_1 = 25$ using the crops of sizes 186×133 pixels and mixup augmentation as done for the faces dataset is employed.

3.5.2 Evaluation Metrics

The primary metric we employ in the experimentation is the EER, defined as the value at which the FAR is equal to the Fase Rejection Rate (FRR). For a given threshold τ, FAR indicates the samples that should have been rejected over the total number of samples, whereas FRR indicates the number of rejected samples that should have been accepted over the total number of samples.

It is crucial to notice that the FAR is a critical parameter in biometric authentication systems: a significant value indicates a high number of unauthorized users wrongly authorized by the system. Indeed the situation is more dangerous compared to high FRR. For a good biometric authentication system, it is desired to have a low FAR; for this, we test the systems at low FAR values: we report the Genuine Acceptance Rate (GAR), namely the relative number of correctly accepted users at FAR equal to 10^{-2} and 10^{-3}. Finally, we also report the maximum accuracy, which defines the number of correctly classified users.

In the results section, we first compute the metrics independently for each considered users and report the resulting average value with the standard deviation. This gives an insight into how the system performs on average on a per-user basis. Further to better understand the system's overall performance, we additionally report the aggregated results on all the user's scores and illustrate the Receiver Operating Characteristic (ROC) curve computed on the aggregated scores of the considered users.

3.5.3 Dimensionality of latent space

An important parameter in the design of AuthNet is the choice of the latent space dimensionality d. The datasets we are considering are medium sized, thus it is not surprising that a smaller d achieves better results. In case of large datasets that are not being limited by the data scarcity, a larger latent space improves the data separation and leads to improved performance.

In our tests, we fixed the hyperparameter $d = 1$, since in our experiments this choice gave us better results as can be seen in Tab. 3.1. Intuitively, as the latent space grows in dimensionality, a larger number of training samples are required to avoid overfitting.

MultiPIE has a relatively larger size compared to Yale dataset, it can be observed from Tab. 3.1 that for MultiPIE higher GAR is achieved at larger values of d compared to Yale dataset.

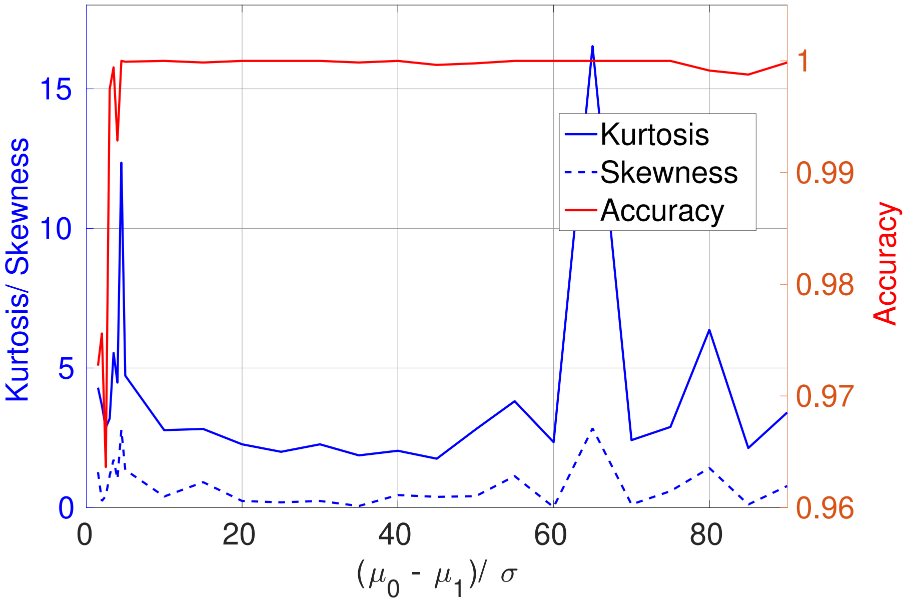


Figure 3.5: Accuracy, kurtosis and skewness comparison of a randomly selected user from CMU-MultiPIE having $\mathbb{P}_0 = \mathcal{N}(k,1)$ where $k = [0.5,90]$, and $\mathbb{P}_1 = \mathcal{N}(0,1)$. If the means of the two distribution are too far apart the training process gets unstable, hence it affects the accuracy, kurtosis and skewness of the imposed distributions.

3.5.4 Parameters of authorized and unauthorized users distributions

The target distributions for authorized and unauthorized users are set to be Gaussian for AuthNet. The choices come from the fact that from the central limit theorem, the output of a large enough fully connected layer naturally tends to be Gaussian distributed [95, 96]. We set the distribution to be $\mathbb{P}_1 = \mathcal{N}(0,1)$ and $\mathbb{P}_0 = \mathcal{N}(40,1)$. $\mu_1 = 0$ and $\mu_0 = 40$ are chosen to be different enough to keep the distributions far apart from each other.

Fig. 3.5 depicts the maximal accuracy along with the kurtosis and skewness of the latent space representation as a function of μ_0 for a randomly selected CMU-MultiPIE user for AuthNet. It can be observed the maximal accuracy region corresponds to $15 \leq \mu_0 \leq 45$; further, in this region, the skewness and kurtosis are close to zero and three, respectively, showing that the training converges to Gaussian distributions. Further, if the difference between μ_0 and μ_1 is too large (e.g., $\mu_0 > 50$), the training process becomes unstable, and the distributions become far from Gaussian.

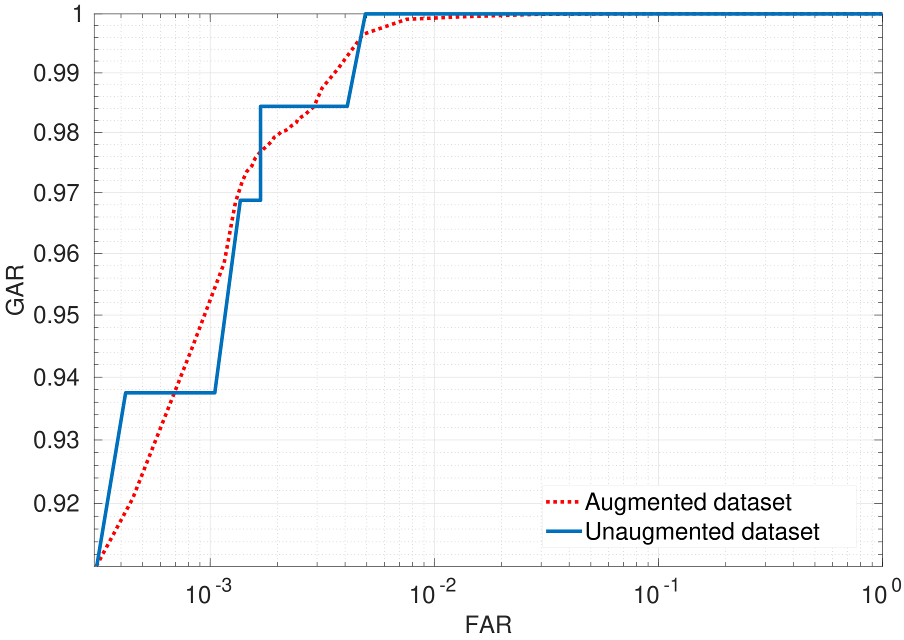


Figure 3.6: ROC comparison on overall results of the different users from FVC2006 DB2 for augmented and unaugmented datasets. The augmented dataset shows the same performance without quantization of probability values.

3.5.5 Results

Before moving to results, it is essential to consider that the performance metrics' precision is proportional to the number of test samples. The maximum obtainable precision for the considered metrics (explained in Sec. 3.5.2) is given by $1/c$ with $c = \min\{L \times F, Q \times F_1\}$. Therefore, we will verify that the proposed augmentation strategy does not introduce any bias on the measured performance (in the ROC curve).

From Fig. 3.6, it can be observed that the augmentation avoids the coarse quantization of probability values without introducing any bias. For this reason, we will compute the considered metrics on the augmented datasets.

In addition to biometric-related methods, we also report the comparisons the comparison of the Encoder network of AuthNet-R used as a classifier and trained with the sigmoid cross-entropy loss. We have discussed classifiers' issues having highly non-linear and complex to analyze boundaries in Sec. 5.1. Thus, we evaluate deep learning classifiers' behavior on the same architecture as the AuthNet-R encoder, which is not trained in an adversarial way to assess and evaluate the adversarial scheme's benefits employed in AuthNet.

Tab. 3.2 presents the results achieved by AuthNet and benchmarking methods in terms of EER, GAR values at FAR=$\{10^{-2}, 10^{-3}\}$ and maximum accuracies on the individual and aggregated scores. Fig. 3.7-3.9 depict the histogram of the aggregated scores obtained by different methods. The ROC comparison for different benchmarking methods is depicted in Fig. 3.10. Lastly, for the sake of readability, unless differently specified from now on we will refer to both AuthNet-R and AuthNet-D as “AuthNet”.

Face authentication

We employ CMU Multi-PIE and Yale Face database B, as detailed in Sec. 3.5.1 for face authentication. We compare with ArcFace [66] and FaceNet [65] for benchmarking with state-of-the-art deep learning techniques. ArcFace and FaceNet tend to work better on aligned face patches. For CMU-Multi-PIE, the dataset is preprocessed by aligning and cropping the input faces using Multi-task Cascaded Convolutional networks (MTCNN) [98] the well-known approach for joint face detection and alignment. Yale Face database comprises frontal face images of the subjects, so face alignment and crop are not needed.

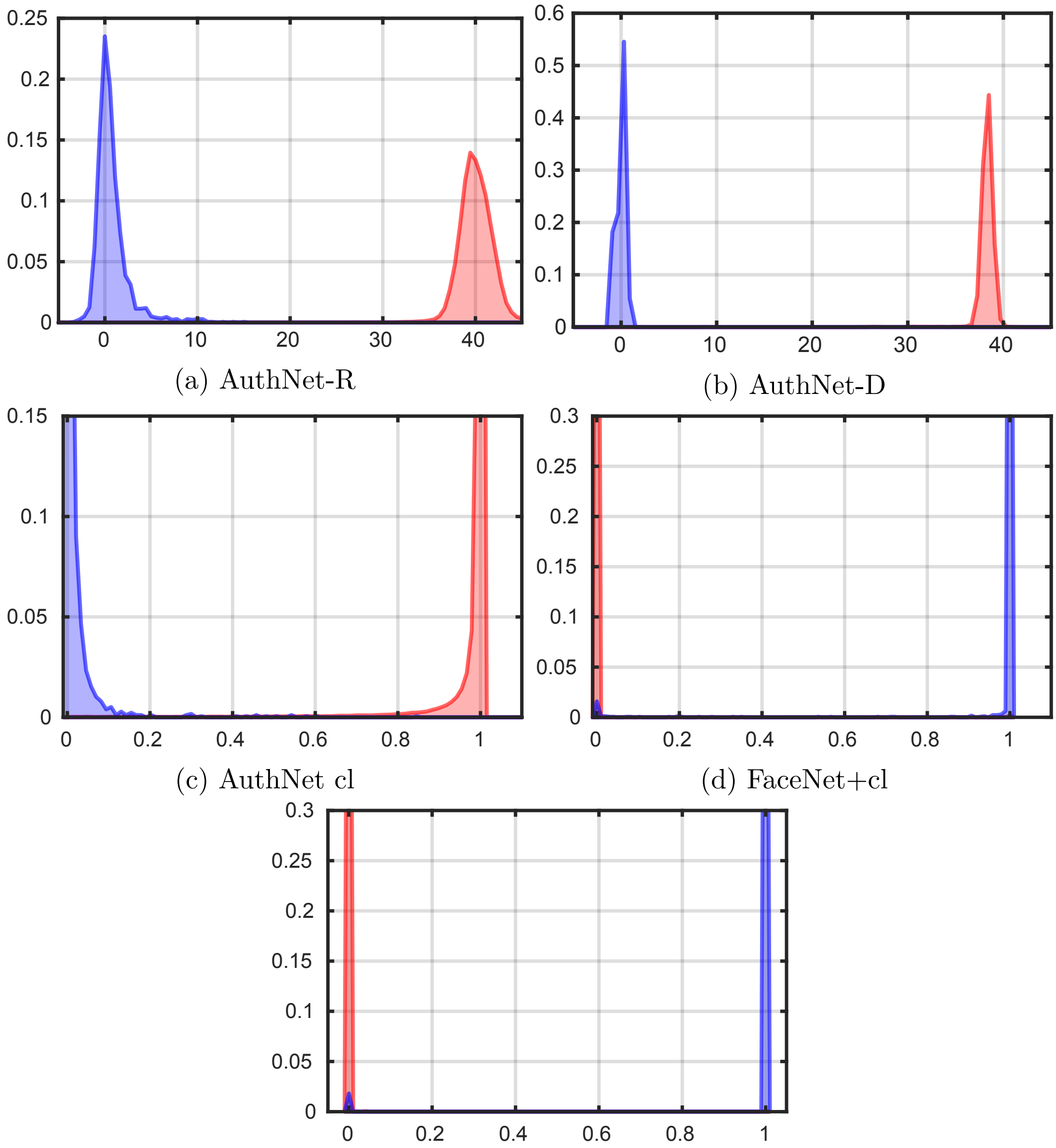
Regarding the training process of ArcFace and Facenet, as described in their respective papers, we employ the standard architecture using 512-dimensional embeddings. Given the above methods are meant to learn a generic face embedding for either face recognition, verification, or clustering:

1. a large training dataset is required
2. user-specific embedding cannot be learned

This will lead to an unfair comparison with AuthNet. To alleviate this issue and make the comparison fair, a two-step approach is followed. At first, FaceNet and ArcFace are trained on the large CASIA WebFace dataset [99] to obtain 512 dimensional embeddings from given input face images. Afterward, two-class FC classifiers (one for each user) are trained to classify the embeddings as either authorized or unauthorized.

Tab. 3.2 illustrates a comparison of EER, GAR at FAR=$\{10^{-2}, 10^{-3}\}$, and maximum accuracy for CMU Multi-PIE and Yale Face Database B, calculated on the users individual and the aggregated scores. From the results, it can be observed that, in terms of EER, AuthNet attains the lowest value outperforming other methods. Further, a minimal advantage of AuthNet-R to AuthNet-D can also be observed. Nevertheless, as shown in later experiments, the performance of these two AuthNet flavors’ is comparable, and a clear winner cannot be identified.

It is also interesting to observe that for AuthNet, high GAR values are obtained even for minimal FAR values. Multi-PIE’s high performance compared to Yale Face database B is understandable since the former has a significantly larger number of high-quality samples per user compared to other datasets. Additionally, AuthNet outperforms competing methods in terms of maximum accuracy achieved.



(a) AuthNet-R

(b) AuthNet-D

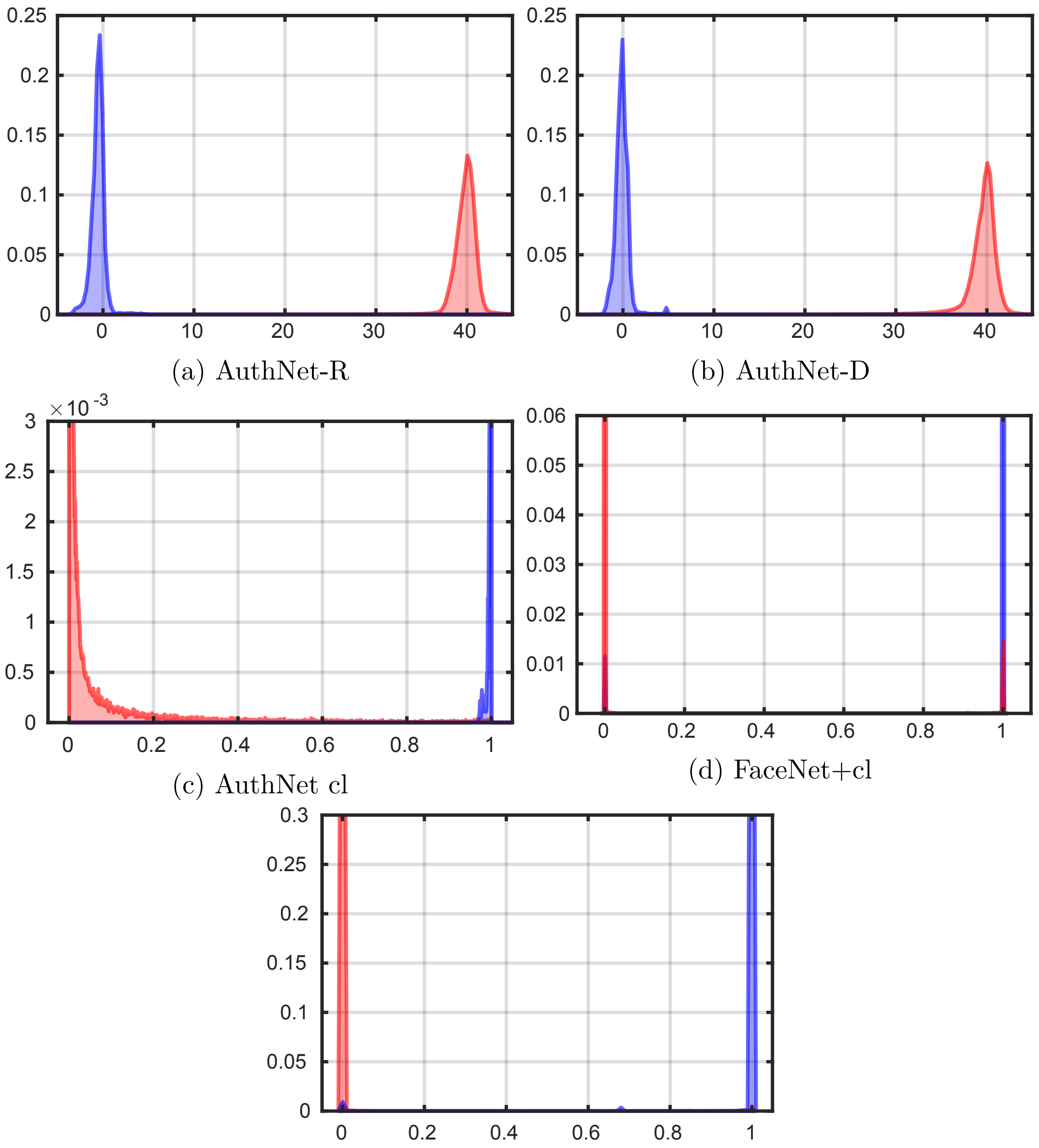
(c) AuthNet cl

(d) FaceNet+cl

(e) ArcFace + cl

Figure 3.7: MultiPie authentication scores for authorized users (blue) and unauthorized users (red). (a), (b) Histogram of **z** decision statistics of AuthNet; (c) Histogram of the sigmoid outputs of AuthNet encoder classifier; (d) Histogram of the sigmoid outputs of FaceNet embeddings classifier; (e) Histogram of the sigmoid outputs of ArcFace embeddings classifier. The plots in (c)-(e) depict a detailed view to better appreciate the leakage effects.

For the AuthNet encoder classifier, the performance in terms of EER is an order of magnitude less than that of AuthNet. In further detail, we can exclude that this



(a) AuthNet-R

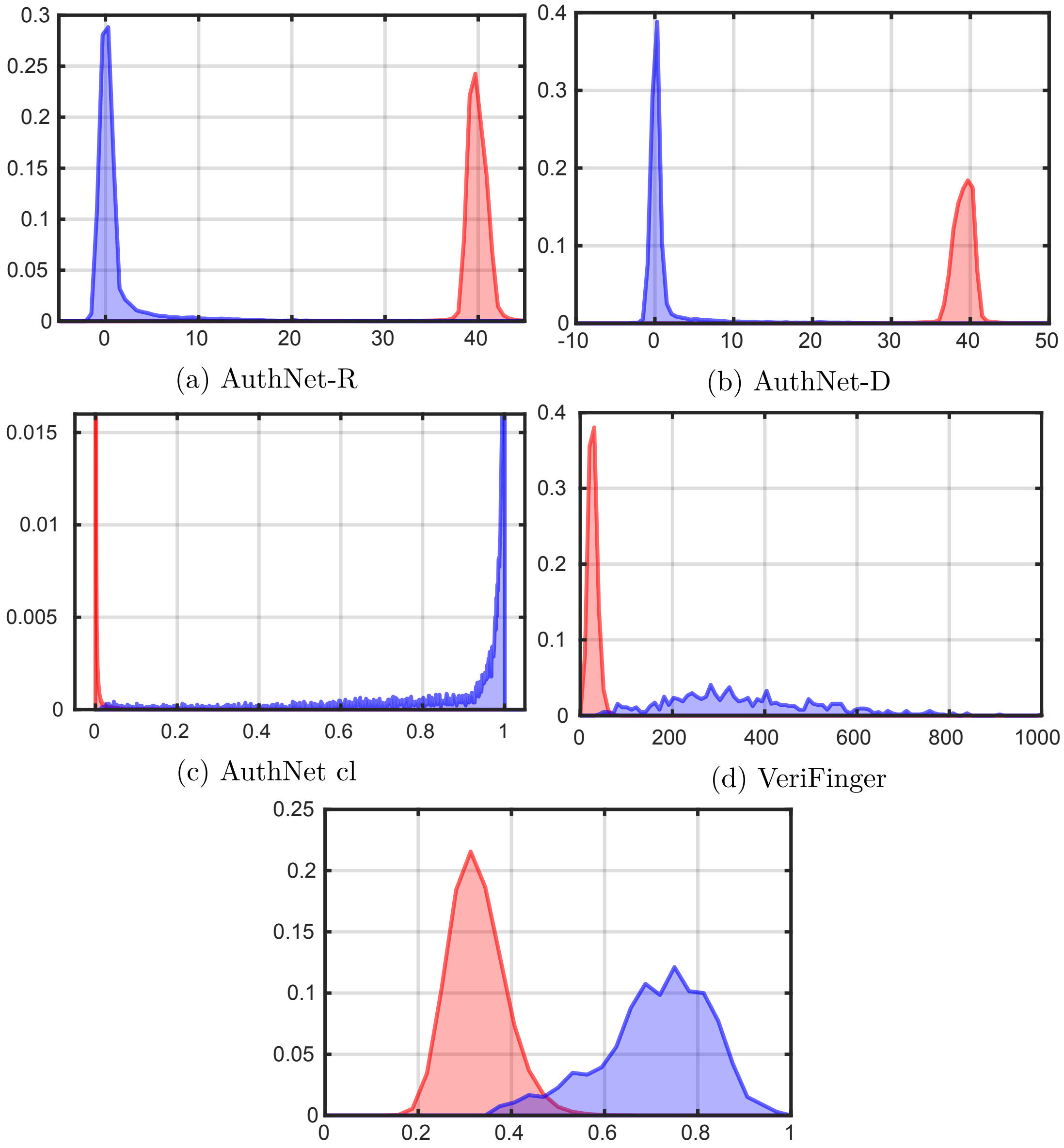
(b) AuthNet-D

(c) AuthNet cl

(d) FaceNet+cl

(e) ArcFace + cl

Figure 3.8: Face Yale authentication scores for authorized users (blue) and unauthorized users (red). (a), (b) Histogram of **z** decision statistics of AuthNet; (c) Histogram of the sigmoid outputs of AuthNet encoder classifier; (d) Histogram of the sigmoid outputs of FaceNet embeddings classifier; (e) Histogram of the sigmoid outputs of ArcFace embeddings classifier. The plots in (c)-(e) depict a detailed view to better appreciate the leakage effects.



(a) AuthNet-R

(b) AuthNet-D

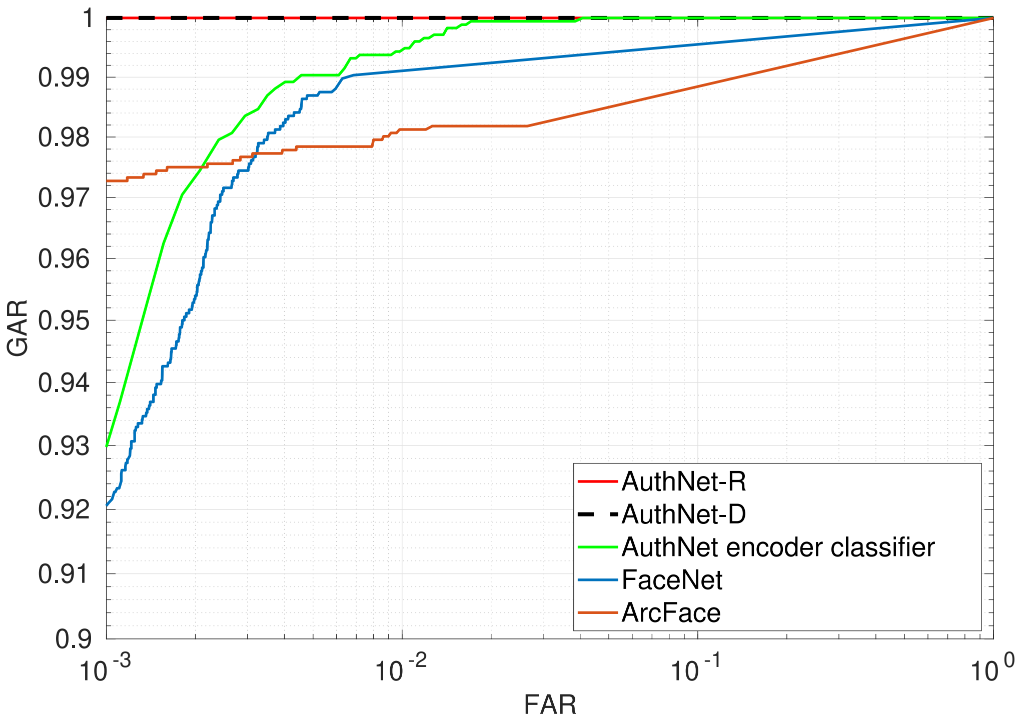
(c) AuthNet cl

(d) VeriFinger

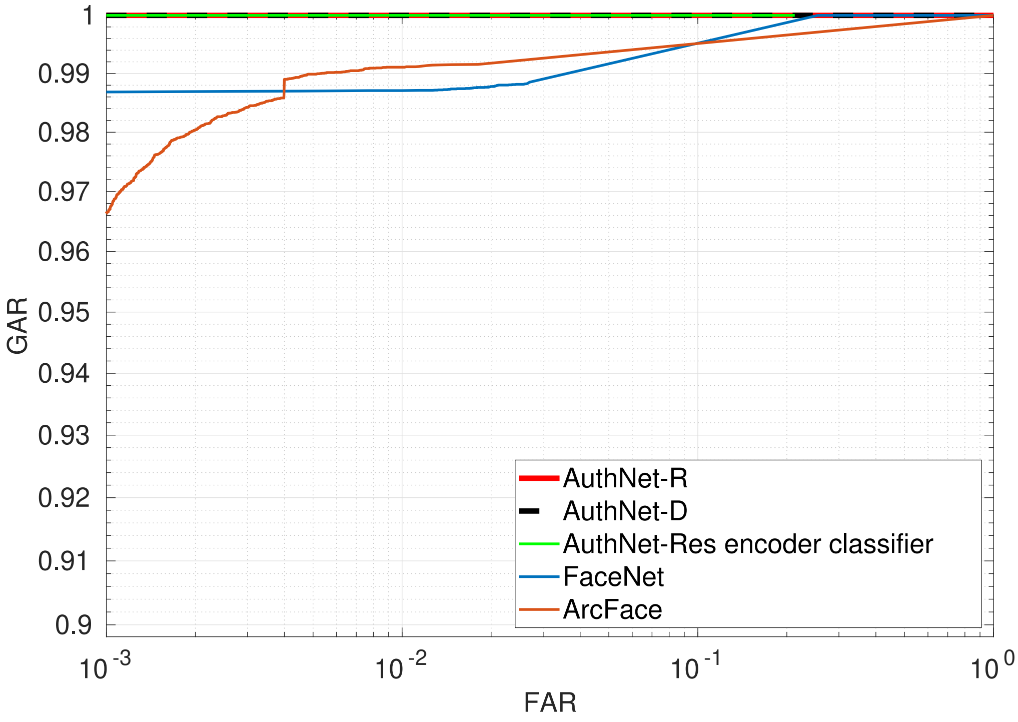
(e) Hybrid app

Figure 3.9: Fingerprint authentication scores for authorized users (blue) and unauthorized users (red). (a), (b) Histogram of **z** decision statistics for AuthNet; (c) Histogram of the sigmoid outputs of the AuthNet encoder classifier; (d) histogram of the matching scores of Verifinger; (e) histogram of the matching scores of the hybrid approach. The plot in (c) depicts a detailed view to better appreciate the leakage effects.

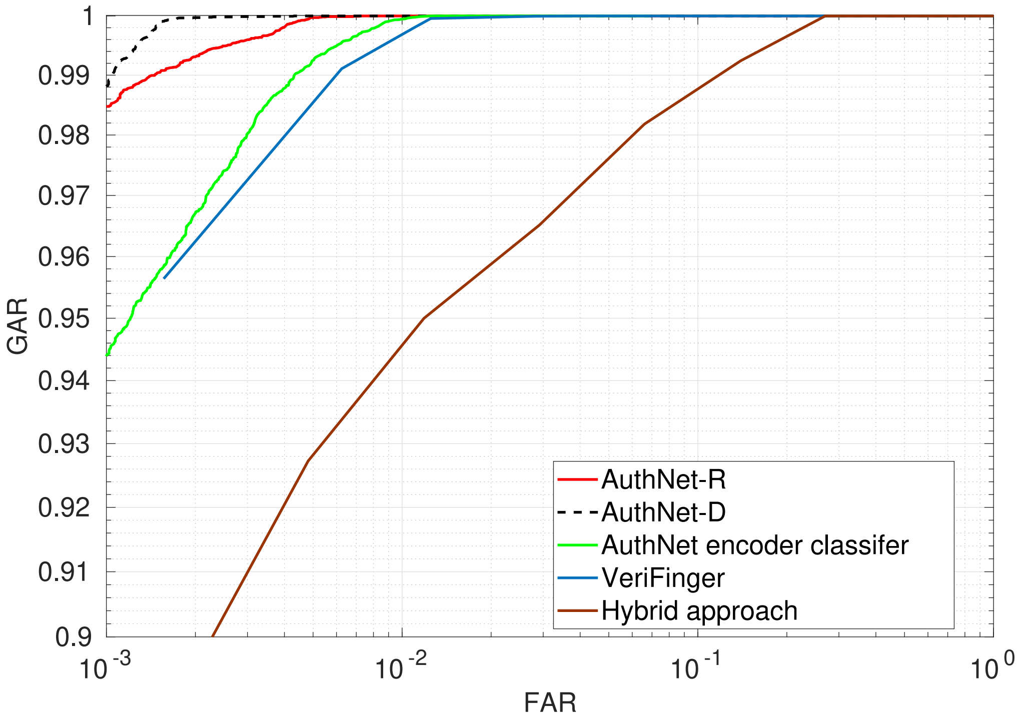
is due to the AuthNet encoder classifier overfitting on the negative samples. This



(a) CMU MULTI-PIE



(b) Face Yale dataset



(c) Fingerprint dataset

Figure 3.10: ROC comparison on aggregated results of users for faces (a) CMU Multi-PIE, (b) Face Yale database B, - fingerprint (c) FVC 2006 DB2 datasets. (a-b); AuthNet is compared with the AuthNet encoder classifier, FaceNet [65] and ArcFace [66] in (c); with AuthNet encoder classifier, VeriFinger [97] and the hybrid approach [81] in (b). In all the cases, AuthNet (red) and (black) achieves higher GAR with respect to other authentication schemes at different values of FAR.

case is seen by looking at Fig. 3.13 where it is depicted the ROC for the considered approaches when tested on out-of-domain or never-seen negative examples. It can be seen that the performance drop of the AuthNet encoder classifier is mostly bounded, and thus the lower performance is due to the lack of regularization of the decision space. Clearly, the results of this comparison imply that by regularizing the latent space through well-behaved distributions, it is possible to increase the system's accuracy by decreasing the number of false positives. This accentuates the superiority of the proposed latent space regularization over a traditional classifier. Further, the achieved EER by FaceNet and ArcFace is also an order of magnitude less than that of AuthNet. Additionally, for small FAR values, the genuine acceptance for these methods significantly reduces, which is not the case with AuthNet. This indicates a high variability of the results on a per-user basis, which can be observed from both individual and aggregated user scores in Tab. 3.2.

To better appreciate the effect of latent space regularization for AuthNet, the

method	$\sigma_{a=1}$	$\sigma_{a=0}$	$\beta_{2,a=1}$	$\beta_{2,a=0}$
AuthNet	1.108	1.451	3.369	3.282
AuthNet enc CL	7.176	7.739	3.985	3.809
FaceNet	7.446	8.517	3.843	3.918
ArcFace	6.919	10.183	3.514	4.215

Table 3.4: Standard deviation σ and Kurtosis β_2 of normalized test logit scores for authorized and unauthorized users.

face authentication scores for authorized and unauthorized users are depicted in Fig. 3.7 and 3.8 for different benchmarking algorithms. The blue curve in the figure corresponds to the histogram of the score obtained for the authorized users, and the red curve depicts the histogram of the scores for unauthorized users. The histogram of the $\mathbf{z}$ scores obtained from AuthNet-R and AuthNet-D are depicted in Fig. 3.7a, 3.7b for Multi-PIE and Fig. 3.8a, 3.8b for Yale Face database B respectively. It can be noticed that for both datasets, AuthNet very effectively separates authorized and unauthorized user samples, with minimal mixing of authorized and unauthorized user distributions. The scores of the sigmoid output obtained from the AuthNet encoder classifiers are shown in Fig. 3.7c and 3.8c. It can be observed that the output of the sigmoid activation function is peaked at 0 and 1. There is a noticeable spillover in the area in between. This is the reason behind the lower EER and GAR at small values of FAR. The histogram of the sigmoid output obtained from FaceNet and ArcFace embeddings classifiers is depicted in Fig. 3.7d, 3.7e for Multi-PIE and Fig. 3.8d, 3.8e for Yale Face database B, respectively. In both cases, it is possible to appreciate a non-perfect separation of the scores: these misclassified users eventually lead to lower performance.

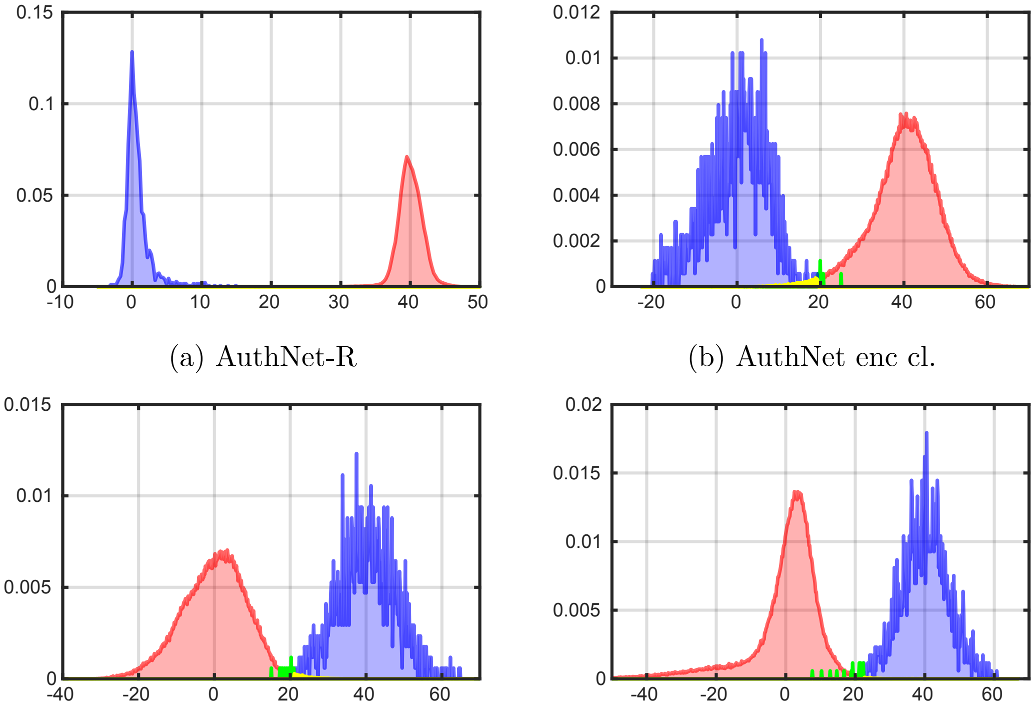
Lastly, Fig. 3.10a and 3.10b illustrates the ROC comparison of AuthNet to other benchmark techniques on the users' aggregated scores. It can be seen that the ROC curves for AuthNet lie above all other comparison methods and consistently achieve higher GAR even at very small values of FAR, proving its superiority.

Fingerprint authentication

In order to show that AuthNet works seamlessly across different biometric traits. We also test AuthNet for fingerprint authentication. For the fingerprints authentication, we evaluate the methods on FVC 2006 DB2 dataset, detailed in Sec. 3.5.1.

We compare AuthNet with AuthNet encoder classifier, Verifinger [97], and the hybrid approach described in [81] for benchmarking. Verifinger is a renowned and commercially available system commonly employed for minutiae extraction and fingerprint matching achieving state-of-the-art performance in fingerprint identification [100].

The comparison of EER, maximum accuracy, and GAR of AuthNet at small FAR values with the comparison methods is presented in Tab. 3.3. It can be



(a) AuthNet-R

(b) AuthNet enc cl.

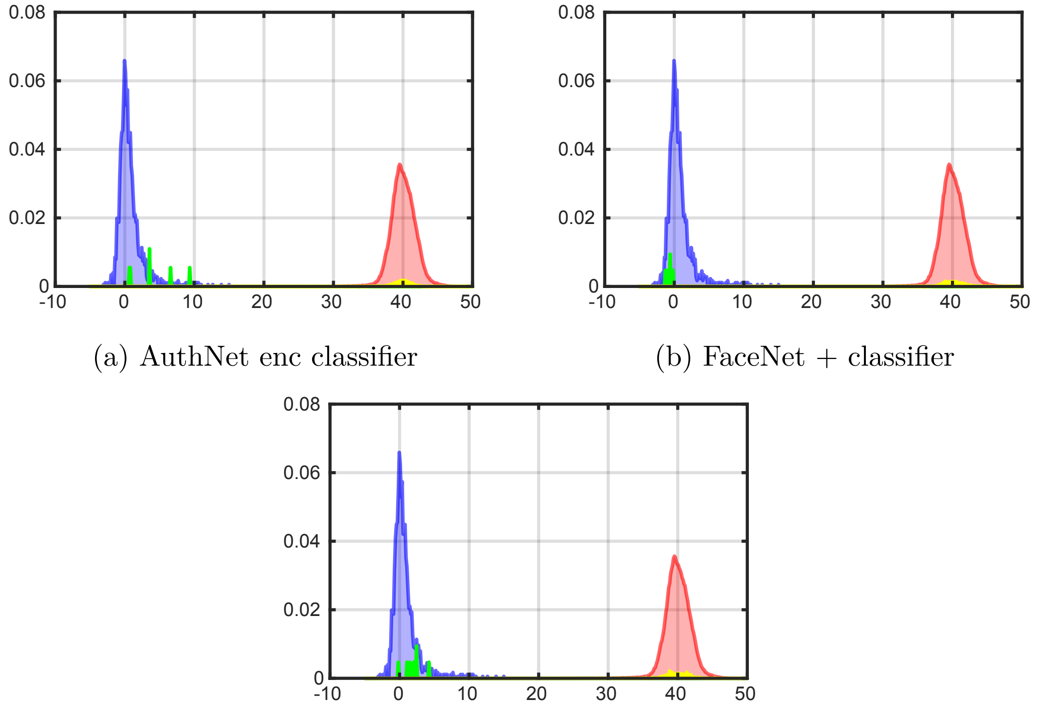
(c) FaceNet + cl.

(d) ArcFace + cl.

Figure 3.11: Normalized logits scores for correctly accepted authorized users (blue), wrongly rejected authorized users (green), correctly rejected unauthorized users (red), and wrongly accepted unauthorized users (yellow). (a), (b) Histogram of **z** decision statistics of AuthNet; (c) Histogram of the logits scores of AuthNet encoder classifier; (d) Histogram of the logits scores of FaceNet embeddings classifier; (e) Histogram of the logits scores of ArcFace embeddings classifier.

observed from Tab. 3.3, that the proposed AuthNet achieves the lowest EER and highest accuracy, significantly outperforming all benchmark methods. Regardless, differently from the previous results, it can be noted that AuthNet-D has a slight performance advantage over AuthNet-R. In general, it is hard to state which flavors of two AuthNets achieve higher performance. Undoubtedly, AuthNet performance, to some extent, is independent of the encoder network architecture. As long as the encoder network has enough capacity, any recent CNN architecture will reach, on average, high performance.

Further, for small FAR values, both AuthNet-R and AuthNet-D achieve high values of GAR. The other approaches including Verifinger, AuthNet encoder classifier, and hybrid approach also achieve small EER values; yet, it can be noted that the GAR values significantly drop as the FAR values are decreased, which is not the case with AuthNet.



(a) AuthNet enc classifier

(b) FaceNet + classifier

(c) ArcFace + classifier

Figure 3.12: Correct mapping of the misclassified users by other methods using AuthNet: mapping of the wrongly rejected authorized users (green) and wrongly accepted unauthorized users (yellow) by competing methods on AuthNet, (a) misclassified users of AuthNet encoder classifier mapped on AuthNet; (b) misclassified users of FaceNet embeddings classifier mapped on AuthNet; (c) misclassified users of ArcFace embeddings classifier mapped on AuthNet.

Further, Fig. 3.9a and 3.9b show that the proposed method AuthNet very well separates the authorized and unauthorized users. Contrarily, when using non-deep learning approaches including Verifinger as shown in Fig. 3.9d, and the hybrid approach in Fig. 3.9e, the authorized and unauthorized users do not have a clear separation of scores. As a result, the corresponding regions are not well-behaved. From Fig. 3.9c it can be noticed that similarly to the case of face datasets, while AuthNet encoder classifier provides a separation between the scores, some “leakage” is also introduced.

Finally, in Fig. 3.10c, the ROC comparison of AuthNet to other fingerprint authentication methods is illustrated. In Fig. 3.10c, the red curve shows the GAR at different FAR values obtained by AuthNet. It can be observed that the ROC curve of AuthNet lies above other benchmarking methods. Furthermore, it can be clearly observed here that at small values of FARs, AuthNet surpasses all the other

Dataset	Method	ΔGAR@10^{-1}FAR%	ΔGAR@10^{-2}FAR%	ΔGAR@10^{-3}FAR%	ΔMax accuracy
YTF	**AuthNet-R**	**0***	**0.056**	**5.852**	**0.487**
	AuthNet enc. classifier	0*	0.227	22.273	0.658
	FaceNet	0.632	18.309	22.038	0.897
	ArcFace	1.132	11.989	22.501	1.842
LFW	**AuthNet-R**	**0***	**0***	**3.068**	**0.309**
	AuthNet enc. classifier	0*	0.283	9.773	0.621
	FaceNet	0.611	19.762	22.102	0.960
	ArcFace	1.189	24.829	45.965	0.387
CALFW	**AuthNet-R**	**0***	**0***	**1.080**	**0.136**
	AuthNet enc. classifier	0*	0.227	35.568	0.648
	FaceNet	0.622	18.521	21.725	1.100
	ArcFace	1.330	8.057	15.001	1.770
Caltech 101	**AuthNet-R**	**0***	**0***	**6.762**	**0.381**
	AuthNet enc. classifier	0*	13.068	88.470	1.468
	FaceNet	0.609	22.497	25.341	1.190
	ArcFace	1.130	5.398	15.342	1.421

Table 3.5: Absolute performance drop comparison of AuthNet and benchmarking methods when trained on MultiPIE and tested on different datasets. We mark as 0* values below the minimum achievable precision, i.e. smaller than $5.6 \cdot 10{-}4$.

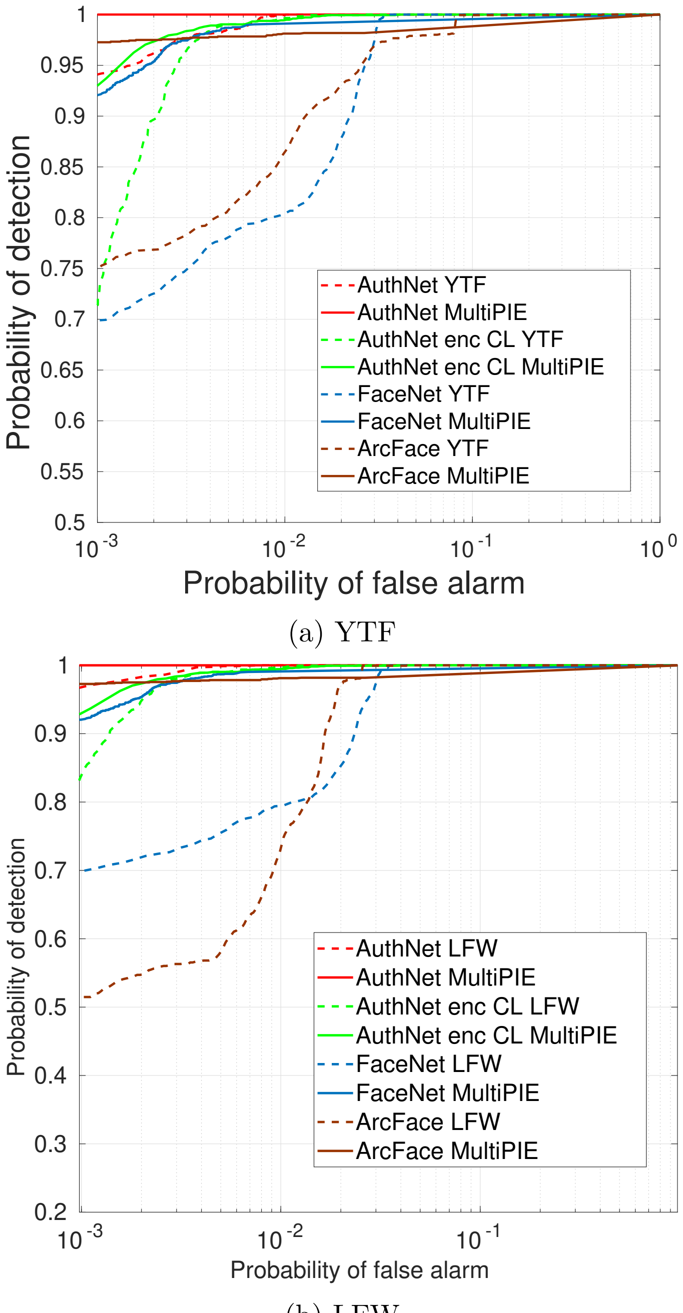
competing algorithms significantly by achieving the highest GAR values.

3.6 In-depth analysis of AuthNet

To better understand the performance improvement of AuthNet concerning competing methods, a more in-depth technical analysis is provided to explain how the regularization of the distributions performed by AuthNet yields fewer misclassifications than existing methods. Further, it is shown how AuthNet can correctly classify samples that are misclassified by competing approaches.

3.6.1 Motivation behind higher misclassification rate by competing methods:

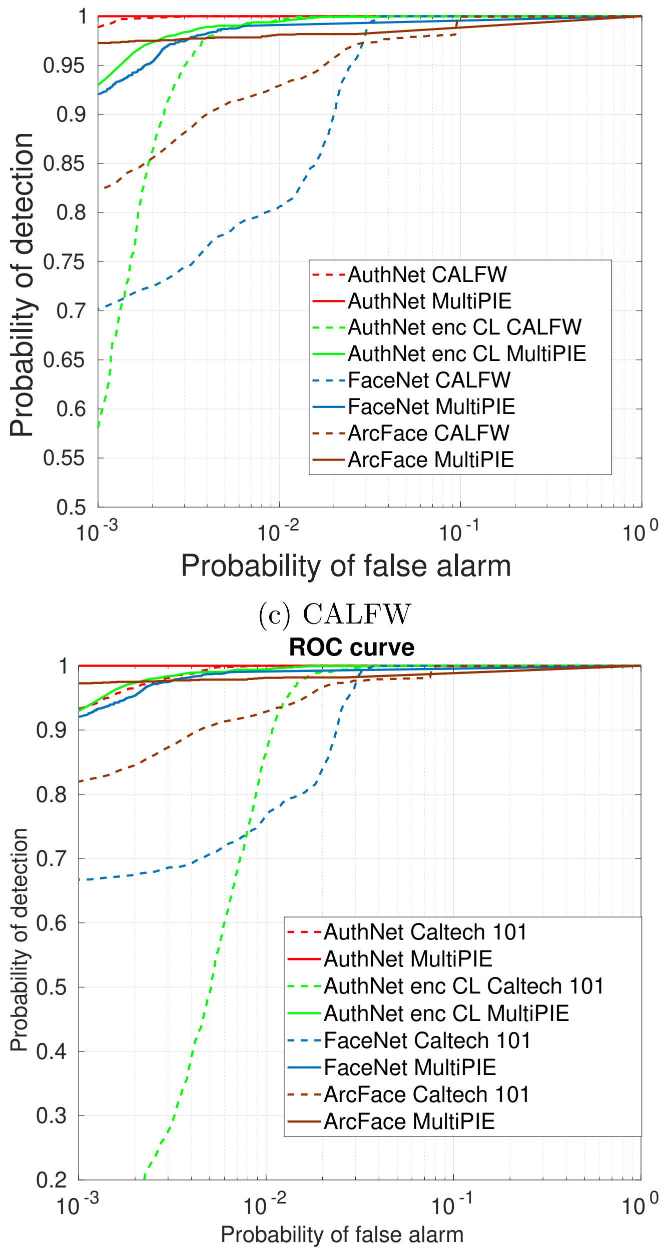
Firstly in Fig. 3.11 the latent space outputs of AuthNet and the logit scores obtained through the competing methods, normalized to the target means of $\mu_1 = 0$ for the authorized users and $\mu_0 = 40$ for unauthorized users are presented; this normalization permits us to compare these methods with AuthNet directly. It can be noted that the logit scores of the other methods naturally tends to be Gaussian, from the central limit theorem [95, 96]. During the training of AuthNet, the target distributions are enforced to follow well separated Gaussian distributions, with predefined mean and standard deviation. This is not particularly enforced for traditional classification methods, which leads to distributions with unpredictable mean and standard deviation. Consequently, it can be observed in Fig. 3.11 that compared to AuthNet, higher variance with heavier tails is exhibited by the normalized logit score distributions of the competing methods. Moreover, in Fig. 3.11



(a) YTF

(b) LFW

normalized logit scores for correctly accepted authorized users (blue), wrongly rejected authorized users (green), correctly rejected unauthorized users (red), and wrongly accepted unauthorized users (yellow) are highlighted. It can be clearly observed that for AuthNet, the authorized and unauthorized users scores are well separated based on the predefined target distributions, resulting in very few misclassifications, i.e., false rejections of authorized users (green) and false acceptance



(c) CALFW

(d) Caltech 101

Figure 3.13: ROC comparison of AuthNet and benchmarking methods when tested on the same dataset used during training (MultiPIE) and on face (YTF, LFW, CALFW) and non-face (Caltech 101) datasets that have not been used during training. In all the cases, AuthNet performs consistently and give stable GAR at different FAR values.

of unauthorized users (yellow) area. However, for the competing methods, the logit score distributions of the authorized and unauthorized users are broader, leading to a higher number of misclassifications as can be observed from the green and yellow areas.

Standard deviation σ and kurtosis β_2 of the latent space features of AuthNet and the normalized logit scores obtained by different methods are reported in Tab. 3.4. The table shows that the lack of regularization of the distributions in the competing methods tends to have much higher σ. Similarly, the competing methods' distributions are heavy-tailed, which can be seen from the measured values of β_2. This points out a higher spread of the authorized and unauthorized user distributions with respect to the mass center, resulting in a higher number of misclassifications.

In Fig. 3.12, depicts latent features obtained by AuthNet, green highlights the latent feature outputs corresponding to authorized users that are wrongly rejected by competing networks, whereas yellow highlights the features corresponding to wrongly accepted unauthorized users.

In all the cases, AuthNet maps the wrongly accepted unauthorized users near the mass center of correctly rejected unauthorized users, i.e., the red area. Also, AuthNet correctly maps the wrongly rejected authorized users in the right class in the blue area.

In summary, defining well-separated target Gaussian distributions having specified mean and standard deviation during training avoids the spread of the authorized and unauthorized users samples yielding a lower number of misclassifications.

3.7 Robustness analysis

In this second set of experiments, we show that regularizing the latent space to simple target distributions not only leads to improved accuracy but also more robust authentication. Particularly AuthNet and the benchmark methods trained on MultiPIE are tested on datasets that the network has not seen during the training. Additionally, we also test the robustness of the proposed approach against targeted perturbations.

3.7.1 Evaluation on new datasets not seen during training

To show the robustness and resilience of AuthNet against *face* datasets that the network has never seen during training, we test AuthNet-R and competing methods trained on MultiPIE on LFW [101], YTF [102], and CALFW [103] datasets. Fig. 3.13 shows the ROC comparison of methods trained and tested on MultiPIE versus the same methods trained on MultiPIE and tested on YTF, LFW, and CALFW datasets, for the class of unauthorized users (note that in this setup the unauthorized users are not present in the test dataset). The solid curves depict the results

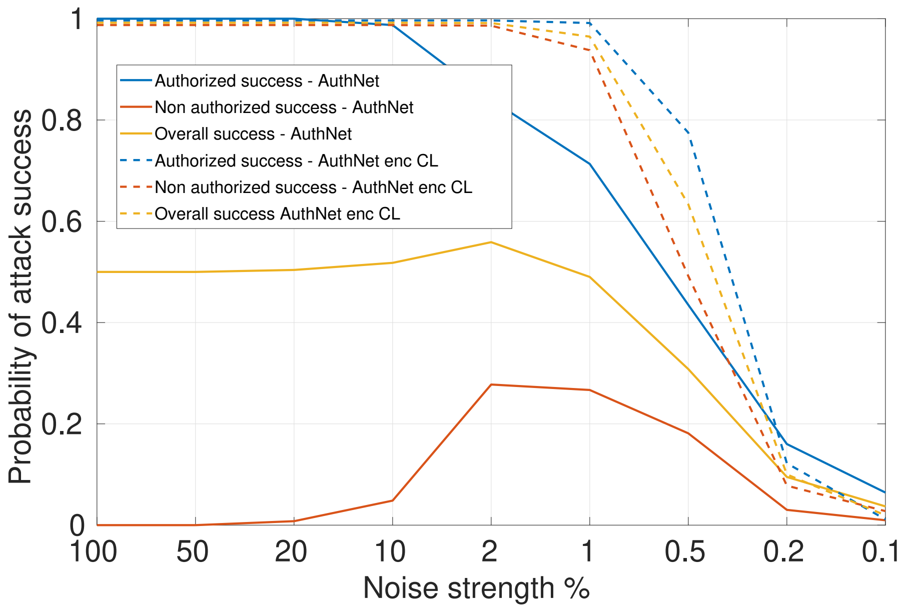


Figure 3.14: Probability of success of FGSM for authorized, unauthorized users and overall success as a function of the noise strength for AuthNet-R and AuthNet-R encoder classifier.

when methods are trained and tested on the same dataset, the dotted curves depict the test results on the datasets which the network has not seen during training. The robustness is measured in terms of the performance drop on the datasets that have not been seen during the training. It can be observed that AuthNet is robust against the datasets which were not presented at training time: it correctly maps the samples from these datasets to the unauthorized target distribution. This effect is more significant at small FAR (10^{-3}) where a large performance drop can be observed for the competing methods, whereas AuthNet maintains high GAR value, outperforming them by a big margin.

For a more detailed analysis, Tab. 3.5 reports the absolute difference in GAR at different values of FAR and the maximum accuracy difference achieved by different methods when tested on MultiPIE versus the other datasets. It can be seen that AuthNet consistently outperforms all competing methods, yielding a very small performance drop when tested on different datasets. The effect is very evident at small values of FAR.

To further evaluate the robustness of AuthNet, we also considered a non-face dataset: we test AuthNet and competing methods trained on MultiPIE on Caltech 101 [104] dataset. This dataset does not include faces and it is made of images of objects belonging to 101 different categories. From both Fig. 3.13d and Tab. 3.5 it can be observed that the performance drop is very significant for the competing methods. Conversely, AuthNet still maps the images of Caltech 101 to the unauthorized distribution giving stable results even at small FAR values.

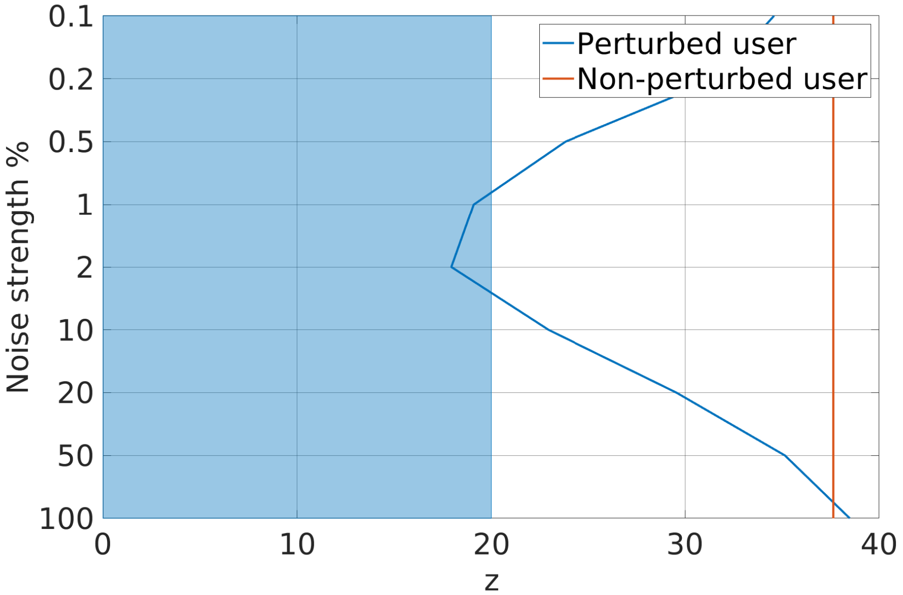


Figure 3.15: Trajectory of decision statistics for a perturbed sample (from an unauthorized user) in the latent space at different noise strength levels.

The results in this section show that regularizing the latent space using well-behaved target distributions leads to robust authentication against features that have never been seen before. Furthermore, the behavior of the non-authorized region of AuthNet is consistent across different datasets.

3.7.2 Evaluation on targeted perturbations

Further, we analyze the robustness of the proposed AuthNet against the targeted perturbations. We consider the white-box Fast Gradient Sign Method (FGSM) [16] due to its simplicity and speed in crafting the perturbations. In FGSM, the input samples are modified to maximize the loss based on the backpropagated gradients. The model back propagates to the input data to calculate $\nabla_x J(\theta, \mathbf{x}, a)$, then the input samples are adjusted by a step of ϵ in the direction of $\text{sign}(\nabla_x J(\theta, \mathbf{x}, a))$ that will maximize the loss.

To highlight the advantages of learning the mapping instead of the boundaries, we compare AuthNet-R with the AuthNet encoder classifier trained on Multi-PIE. The motivation is to show that for traditional methods producing arbitrary boundaries, it is possible to craft samples that lead to incorrect classification with minimal perturbation. In contrast, for the proposed method, this is much more difficult, leading to improved robustness.

We perturb every test sample with ℓ_∞ bounded perturbation and aggregate the results both for AuthNet and AuthNet encoder classifier. Noise vector $\mathbf{n}$ is defined such that $\|(\mathbf{n})\|_\infty \leq \epsilon$ where noise strength ϵ is defined as the ratio $\|(\mathbf{n})\|_\infty / \|(\mathbf{x})\|_\infty$. As an example, 100% noise strength indicates the model can corrupt the image with noise values within the input image's full range.

Fig. 3.14 illustrates the probability of success of FGSM as a function of the noise strength. For AuthNet, it can be observed that to move the authorized users into the unauthorized region ($\mathbf{z} < 20$), we need a large noise strength, i.e., greater than 10% of the maximum pixel values of the input images, to have a high probability of success. The probability of success decreases by lowering ϵ accordingly. Conversely, it is more challenging to grant access to unauthorized users. The maximum probability of success is 0.27, which is reached at 2% noise strength. Additionally, even for substantial perturbations, the probability of success in such a setting is close to zero. This is attributable to the latent space regularization of AuthNet: authorized users are strictly enclosed within the high mass region of $\mathbb{P}_1$. For strong perturbations, the likelihood of the perturbed samples for unauthorized users increases. Additionally, we investigate this effect in Fig. 3.15 where we show the trajectory of $\mathbf{z}$ in the latent space: for a perturbed sample of an unauthorized user as a function of the noise level. It can be observed for large perturbations that $\mathbf{z}$ stays within the high mass region of $\mathbb{P}_0$. Also, for limited values ofϵ, i.e., less than 1%, the value of $\mathbf{z}$ remains close to 40. The region between these limits may lead to misclassification of unauthorized users. This behavior is interpreted as follows; the regularized decision boundary provided by AuthNet does not allow to choose an easy path for crossing the boundary from a generic point within the decision region, i.e., every point on the other side of the boundary tends to be equally far away. In Fig. 3.14 we compare these results of AuthNet with those of the AuthNet encoder classifier; it is immediate to notice that overall FGSM is much more successful for large noise strength. Furthermore, in this case, FGSM targeting authorized users is more successful. This proves our conjecture that the likelihood of targeted perturbations to succeed can be reduced by properly regularizing the latent space. The highly complex boundaries learned through a classifier are more vulnerable to adversarial perturbations.

3.8 Conclusions

A novel approach for biometric authentication based on adversarial learning in which the latent space regularization leads to improved robustness and accuracy of the biometric authentication is presented. The behavior can be attributed to the fact that the non-linear boundaries learned by standard deep learning classifiers become very complex as they try to fit the training data, leaving room for misclassification. For AuthNet, adversarial learning enables much simpler boundaries by mapping the input space into the latent space. Experimentation on multiple large biometric datasets with several state-of-the-art benchmark methods show that AuthNet consistently outperforms existing techniques. Additionally, it was shown that regularizing the latent space makes the architecture less vulnerable to targeted and untargeted perturbations.

Chapter 4

Learning mappings onto regularized latent spaces for biometric authentication

Like the previous chapter, this chapter also takes into account the generic biometric authentication scenario. For introduction and background review, please refer to sections 3.1 and 3.2. This chapter illustrates that the mapping of biometric traits onto target distributions can also be achieved using statistical approaches. The resulting method, RegNet, uses KL-divergence to shape the latent space according to the target distributions. Additionally, the chapter compares the two methods based on adversarial learning and statistical approach.

4.1 Proposed Method

The key aim of RegNet is to learn a mapping onto well-behaved target distributions in latent space from the distribution of the input biometric traits of authorized and unauthorized users using a statistical method alternative to adversarial learning as done in chapter 3. In RegNet we propose a network that *directly* generates samples from the intended distribution instead of adversarial training. The authorized user's biometric traits should be mapped into a target distribution whose mass center is far enough from the unauthorized user's target distribution. Since the latent space is well behaved, it is possible to distinguish between two classes using a simple thresholding decision.

RegNet works in two phases: enrolment and authentication as a biometric authentication method. In the following, these two stages, which are specifically linked to training and testing phases within the context of deep neural networks, are discussed.

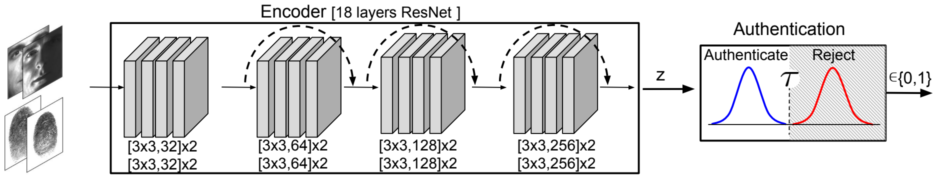


Figure 4.1: RegNet architecture. The biometric traits are given as input to the encoder; the output is a sample $\mathbf{z}$ from either $\mathbb{P}_1$ or $\mathbb{P}_0$. During the authentication phase, given $\mathbf{z}$ a thresholding decision can be applied to determine the user's class.

4.1.1 Enrollment

In this phase the network has to learn the distribution of the biometric traits of the authorized user (respectively unauthorized users) and has to generate a sample drawn from the authorized (respectively unauthorized) target distribution. It becomes evident that, in order to define a proper loss function, we should minimize a suitable distance metric between the distributions of the generated samples and the target ones. Let $\mathcal{B} = \{\mathcal{B}_{a=0}, \mathcal{B}_{a=1}\}$ denote the set of all possible biometric traits and $a \in \{0{,}1\}$ an indicator variable such that $a = 1$ represents the authorized user and $a = 0$ represents all other unauthorized users. Thus, let us first define the desired target distributions $\mathbb{P}_1$ and $\mathbb{P}_0$ (for authorized and unauthorized users respectively) as two multivariate Gaussian distributions over a d-dimensional space:

$$\mathbb{P}_1 = \mathcal{N}(\boldsymbol{\mu}_{T1}, \boldsymbol{\Sigma}_{T1}),\ \mathbb{P}_0 = \mathcal{N}(\boldsymbol{\mu}_{T0}, \boldsymbol{\Sigma}_{T0}),$$

where $\boldsymbol{\Sigma}_{T1} = \sigma^2_{T1}\mathbb{I}_d$ and $\boldsymbol{\Sigma}_{T0} = \sigma^2_{T0}\mathbb{I}_d$ are defined as diagonal covariance matrices and $\boldsymbol{\mu}_{T1} = \mu_{T1}\mathbf{1}^T$, $\boldsymbol{\mu}_{T0} = \mu_{T0}\mathbf{1}^T$ are the mean vectors.

At this point, in order to define a suitable distance metric let us define the output of the encoding network as $\mathbf{z} = H(\mathbf{x})$, where $\mathbf{z} \in \mathbb{R}^d$ is the latent mapping and $\mathbf{x} \in \mathbb{R}^n$ is the input biometric trait. The goal is to learn an encoding function of the input biometric trait $\mathbf{z} = H(\mathbf{x})$ such that $\mathbf{z} \sim \mathbb{P}_1$ if $\mathbf{x} \in \mathcal{B}_{a=1}$ and $\mathbf{z} \sim \mathbb{P}_0$ if $\mathbf{x} \in \mathcal{B}_{a=0}$, with $\mathbb{P}_1$ and $\mathbb{P}_0$ the target distributions in the latent space.

We are now interested in computing the statistics of the generated samples $\mathbf{z}$, thus we should recall that during the training the network is given as input a batch of biometric traits $\mathbf{X} \in \mathbb{R}^{b\times n}$ with b being the batch size, thus resulting in $\mathbf{z} \in \mathbb{R}^{b\times d}$ after the encoding. Therefore, we can compute the first and second order statistics (over a batch) of the encoded representations $\mathbf{z}_a, \mathbf{z}_b$ related to authorized ($\boldsymbol{\mu}_{O1}$,$\boldsymbol{\Sigma}_{O1}$) and unauthorized ($\boldsymbol{\mu}_{O0}$,$\boldsymbol{\Sigma}_{O0}$) input biometric traits respectively. More specifically, we have that $\boldsymbol{\mu}^{(i)}_{O1} = \mathbb{E}[\mathbf{z}^{(i)}_a]$ and $\boldsymbol{\Sigma}^{(ii)}_{O1} = \text{var}(\mathbf{z}^{(i)}_a)$, where $^{(i)}$ denotes the i-th colum and $^{(ii)}$ the i-th diagonal entry.

Having defined the statistics of both target and encoded samples distributions, we can define a suitable metric to compare how far the distributions are from each other. More in detail we employ the KL divergence, which for multivariate Gaussian distributions (in case of authorized input biometric traits) can be written as:

$$\mathcal{L}_a = \frac{1}{2}\Bigg[\log\frac{|\mathbf{\Sigma}_{T1}|}{|\mathbf{\Sigma}_{O1}|} - d + \mathrm{tr}(\mathbf{\Sigma}_{T1}^{-1}\mathbf{\Sigma}_{O1}) + \\ + (\boldsymbol{\mu}_{T1} - \boldsymbol{\mu}_{O1})^{\intercal}\mathbf{\Sigma}_{T1}^{-1}(\boldsymbol{\mu}_{T1} - \boldsymbol{\mu}_{O1})\Bigg]$$

For the case of diagonal covariance matrices we are considering can be rewritten as

$$\mathcal{L}_a = \frac{1}{2}\left[\log\frac{\sigma_{T1}^{2d}}{\prod_i \mathbf{\Sigma}_{O1}^{(ii)}} - d + \frac{\sum_i \mathbf{\Sigma}_{O1}^{(ii)}}{\sigma_{T1}^2} + \frac{||\boldsymbol{\mu}_{T1} - \boldsymbol{\mu}_{O1}||_2}{\sigma_{T1}^2}\right].$$

In a similar fashion we can obtain $\mathcal{L}_u$ by considering the statistic's of both target and encoded distributions in the case of unauthorized input biometric traits.

Then, the loss function which the encoder network has to minimize is given by $\mathcal{L} = \frac{1}{2}\mathcal{L}_a + \frac{1}{2}\mathcal{L}_u$, which achieves its minimum when the statistics of the two generated distributions will match that of the target ones.

At this point we note that we are shaping the distribution of the encoded samples by only enforcing first and second order statistics. Indeed, from our experiments we have observed that these statistics are sufficient to shape the encoded samples distributions to closely follow the target ones. This leads us to conjecture that the encoder output tends to a maximum entropy distribution (Gaussian) and thus first and second order moments are sufficient to shape the latent space as intended.

4.1.2 Authentication

Similar to AuthNet, the trained encoder is used for user authentication after the enrollment phase is completed; see section 3.3.4.

4.1.3 Architecture details

RegNet addresses image biometric data. Contrarily to AuthNet, which employees encoder and discriminator sub-networks during the training phase, RegNet only employees the encoder network same as of AuthNet, see Fig. 3.3 which significantly reduce the total parameters. The encoder is a convolutional neural network, we use in particular a four-block ResNet-18 architecture [105] consisting of increasing 3×3 filters. The last layer is a fully connected layer which maps the final filter output to $\mathbf{z}$: the d-dimensional latent representation. The biometric traits are given as input to the encoder; the output is a sample $\mathbf{z}$ from either $\mathbb{P}_1$ or $\mathbb{P}_0$. We set $d = 3$ for the experiments, as it results in better separation and higher performance. In addition, we set $\mu_{T1} = 0$, $\mu_{T0} = 40$ and $\sigma_{T1} = \sigma_{T0} = 1$. We use Adam optimizer to optimize the network and apply stochastic gradient descents to mini-batches of 100 samples. During the authentication phase, given $\mathbf{z}$ a thresholding decision can be applied to determine the user's class.

Dataset	Method	EER%	GAR@10^{-1}FAR%	GAR@10^{-2}FAR%	Accuracy@EER
Face - Yale B	**RegNet**	**0.023**	**100.0**	**100.0**	**99.977**
	RegNet enc. classifier	0.040	100.0	100.0	99.960
	AuthNet	**0.019**	**100.0**	**100.0**	**99.991**
	FaceNet	1.286	98.819	98.712	98.714
	ArcFace	0.893	99.159	99.108	99.107
	Fisherfaces	15.351	84.215	61.135	84.649
Face - Multi-PIE	**RegNet**	**0.045**	**100.0**	**100.0**	**99.955**
	RegNet enc. classifier	0.676	100.0	99.432	99.324
	AuthNet	**0.001**	**100.0**	**100.0**	**99.998**
	FaceNet	0.930	99.368	99.201	99.070
	ArcFace	1.811	98.811	98.125	98.189
	Fisherfaces	32.620	10.002	2.800	67.379

Table 4.1: Performance comparison of RegNet with respect to other biometric authentication schemes for faces. Average values computed on the aggregated scores are reported.

Dataset	Method	EER%	GAR@10^{-1}FAR%	GAR@10^{-2}FAR%	Accuracy@EER
Fingerprint FVC 2006	**RegNet**	**0.476**	**100.0**	**99.934**	**99.524**
	RegNet enc. classifier	0.565	100.0	99.845	99.435
	AuthNet	**0.339**	**100.0**	**100.0**	**99.740**
	Verifinger	0.758	100.0	99.796	99.361
	Hybrid approach [81]	3.200	98.182	94.854	96.799

Table 4.2: Performance comparison of RegNet with respect to other biometric authentication schemes for fingerprints. Average values computed on the aggregated scores are reported.

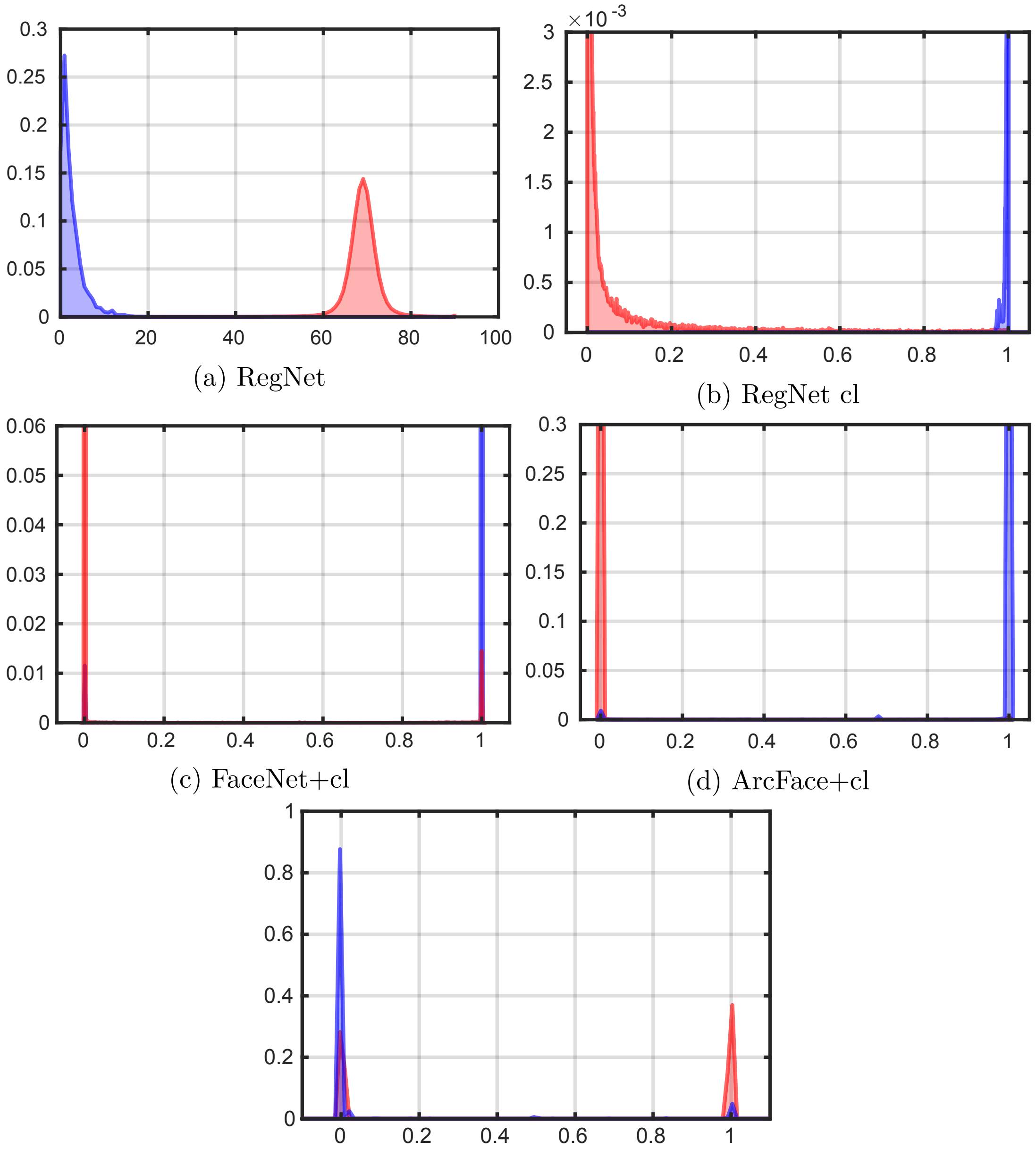
Preprocessing and training parameters

The network is trained using Adam optimizer [88] with an iterative algorithm, as discussed in [52]. Weight decay of 0.0004 and dropout of 0.7 is employed. Initially, for the first 5000 iterations, the learning rate is 0.001 and is then decreased by a factor of 10 after every 2000 iteration. In total, the network is trained for 10000 iterations. In RegNet, the only preprocessing considered is the energy normalization of the input images

4.2 Experimental settings and results

4.2.1 Datasets

Similar to AuthNet, for Face authentication task we employ the CMU Multi-PIE Dataset [90] and cropped version of *extended Yale Face Database B* [91]. **Fingerprint** authentication experiments are carried out on *Fingerprint Verification Competition (FVC 2006) DB2* dataset [92]. For details please refer to section 3.5.1.



(a) RegNet

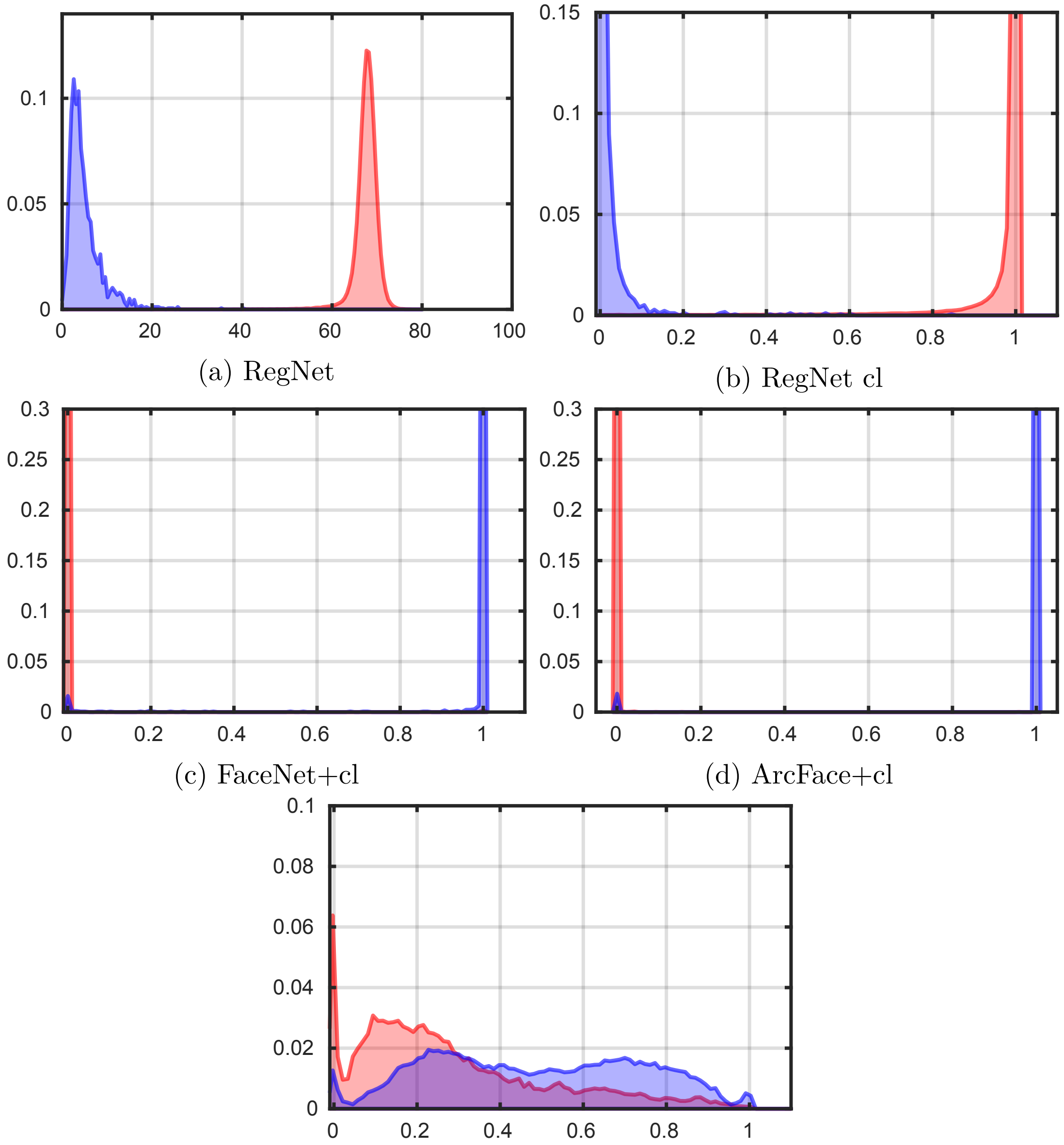
(b) RegNet cl

(c) FaceNet+cl

(d) ArcFace+cl

(e) Fisherfaces

Figure 4.2: Face authentication scores for authorized users (blue) and unauthorized users (red) for Yale B. (a) Histogram of $||\mathbf{z}||_2$ decision statistics of RegNet; (b) Histogram of the sigmoid outputs of RegNet encoder classifier; (c) Histogram of the sigmoid outputs of FaceNet embeddings classifier; (d) Histogram of the sigmoid outputs of ArcFace embeddings classifier; (e) Histogram of the normalized matching distances of Fisherfaces. The plots in (b)-(c) depict a detailed view to better appreciate the leakage effects.



(a) RegNet

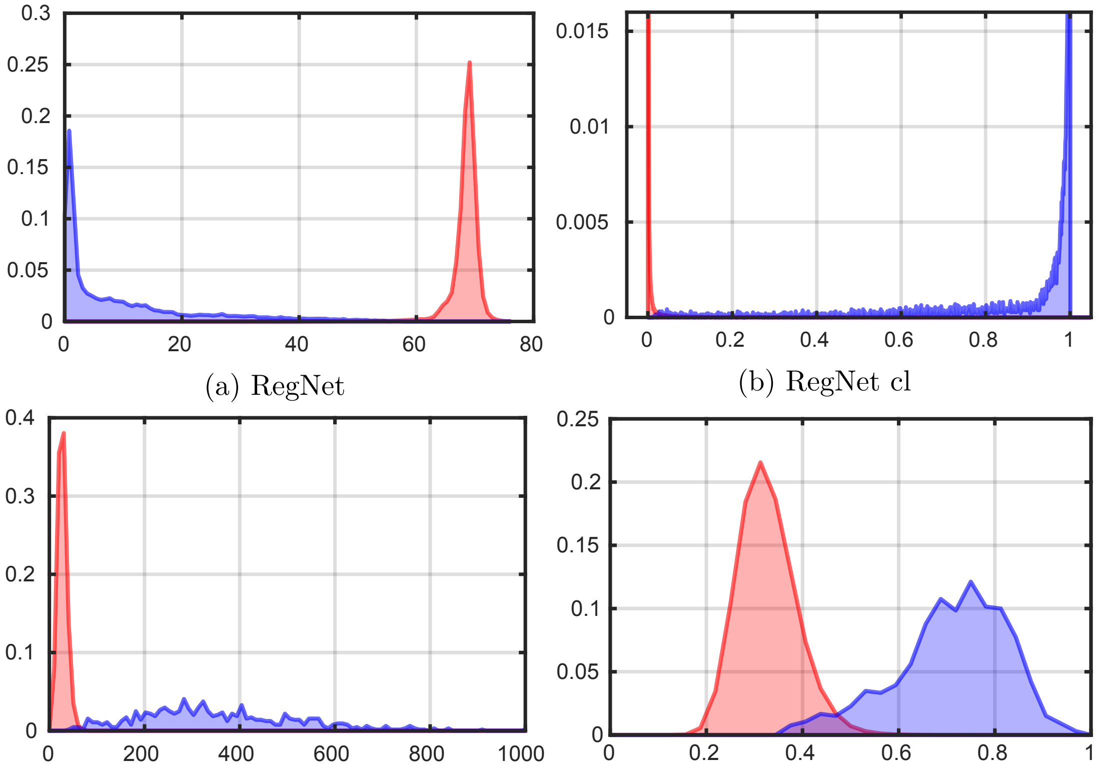
(b) RegNet cl

(c) FaceNet+cl

(d) ArcFace+cl

(e) Fisherfaces

Figure 4.3: Face authentication scores for authorized users (blue) and unauthorized users (red) for Multi-PIE. (a) Histogram of $||\mathbf{z}||_2$ decision statistics of RegNet; (b) Histogram of the sigmoid outputs of RegNet encoder classifier; (c) Histogram of the sigmoid outputs of FaceNet embeddings classifier; (d) Histogram of the sigmoid outputs of ArcFace embeddings classifier; (e) Histogram of the normalized matching distances of Fisherfaces. The plots in (b)-(c) depict a detailed view to better appreciate the leakage effects.



(a) RegNet

(b) RegNet cl

(c) VeriFinger

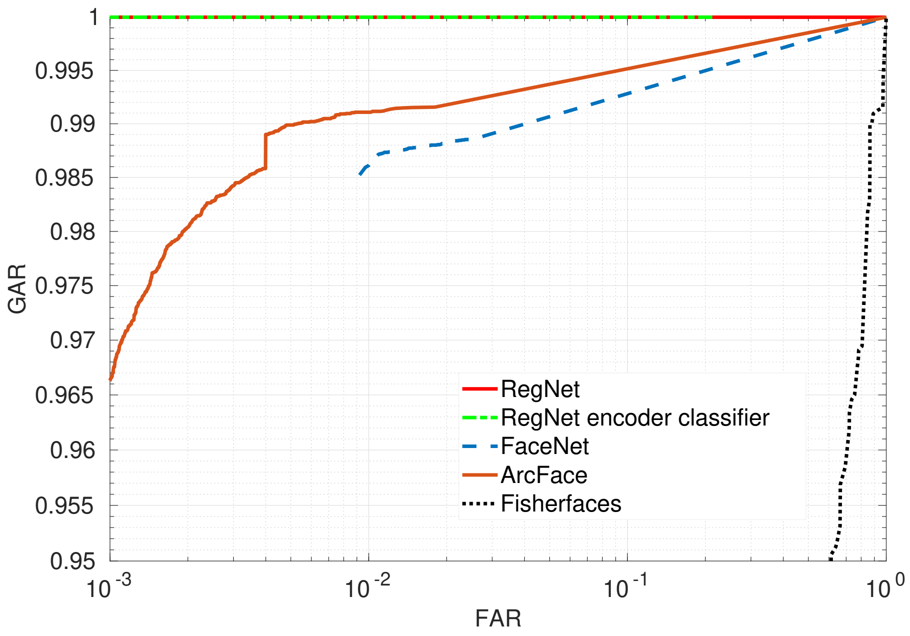
(d) Hybrid app

Figure 4.4: Fingerprint authentication scores for authorized users (blue) and unauthorized users (red). (a) Histogram of $||\mathbf{z}||_2$ decision statistics for RegNet; (b) Histogram of the sigmoid outputs of the RegNet encoder classifier; (c) histogram of the matching scores of Verifinger; (d) histogram of the matching scores of the hybrid approach. The plot in (b) depicts a detailed view to better appreciate the leakage effects.

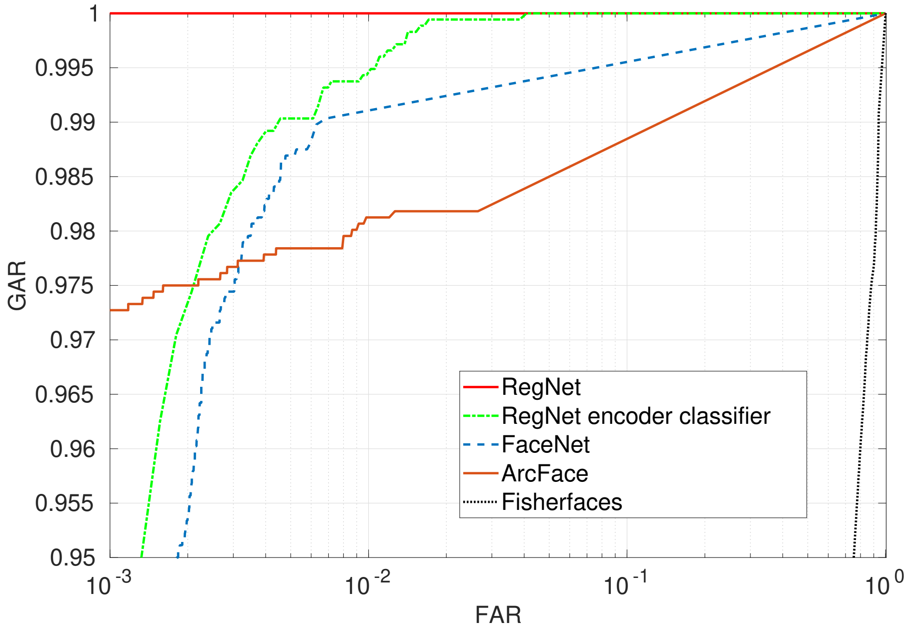
4.2.2 Results

Face authentication. In this case, we compare the RegNet results with AuthNet, the RegNet encoder classifier, the Fisherfaces approach [57], the FaceNet [65] and the ArcFace [66]. The RegNet encoder classifier has the same structure as the RegNet encoder but has been trained in a more traditional fashion via sigmoid cross-entropy loss. This network does not use a variational loss function, so it allows us to evaluate how the learning of mapping leads to an improvement in performance over a conventional neural network. As far as FaceNet and ArcFace are concerned, since it is not possible to train them from scratch due to the extreme data scarcity, we compute the 512-dimensional embedding of the input images using the pre-trained CASIA WebFace dataset [99] network. The classifier is then independently trained on the embedding of each user.

AuthNet obtains the highest performance for all metrics, as shown in Tab. 4.1.

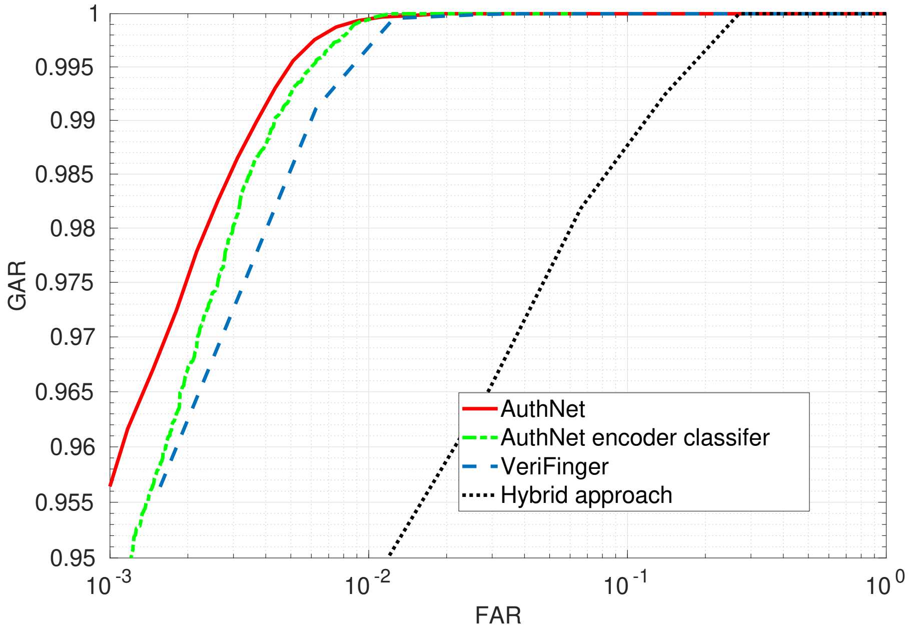


(a) Face dataset - Yale B



(b) Face dataset - MultiPIE

It is important to note that while sharing RegNet architecture, the RegNet encoder classifier results in lower performance in particular at low FAR, see Fig. 4.5(a)-(b). This implies that a more robust classification scheme is obtained by having well-defined regions in target distributions latent space. As shown in Fig. 4.3(a) RegNet efficiently separates the authorized and unauthorized users. Other methods



(c) Fingerprint dataset

Figure 4.5: ROC comparison on overall results of 32 users for faces (a)-(b) and fingerprint (c) datasets. RegNet is compared with the RegNet encoder classifier, FaceNet [65], ArcFace [66] and Fisherfaces [57] in (a)-(b); with RegNet encoder classifier, VeriFinger [97] and the hybrid approach [81] in (c).

also lead to a good separation, see Fig 4.3(b)-(c) and 4.2(b)-(c); but they fail to assign the correct score to all the unauthorized users and inflict a “leakage”. The ROC comparison in Fig. 4.5(a)-(b) more clearly illustrate this behavior. Further, it can be noted that when compared to the RegNet encoder classifier, the proposed method performs better at low FAR values.

At this point, it is interesting to note that deep learning methods can achieve higher performance when tested on the Multi-PIE data set. Although this dataset is more complex than Yale B, it has more samples due to unconstrained acquisitions. For this reason, methods that can learn complex data features will benefit. While relying on properly aligned and constrained images, traditional approaches, such as Fisherfaces, display a drop in performance when tested on the complex Multi-PIE dataset.

Fingerprint authentication. For fingerprint authentication, we compare RegNet to AuthNet, the RegNet encoder classifier, the Verifinger [97], and the hybrid approach described in [81]. As for the EER, the proposed method achieves an EER of 0.476% outperforming all other methods. The proposed methods outperforms the hybrid method and improves over the RegNet encoder classifier and verifinger in terms of genuine acceptance rate (GAR) for small FAR values. As previously observed, in the case of face authentication, it can be seen in Fig 4.4(a)

Properties	**AuthNet**	**RegNet**
Performance	high	slightly lower
Training complexity	hard	easy
Predictability	less	high

Table 4.3: Comparison of RegNet with AuthNet in terms of performance, training complexity, and predictability of the distributions.

how RegNet effectively separates authorized and unauthorized users. However, it should also be noted that the distribution of authorized users spread more in the case of a face dataset. This could be due to the relatively limited number of training samples for the authorized user, which is only 10 prior to the augmentation. For non-deep learning methods (see figure 4.4(c)-(d), there is no clear separation between the scores of the authorized and unauthorized users. Furthermore, the RegNet encoder classifier also introduces some “leakage”. In Fig 4.5(c), this aspect can be further analyzed: RegNet outperforms all other methods by achieving the highest GAR values, even for low FAR values.

The above results greatly motivate the intuition behind RegNet: learning the mapping rather than classification boundaries results in an improved performance and classifiers robustness.

4.3 Conclusions

We presented two novel approaches addressing the biometric authentication problem with deep neural networks. Rather than learning complex boundaries, the proposed methods aim to learn to map onto the target distributions allowing for simple threshold-based classification. With extensive experimentation on different biometric traits, we demonstrate that both RegNet and AuthNet are effective general-purpose biometric authentication frameworks capable of achieving low EER and good latent space separation.

However, the experiments demonstrate that training AuthNet is very difficult; it takes a much longer time for the network to converge; however, RegNet training is fast and straightforward. Further, RegNet’s latent distributions are more predictable compared to AuthNet’s. In conclusion, the general comparison shows that RegNet is easier to train and offers high precision, making it a preferred choice.

Chapter 5

BioMetricNet: deep unconstrained face verification through learning of metrics regularized onto Gaussian distributions

5.1 Introduction

Early attempts in biometric authentication needed to design handcrafted features that could capture each person's most significant traits. Moreover, a precisely aligned and illumination normalized picture was needed for them to perform well. The complexity of handling the non-linear variations which may occur later in the face image has shown that in non-ideal conditions, those techniques tend to fail.

The use of features learned from CNN networks, e.g., DeepFace [106], and DeepID [107], made a breakthrough possible. As in previous methods, the distance measure (typically $\ell 2$ norm) was being employed for the verification task once the features of two test faces have been computed: if the distance is below a certain threshold, the two test faces are classified as belonging to the same person, otherwise not. The softmax cross-entropy loss is used to compute those features. Indeed, it has been found that the ability to generalize can be improved by maximizing inter-class variance and reducing the intra-class variance. This strategy was adopted by works like [108, 69], which take into account the large margin between "contrastive" embeddings in Euclidean space. FaceNet [65] introduced the triplet loss, whereby the distance between embeddings is relative rather than an absolute distance. The introduction of anchor samples into the training process enables learning embedding, which reduces the anchor-positive distance while maximizing

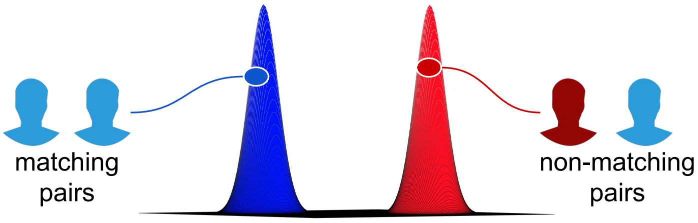


Figure 5.1: The goal of BioMetricNet is to map the input pairs onto target distributions in the latent space. Matching pairs (same user - blue) are mapped to a target distribution whose mean value is far from that of the non-matching pairs (different users - red).

the anchor-negative distance. Although this latter work has resulted in better embedding, however, it was shown that training is often complex [109]. Finally, the focus was shifted to the design of new architectures with other metrics than the ℓ_2 norm to provide tighter margins. The authors in [110] and [111] propose to use angular distance metrics to enforce a large margin between the negative examples and thus to reduce the number of false positives.

A pre-determined analytical metric for the distance between the two embeddings is employed for all the above-mentioned methods. The loss function is designed to ensure that the negative pairs have a large margin (in terms of the metric employed) while also compacting the distance between positives pairs. It is important to emphasize that the metric chosen in designing such neural networks is crucial. Indeed, shifting from the Euclidean to angular distance metrics [112, 113] lead to a major improvement in performance.

We propose a different approach: we strive to learn the most discriminative features and learn the best (possibly highly non-linear) metric to compare such features. The only condition we impose is how the metric should behave, based on whether the features are matching or not. In particular, the metric output is regularized such that the values in it follow two distinct statistical distributions: one for matching pairs and the other for non-matching pairs (see Fig. 5.1).

[114] discussed the idea to focus on the empirical distribution of feature distances to enhance their discrimination. The authors introduce the histogram loss to minimize the overlap of matching and non-matching feature pairs to obtain more regularized features. However, while this method is well suited to clustering tasks where only the relative differences between pairs are compared, it does not match the verification problem that we examine: the boundary of decision between the two histograms depends highly on the data set employed. It does not generalize well across different data distributions. The method we follow is rather different: by regularizing the latent space using target distributions, we have a known and fixed decision boundary, which generalizes well across different datasets. In [115]

and [116], this seminal idea of employing the target distributions was first presented and it is described in detail in Chapter 3 of this thesis. However, it must be emphasized that in the case of [115] and [116] latent space regularization was used to address the one-vs-all classification problem, such that biometric traits of a single user would be mapped onto a distribution and those of any other user on a different distribution. The above methods also required user-specific training, that can be inconvenient is some practical settings..

The neural network proposed here, which we refer to as BioMetricNet, in addition to learning features, shapes the decision metric so that pairs of similar faces are mapped to a distribution, while pairs of different faces are mapped to another distribution, thus eliminating the need for user-specific training. This approach has several advantages: i) The distributions are known and generally simple, the decision boundaries are also simple. This contrasts the typical behavior of neural networks, which tends to generate highly complex boundaries; ii) If the distributions are taken as Gaussian with equal variance, the optimum decision boundary is a hyperplane. This leads to a very simple classifier that learns a complex mapping to a simple latent space that mimics kernel-based methods. Moreover, Gaussian distributions are amenable to writing the loss function in closed form; iii) Mapping to known distributions allows confidences to be obtained for every test sample, as difficult pairs are mapped to the distribution tails. The distribution of the metric output values is known in BioMetricNet; this makes it possible to change the decision threshold to the desired level of false alarm or genuine acceptance rates.

The resulting design, with the best-learned metric, allows for improvement over the state-of-the-art performance on several challenging datasets, as will be shown in Sec. 5.3.7. We underline that while BioMetricNet is used on faces in this thesis, the method is general and can be used for other biometric traits.

5.2 Proposed Method

BioMetricNet strives to learn meaningful features of the input faces along with a discriminative metric to be used to compare two sets of facial features. More specifically, as depicted in Fig. 5.2, BioMetricNet is made of two sub-networks: FeatureNet and MetricNet. The former is a siamese network which processes pairs of input faces $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ and outputs a pair of facial features $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2]$ for both matching and non-matching input pairs. MetricNet is then employed to map these feature pairs onto a point $\mathbf{z}$ in a p-dimensional space in which a decision is made. These two networks are trained as a single entity to match the desired behavior. Their architecture is described more in detail in Sec. 5.2.1.

The novelty of our approach is that we do not impose any predetermined metric between $\mathbf{f}_1$ and $\mathbf{f}_2$: the metric is rather learned by MetricNet shaping the decision space according to two target distributions through the loss function, as described

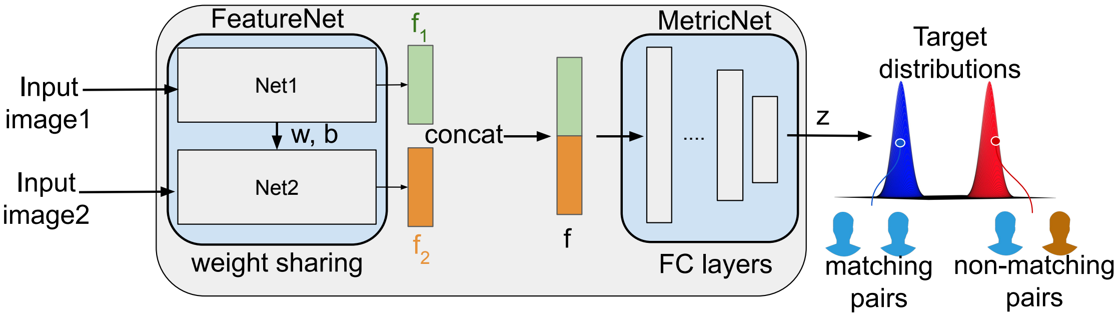


Figure 5.2: BioMetricNet architecture during the training phase. After face detection and alignment, matching and non-matching face pairs are given as an input to the FeatureNet to extract the discriminative face features from the image space $\mathbf{x}$ into feature vector space $\mathbf{f}_i \in \mathbb{R}^d$. The feature vectors are concatenated $\mathbf{f} = [\mathbf{f}_1\mathbf{f}_2] \in \mathbb{R}^{2d}$ and passed to the MetricNet which maps $\mathbf{f}$ onto well-behaved target distributions $\mathbf{z} \in \mathbb{R}^p$ in the latent space.

in the following. The loss function forces the value of the learned metric to follow different statistical distributions when applied to matching and non-matching pairs, respectively. Although arbitrary target distributions can be employed, a natural choice is to use distributions that have far-enough mass centers, lead to simple decision boundaries, and lend themselves to writing the loss function in a closed form.

For BioMetricNet, let us denote as $\mathbb{P}_m$ and $\mathbb{P}_n$ the desired target distributions for matching and non-matching pairs, respectively. We choose $\mathbb{P}_m$ and $\mathbb{P}_n$ to be multivariate Gaussian distributions over a p-dimensional space:

$$\mathbb{P}_m = \mathcal{N}(\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m),\ \mathbb{P}_n = \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n), \tag{5.1}$$

where $\boldsymbol{\Sigma}_m = \sigma_m^2 \mathbb{I}_p$ and $\boldsymbol{\Sigma}_n = \sigma_n^2 \mathbb{I}_p$ are diagonal covariance matrices and $\boldsymbol{\mu}_m = \mu_m \mathbf{1}_p^T$, $\boldsymbol{\mu}_n = \mu_n \mathbf{1}_p^T$ are the expected values. The choice of using Gaussian distributions is a very natural one in this context. Because of the central limit theorem [95], the output of fully connected layers tends to be Gaussian distributed. Moreover, if $\boldsymbol{\Sigma}_m = \boldsymbol{\Sigma}_n$, then a linear decision boundary (hyperplane) is optimal for this Gaussian discrimination problem. Therefore, while in general BioMetricNet can be trained to match arbitrary distributions, in the following we will describe this specific case. It can also be noted that using different variance for the two distributions would complicate the choice of the parameters, since the optimal variance will be specific to the considered dataset in order to match its intra and inter-class variances.

As said above, in the Gaussian case the loss function can be written in closed form. Let us define $\mathbf{x}_m$ and $\mathbf{x}_n$ as the pairs of matching and non matching face images, respectively. In the same way we define $\mathbf{f}_m$ and $\mathbf{f}_n$ as the corresponding features output by FeatureNet. MetricNet can be seen as a generic encoding function $H(\cdot)$ of the input feature pairs $\mathbf{z} = H(\mathbf{f})$, where $\mathbf{z} \in \mathbb{R}^p$, such that $\mathbf{z}_m \sim \mathbb{P}_m$ if $\mathbf{f} = \mathbf{f}_m$ and $\mathbf{z}_n \sim \mathbb{P}_n$ if $\mathbf{f} = \mathbf{f}_n$. As previously described, we want to regularize the metric space where the latent representations $\mathbf{z}$ lie in order to constrain the metric behavior. Since the distributions we want to impose are Gaussian, the Kullback-Leibler (KL) divergence between the sample and target distributions can be obtained in closed-form as a function of only first and second order statistics and can be easily minimized. More specifically, the KL divergence for multivariate Gaussian distributions can be written as:

$$\mathcal{L}_m = \frac{1}{2}\left[\log\frac{|\mathbf{\Sigma}_m|}{|\mathbf{\Sigma}_{Sm}|} - p + \mathrm{tr}(\mathbf{\Sigma}_m^{-1}\mathbf{\Sigma}_{Sm}) + (\boldsymbol{\mu}_m - \boldsymbol{\mu}_{Sm})^\intercal\mathbf{\Sigma}_m^{-1}(\boldsymbol{\mu}_m - \boldsymbol{\mu}_{Sm})\right]. \quad (5.2)$$

where the subscript S indicates the sample statistics.

Interestingly, since we only need the first and second order statistics of $\mathbf{z}$, we can capture this information batch-wise. As will be explained in detail in Sec. 5.2.2, during the training the network is given as input a set of face pairs from which a subset of $b/2$ difficult matching and $b/2$ difficult non-matching face pairs are extracted, being b the batch size. Letting $\mathbf{X} \in \mathbb{R}^{b\times r}$ with r the size of a face pair, this results in a collection of latent space points $\mathbf{z} \in \mathbb{R}^{b\times p}$ after the encoding. We thus compute first and second order statistics of the encoded representations $\mathbf{z}_m, \mathbf{z}_n$ related to matching ($\boldsymbol{\mu}_{Sm}$,$\mathbf{\Sigma}_{Sm}$) and non-matching ($\boldsymbol{\mu}_{Sn}$,$\mathbf{\Sigma}_{Sn}$) input faces respectively. More in detail, let us denote as $\mathbf{\Sigma}_{Sm}^{(ii)}$ the i-th diagonal entry of the sample covariance matrix of $\mathbf{z}_m$. The diagonal covariance assumption allows us to further simplify (5.2) as:

$$\mathcal{L}_m = \frac{1}{2}\left[\log\frac{\sigma_m^{2p}}{\prod_i \mathbf{\Sigma}_{Sm}^{(ii)}} - p + \frac{\sum_i \mathbf{\Sigma}_{Sm}^{(ii)}}{\sigma_m^2} + \frac{||\boldsymbol{\mu}_m - \boldsymbol{\mu}_{Sm}||_2}{\sigma_m^2}\right]. \quad (5.3)$$

This loss captures the statistics of the matching pairs and enforces the target distribution $\mathbb{P}_m$. For brevity we omit the derivation of $\mathcal{L}_n$ which is obtained similarly.

Then, the overall loss function which will be minimized end-to-end across the whole network (FeatureNet and MetricNet) is given by $\mathcal{L} = \mathcal{L}_m + \mathcal{L}_n$.

5.2.1 Architecture

Below we discuss FeatureNet and MetricNet's design and implementation approach.

FeatureNet

FeatureNet attempts to extract from the input pairs the most distinctive facial features. FeatureNet's architectural design is crucial. In general, one can use any advanced neural network architecture that is capable of learning good features. We use a siamese Inception-ResNet-V1 [117] in our tests thanks to its quick convergence. The output block in Inception-ResNet size is 35 ×35 × 256, followed by five blocks of Inception-ResNet-A, ten blocks of Inception-ResNet-B, and five blocks of Inception-ResNet-C. At the end of the network, we use a fully connected layer with an output dimension equal to d. The dropout rate of 0.8 is employed. The pairs of feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ in output of FeatureNet are concatenated resulting in $\mathbf{f} = [\mathbf{f}_1\mathbf{f}_2] \in \mathbb{R}^{2d}$ and given as input to MetricNet.

MetricNet

MetricNet aims to learn the best metric based on the $\mathbf{f}$ vector and map it to the latent space target distributions. MetricNet consists of 7 fully connected layers, with ReLU activation functions at each layer's output. No activation function is used in the last layer. The input size of MetricNet is equal to $2d$; the size progressively decreases by a factor of two; the final layer has an output size equal to p.

It is also noted that MetricNet can model arbitrary nonlinear correlations between the feature vectors by inputing $\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2]$. Indeed, it has proved highly efficient to use an arbitrary combination of input feature entries, see e.g., [118, 119].

5.2.2 Pairs Selection during Training

BioMetricNet chooses the hardest matching and non-matching pairs during the training, i.e., those far from the mean of target distributions and near the threshold for improved convergence. At the end of the forward pass, we pick the pairs with the $\mathbf{z}_m$ output which are sufficiently away from the mass center of $\mathbb{P}_m$, i.e., $||\mathbf{z}_m - \boldsymbol{\mu}_m||_\infty \geq 2\sigma_m$. Similarly, for non-matching pairs, we pick those that result in$||\mathbf{z}_n - \boldsymbol{\mu}_n||_\infty \geq 2\sigma_n$. Then we minimize the loss with the backward pass over a subset of $b/2$ difficult matching and $b/2$ difficult non-matching pairs, with b as the mini-batch size of difficult pairs. The backward pass can be executed only if we can obtain $b/2$ hard matching and $b/2$ hard not matching pairs; otherwise, the mini-batch is discarded.

The reasoning behind this option is the outcome of the latent space regularization. Indeed, when one traverses the latent space from $\boldsymbol{\mu}_m$ to $\boldsymbol{\mu}_n$, one moves from very similar face pairs to very dissimilar ones. Points near the threshold can be considered as pairs with a high degree of uncertainty. The network improves the mapping of “difficult” pairs as training is carried out on pairs for which it is more difficult to determine whether or not they constitute a match.

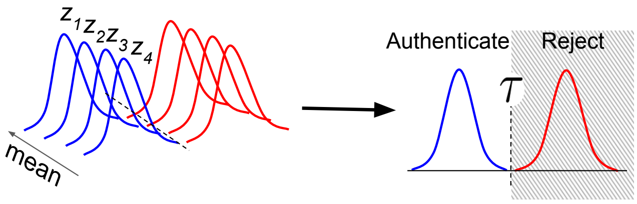


Figure 5.3: During the testing phase, we obtain the latent vectors of the input image pair and its three horizontal flips. For all experiments, the final latent space vector is calculated as $\bar{\mathbf{z}} = \frac{1}{4}\left(\sum_{i=1}^{4} \mathbf{z}_i\right)$. Pairs are classified as matching and non-matching by comparing $\bar{\mathbf{z}}$ with a threshold τ.

5.2.3 Authentication

A pair of images is passed through the entire network during the testing phase to compute the corresponding $\mathbf{z}$ metric value. Then a decision is based on this value. As stated, a hyperplane can be used for optimal decision for our choice of target distributions, i.e., we can use the test

$$(\boldsymbol{\mu}_m - \boldsymbol{\mu}_n)^T \mathbf{z} \lesseqgtr (\boldsymbol{\mu}_m - \boldsymbol{\mu}_n)^T (\boldsymbol{\mu}_m + \boldsymbol{\mu}_n)/2. \tag{5.4}$$

For $p = 1$, this boils down to comparing the scalar $\mathbf{z}$ with a threshold $\tau = (\mu_m + \mu_n)/2$.

However, we do consider an improved approach that can capture additional information: we use flipped images as done in the recent literature [111, 113] to capture additional features. Namely, when we have an image pair, we calculate the metrical $\mathbf{z}$ output on the original pair of images, with 3 pairs that result from the combinations of flipped and non-flipped images. We employ a horizontal flip defined as $(x, y) \rightarrow (width - x - 1, y)$. As a result, four metric values are obtained. Then a decision is taken on a $\bar{\mathbf{z}} = \frac{1}{4}\left(\sum_{i=1}^{4} \mathbf{z}_i\right)$ value, where $\mathbf{z}_i$ is the performance of the i-th image flip combination; see Fig.5.3. The predicted value of $\bar{\mathbf{z}}$ for matching and non-matching pairs is still equal to $\boldsymbol{\mu}_m$ and $\boldsymbol{\mu}_n$ respectively. Thus the test will continue to be valid at $\bar{\mathbf{z}}$ (5.4) . BioMetricNet is shown during the authentication phase in Fig. 5.4. P1 represents the pair of input images, and the three horizontal flips are represented as P2, P3, and P4.

5.3 Experiments

5.3.1 Experimental Settings

The network is trained using a stochastic gradient descent [120, 121] with Adam optimizer [88]. Each epoch consists of 720 individuals and has at least 5 images to ensure that the matching and non-matching pairs are enough. We set the batch

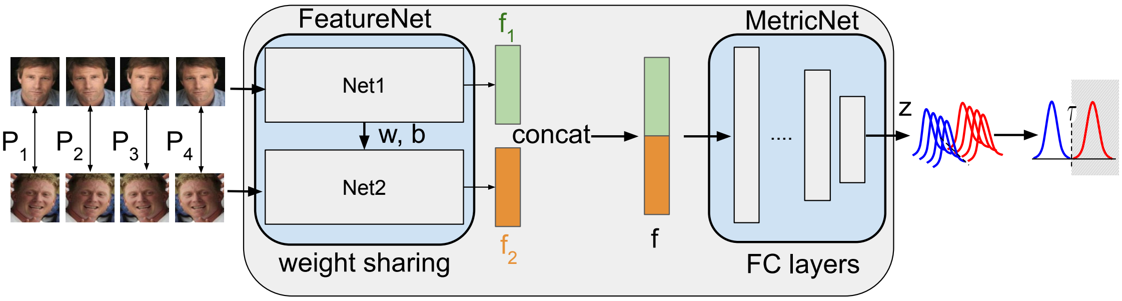


Figure 5.4: BioMetricNet architecture during the testing phase. Given a pair of images to be tested, after face detection and alignment, by accounting for all the possible horizontal flip combinations, we obtain 4 image pairs, i.e. P_1, P_2, P_3 and P_4. The latent vectors of the corresponding pairs are computed and aggregated to $\bar{\mathbf{z}}$ and compared with a threshold τ.

size b to 220 to collect statistically significant first and second-order statistics. It comprises half matching and half non-matching pairs, i.e., each batch is balanced. The initial learning rate is 0.01 with a 0.98 decay factor for every 5 epoch. The network is trained for a total of 500,000 iterations. The Weight decay is set to 2×10^{-4}. We also use dropout with a keep probability value of 0.8. All experiments are conducted in TensorFlow [122]. For the augmentation, the images are flipped horizontally.

5.3.2 Preprocessing

For preprocessing, we follow the strategies used in the recent papers [112, 113, 111]. We employ MTCNN [123] to generate normalized facial crops of size 160×160 based on five facial points both for training and testing sets. The final step is to normalize the images and constrain them to [−1,1], as in [113, 111, 112].

5.3.3 Datasets

Training

The data sets of training are the ones used commonly in recent works. In more detail, we use different datasets in the train and test phases. For training we employ Casia [124] (0.49M images having 10K identities) and MS1M-DeepGlint (3.9M images with 87K identities) [125].

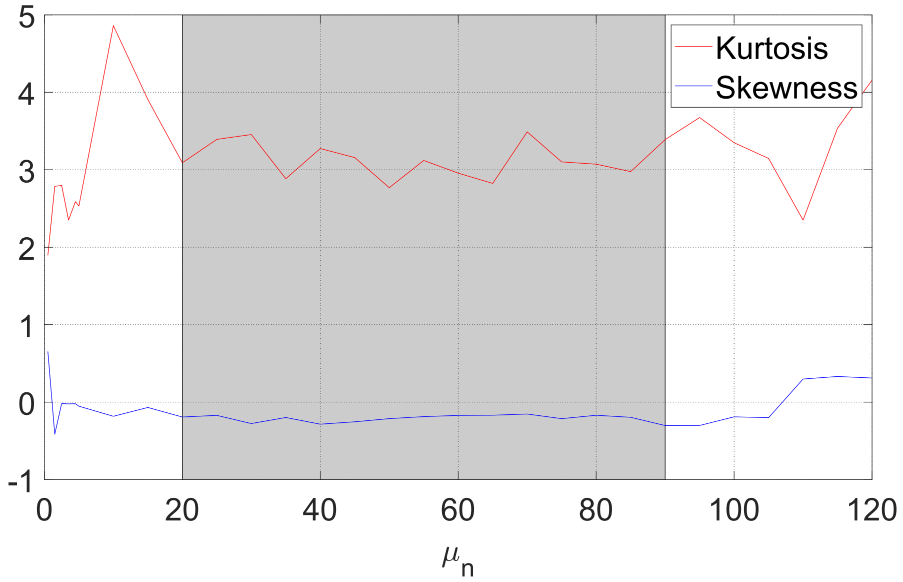


Figure 5.5: Kurtosis and skewness of the latent space metric on LFW when $\mathbb{P}_n = \mathcal{N}(w,1)$ where $w = [0.5,120]$, and $\mathbb{P}_m = \mathcal{N}(0,1)$. If the means of the two distribution are too far apart the training process becomes unstable, hence it affects the kurtosis and skewness of the imposed distributions.

Testing

BioMetricNet has been developed for 1:1 verification in a face authentication scenario in the current setting, particularly when there is a single image template per subject. BioMetricNet has therefore been validated for 1:1 verification on six popular unconstrained face datasets, with the exceptions of large-scale data sets such as MegaFace [126] and the IJB [127] for set-based face recognition, i.e., deciding whether two *sets* of images of a face belong to the same person or not.

The most commonly used data sets for unconstrained face verification on images and videos are Labeled Faces in the Wild (LFW) [128] and The YouTube Faces (YTF) [129]. LFW is composed of 13233 images of 5749 people. The YTF is comprised of 3425 videos of 1595 people. The latest deep learning models for face verification are strong enough to achieve nearly perfect accuracy on LFW and YTF, making the associated results not very informative. We further test BioMetricNet for comprehensive insights on more challenging data sets, including Cross-Age LFW (CALFW) [130], which is constructed by selecting 3000 positive facial pairs with an age gap from LFW, which is intended to add the aging process in intra-class variance and Cross-Pose LFW (CPLFW) [131] designed from 3000 LFW face pairs with pose difference to add intraclass variance pose variance. Finally, we test the method on Celebrity data set for a front and profile view (CFP) [132] with 500 identities with 7000 images and age database (AgeDB) [133] with 16488 images with 568 images.

Dataset	$d = 128$	$d = 256$	$d = 512$	$d = 1024$	$p = 1$	$p = 3$	$p = 8$	$p = 16$
LFW	99.47	99.51	**99.80**	99.63	**99.80**	99.75	99.74	99.72
YTF	97.57	97.76	**98.06**	98.0	**98.06**	97.85	97.73	97.76
CALFW	96.48	96.59	**97.07**	96.78	**97.07**	97.02	96.92	96.93
CPLFW	94.89	94.81	**95.60**	95.25	**95.60**	95.57	95.13	95.43
CFP-FP	99.01	99.08	**99.35**	99.25	99.35	99.33	99.33	**99.47**

Table 5.1: Accuracy (%) for different feature vector d and latent vector p dimensionality. Highest accuracy is obtained for the feature vector of size $d = 512$ and for $p = 1$

We report results for all datasets for 6000 pairs of test images and videos with 3000 matching and 3000 non-matching pairs. We follow the standard *unrestricted with labeled outside data* protocol for reporting the results, as done in [65, 112, 113].

5.3.4 Effect of Feature Vector Dimensionality

We explored the effect of different dimensionality of the feature vector by fixing $p = 1$, and varying d, see Tab. 5.1. It can be observed that small values of d are not sufficient to capture the most discriminative facial features. On the other hand, a too large feature space (1024) causes overfitting and thus a performance drop. We picked the best value, i.e. $d = 512$, since in our experiments as this choice leads to the highest accuracy.

5.3.5 Effect of Latent Space Dimensionality

We have tested various dimensionalities by fixing $d = 512$ in order to choose the optimal latent space size. The results can be seen in Tab. 5.1. In this case, an increase in p leads to a drop in performance as general behavior.

Because p affects the number of parameters at the very bottom of MetricNet (an FC network), its choice greatly influences overall performance. We assume that large p values can be advantageous to a very complex dataset with a typically large amount of training data. Indeed, samples are linearly more separable in higher dimensional latent space. This is even more important when there are very large numbers of data points. On the other hand, too large values of p can lead to a drop in performance because mapping on a large latent space becomes difficult to learn. Tab. 5.1 shows that $p = 1$ is sufficient for most datasets. The CFP-FP, on the other hand, shows that the highest precision is achieved for p =16 (even if relatively small amounts). Perhaps a higher latent space dimensionality, in this case, provides room for better separation. Since $p = 1$ gives optimal or near-optimal outcomes in all instances, we choose this value for the experiments.

Method	**# Image**	**LFW**	**YTF**	**CALFW**	**CPLFW**	**CFP-FP**	AgeDB
SphereFace [111]	0.5M	99.42	95.0	90.30	81.40	94.38	91.70
SphereFace+ [134]	0.5M	99.47	-	-	-	-	-
FaceNet [65]	200M	99.63	95.10	-	-	-	89.98
VGGFace [106]	2.6M	98.95	97.30	90.57	84.00	-	-
DeepID [107]	0.2M	99.47	93.20	-	-	-	-
ArcFace [112]	5.8M	**99.82**	98.02	95.45	92.08	98.37	95.15
CenterLoss [135]	0.7M	99.28	94.9	85.48	77.48	-	-
DeepFace [64]	4.4M	97.35	91.4	-	-	-	-
Baidu [136]	1.3M	99.13	-	-	-	-	-
RangeLoss [137]	5M	99.52	93.7	-	-	-	-
MarginalLoss [138]	3.8M	99.48	95.98	-	-	-	-
CosFace [113]	5M	99.73	97.6	-	-	95.44	-
BioMetricNet	3.8M	**99.80**	**98.06**	**97.07**	**95.60**	**99.35**	**96.12**

Table 5.2: Verification accuracy % of different methods on LFW, YTF, CALFW, CPLFW, CFP-FP and AgeDB. BioMetricNet achieves state-of-the-art results for YTF, CALFW, CPLFW, CFP-FP, and AgeDB and obtains similar accuracy to the state-of-the-art for LFW

5.3.6 Parameters of Target Distributions

5.3.7 Performance Comparison

Tab 5.2 reports the maximum verification accuracy obtained on several datasets through various methods. For YTF and LFW, it can be observed that BioMetricNet achieves higher accuracy than other approaches as stated in the Tab 5.2. The accuracy for YTF and LFW datasets, in particular, is 98.06% and 99.80%, respectively. ArcFace achieves a similar accuracy on these two datasets.

We further test BioMetricNet on more challenging datasets, such as CALFW, CPLFW, CFP-FP, and AgeDB, for a more in-depth comparison. State-of-the-art results for these datasets are far from the "near-perfect" accuracy that we observed previously. As is noted in Tab. 5.2, BioMetricNet outperforms the baseline methods (CosFace, ArcFace, and SphereFace) significantly. For CPLFW, the accuracy of BioMetricNet is 95.60%, resulting in a 3.52% lower error rate than previous state-of-the-art results outperforming ArcFace by a significant margin, as Indicated in Tab 5.2. For CALFW, a 97.07% accuracy is achieved by BioMetricNet, which is 1.62% lower than previous state-of-the-art results. For the CFP dataset, 99.35% accuracy is achieved by BioMetricNet and ArcFace's error rate is reduced by around 1%. Finally, BioMetricNet achieves 96.12%, decreasing the error rate for AgeDB in comparison to ArcFace by approximately 1%.

To sum up, BioMetricNet consistently achieved greater accuracy in comparison with state-of-the-art approaches, showing that the network's discrimination ability is improved by learning the metric in a regularized space to compare face features. This becomes more evident for challenging datasets where the distance from perfect accuracy is higher.

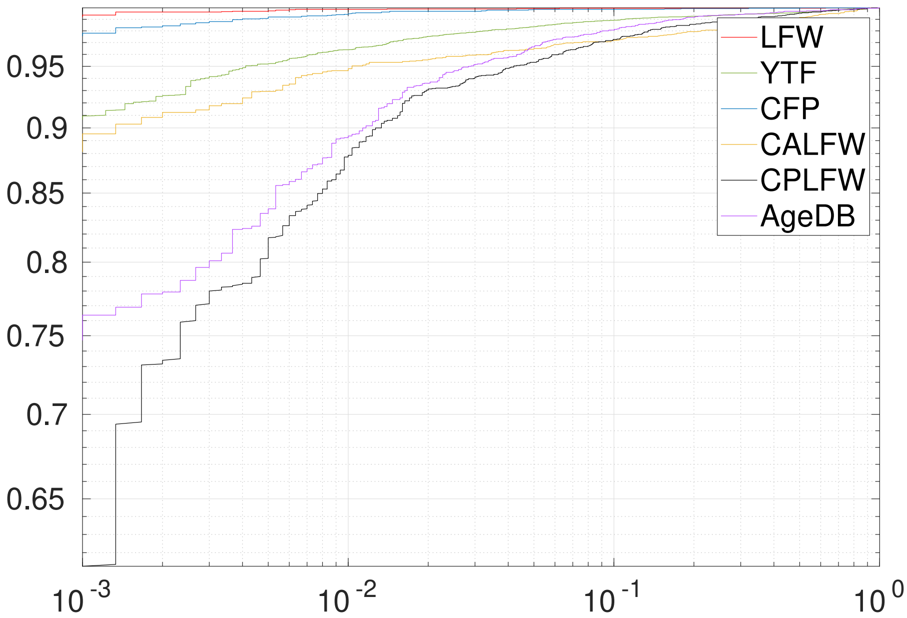


Figure 5.6: ROC curve of BioMetricNet on LFW, YTF, CFP, CALFW, CPLFW and AgeDB.

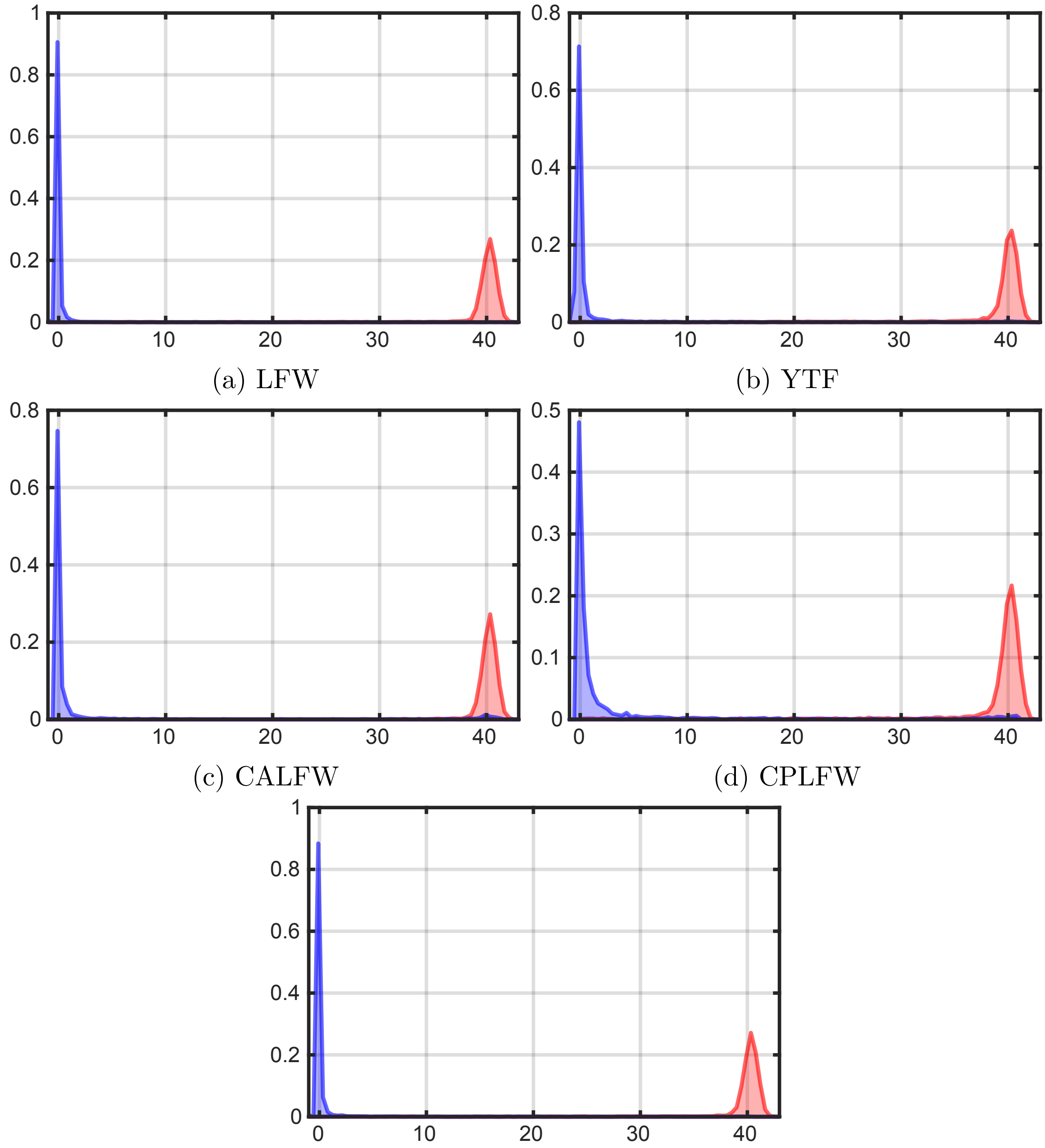
Dataset	GAR@$\mathbf{10^{-2}}$FAR%	GAR@$\mathbf{10^{-3}}$FAR%
LFW	99.87	99.20
YTF	96.93	90.87
CALFW	94.63	88.13
CPLFW	87.73	61.27
CFP-FP	99.43	97.57
AgeDB-30	89.23	74.70

Table 5.3: GAR obtained for LFW, YTF, CFP, CALFW, CPLFW and AgeDB at FAR=$\{10^{-2}, 10^{-3}\}$

5.3.8 ROC Analysis

An overview of the ROC is illustrated in the Fig. 5.6. This curve shows the GAR, the relative number of matching pairs accepted correctly as a function of FAR, which is the relative number of incorrectly accepted non-matching pairs. In addition, in Tab 5.3 we are documenting GAR at various FAR values, namely FAR=$\{10^{-2}, 10^{-3}\}$. With the ROC, we can examine how BioMetricNet's verification task is generalized across various datasets.

It is immediately apparent that since the region between matching and non-matching distributions is clearly separate and barely contaminated, high GARs are obtained in low FAR settings. This is generally true at different "complexity" levels as exposed by the considered datasets. In more details, for LFW at FAR=10^{-2} and



(a) LFW

(b) YTF

(c) CALFW

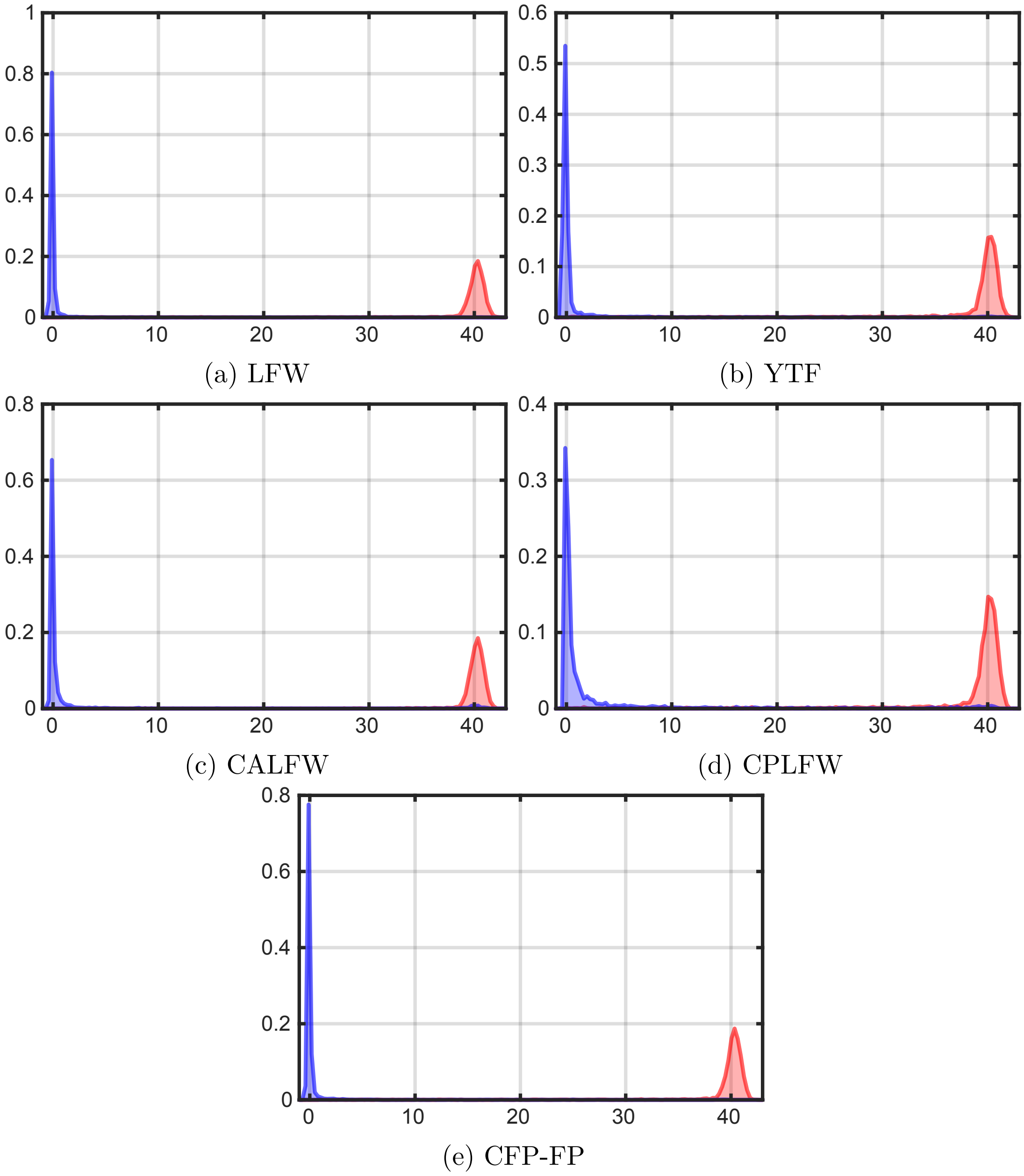
(d) CPLFW

(e) CFP-FP

Figure 5.7: Histogram of **z** decision statistics of BioMetricNet matching and non-matching pairs from (a) LFW; (b) YTF; (c) CALFW; (d) CPLFW; (e) CFP-FP. Blue area indicates matching pairs while red indicates non-matching pairs.

FAR=10^{-3} high GARs of 99.87% and 99.20% are obtained, see Tab 5.3. GARs of 96.93% and 90.87% was obtained for YTF at FAR =10^{-2} and FAR = 10^{-3}.

For the CFP, the same behavior can be observed. It can be noted that the obtained ROC curves are comparatively lower for the challenging datasets CALFW,



(a) LFW

(b) YTF

(c) CALFW

(d) CPLFW

(e) CFP-FP

Figure 5.8: Histogram of $\bar{\mathbf{z}}$ decision statistics of BioMetricNet matching and non-matching pairs from (a) LFW; (b) YTF; (c) CALFW; (d) CPLFW; (e) CFP-FP. Blue area indicates matching pairs while red indicates non-matching pairs.

CPLFW, and AgeDB than LFW, YTF, and CFP. GARs at FAR=10^{-2} and FAR=10^{-3} are 94.63% and 88.13% respectively for CALFW, 87.73% and 61.27% respectively for CPLFW and 89.23% and 74.70% respectively for AgeDB.

5.3.9 Analysis of Metrics Distribution

BioMetricNet closely maps the matching and non-matching pair to the imposed target Gaussian distributions. To further evaluate the effects of latent space regularization, we present the $\mathbf{z}$ and $\bar{\mathbf{z}}$ histograms computed over different test datasets in Fig. 5.7 and Fig. 5.8 respectively. At first, it can be seen that for both $\mathbf{z}$ and $\bar{\mathbf{z}}$ the proposed regularization can shape the latent space as intended by providing Gaussian-shaped distributions. Observing the $\mathbf{z}$ and $\bar{\mathbf{z}}$ histograms, it can be noted that for all datasets, BioMetricNet very effectively separates matching and non-matching pairs.

In the case of non-matching pairs, the distributions of $\mathbf{z}$ are Gaussian with the parameters chosen. For matching pairs, it should be noted that the $\mathbf{z}$ score has the right mean but tends to have a lower variance than the target distribution. A potential explanation is that matching and non-matching pairs exhibit different variability, so it is difficult to match them to the same variance distributions. Indeed, for a fixed number of persons, the number of possible non-matching pairs is much greater than the number of possible matching pairs. Moreover, the KL divergence is not symmetrical, and the chosen loss tends to favor the distribution of the sample with a smaller variance than the target one, rather than a larger variance. Thus a solution where matching pairs have a smaller variance than the target distribution is favored with respect to a solution where non-matching pairs have a greater variance than the target distribution. We can also note that for more difficult datasets, such as CALFW and CPLFW, the distribution obtained for matching pairs has heavier tails than the target distribution.

The $\bar{\mathbf{z}}$ score histogram in Fig 5.8 shows that the variance of both matching and non-matching pairs is slightly lower than that of $\mathbf{z}$. Since reduced variance means improved accuracy of verification, this justifies using $\bar{\mathbf{z}}$ over $\mathbf{z}$. Besides, the decision boundary that we use depends only on the mean values that are retained and are thus not affected by a small decrease in variance.

5.4 Conclusions

We have presented a novel and innovative approach for unconstrained face verification mapping learned discriminative facial features onto a regularized metric space, in which matching and non-matching pairs follow specific and well-behaved distributions. The proposed solution, which does not impose a specific metric, but allows the network to learn the best metric given the target distributions, leads to improved accuracy compared to the state-of-the-art. In BioMetricNet distances between input pairs behave more regularly, and instead of learning a complex partition of the input space, we learn a complex metric over it which further enables the use of much simpler boundaries in the decision phase. With extensive experiments, on multiple datasets with several state-of-the-art benchmark methods, we showed that BioMetricNet consistently outperforms other existing techniques. Future work will consider BioMetricNet in the context of 3D face verification and adversarial attacks. Moreover, considering the slight mismatch between metric distributions and target distributions, it is worth investigating if alternative parameter choices for the target distributions can lead to improved results.

Chapter 6

Beyond cross-entropy: learning highly separable feature distributions for robust and accurate classification

In recent years, deep neural networks have achieved accuracy comparable to or even greater than that of humans in visual tasks such as [4], write the task in text [6], and [7]. They also demonstrated excellent success with learning complex mappings [9] and handling challenging classification tasks [5]. But as their integration in contemporary society expands, the behavior of deep neural networks is becoming ever more subject to the action of malicious *adversaries*.

Despite the success of deep neural networks, many barriers still hinder their use in areas where safety is important, such as autonomous driving systems and medical diagnostics [12, 15]. Adversarial perturbations, a set of techniques used to tamper with the neural network's inputs, pose a major threat. The modifications are mostly invisible to the human eye, but they can still interrupt algorithm activity and cause unpredictable, undesirable outputs. Malicious attackers might exploit such flaws to cause device malfunctions, and the attack would be tough to detect.

While various countermeasures have been proposed, an efficient defense mechanism against a large range of adversarial perturbation is not yet available. In particular, the drawback of deep learning techniques is that the learned decision boundaries in the feature space are highly complex and non-linear [53]. Works addressing this particular issue [53, 54] concluded that most data points are gathered near to decision boundaries and could have a substantial effect on the robustness of the classifier against perturbations. Recent techniques to address this problem can be found in [139] and [140], where logit regularization and curvature regularization methods are used as adversarial defenses, and also in [141], and [142], where theoretical insights on the impact of the use of unlabeled data and noise injection are

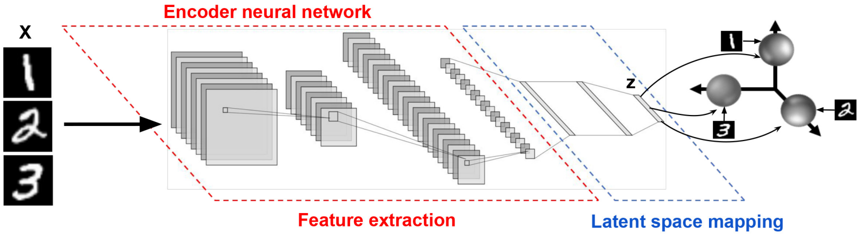


Figure 6.1: The GCCS architecture takes input data and learns discriminative features that are mapped onto Gaussian target distributions in the latent space.

given. At the same time, new strategies are also being developed to create more effective adversarial attacks [21]. In this work, adversarial training is not considered as it entails the cost of producing and training on a large amount of additional input samples; furthermore, adversarial training usually offers robustness against a particular attack, whereas we are interested in tackling the problem of robustness with a more general approach.

In order to improve the robustness of the classifier in the presence of adversarial perturbation, we propose a novel classifier design that goes beyond the cross-entropy loss function. The proposed approach uses a novel objective function that enables the learning of features that maximize inter-class separation and decision variables with well-defined and straightforward distributions linearly separable in latent space. The proposed objective function provides state-of-the-art classification accuracy while at the same time ensuring robustness against adversarial attacks. In order to correctly evaluate the robustness against adversarial examples, we follow the methodological foundations established in [143] and [19].

The resulting classifier uses simple threshold-based decisions in the regularized latent space. This design has several advantages: on the one hand, the accuracy is usually improved with respect to cross-entropy, even in the case of no attacks. On the other hand, such a classifier exhibits remarkably improved robustness against adversarial attacks; indeed, due to the uniformity of the distribution of features in the latent space and the lack of a short path to the adjacent decision region, the attack intensity must be much larger in order to generate misclassification. Finally, the proposed approach can be easily applied to an existing pre-trained cross-entropy based classifier by continuing the training of features and the classification process using our proposed loss function.

In this chapter, we provide a detailed assessment and analysis of the proposed Gaussian Cross Conditional Simplex (GCCS) loss based method in a variety of image classification problems, providing accurate results on well-known datasets such as MNIST [144], FMNIST [145], SVHN [146], as well as more challenging datasets such as CIFAR10, and CIFAR100 [147]. In particular, we have shown that our loss function is inherently more robust than cross-entropy. We support our argument by adopting the state-of-the-art robustness evaluation frameworks [143]. We validate our approach by comparing it to state-of-the-art adversarial robustness techniques and prove that GCCS outperforms these methods under both untargeted (PGD [148]) and targeted attacks (JSMA [149], TGSM [150]).

6.1 Related works

Adversarial robustness is a measure of a network's resilience towards adversarial inputs produced by slightly perturbing inputs in a way that allows them to be misclassified by the network [151].

The concept of adversarial perturbation was first introduced for spam email detection [152, 153]. In the following years, Szegedy et al. [151] showed how neural networks could easily be misclassified if they were fed with specifically altered inputs produced based on the loss function gradient with respect to inputs. In works such as [16, 17, 18], adversarial samples are used as a specific form of data augmentation during the training phase to improve robustness. However, such adversarial training does not prevent adversaries from effectively tampering with the final classification stage [19]. Rather, it has been shown that universal adversarial perturbations can be crafted to induce incorrect classification with high probability independently of the dataset used [20], and also to generalize well over various network structures [150, 154, 155]. Recent theoretical studies have also shown that the robustness of adversarial attacks for classification problems is bound by limits that cannot be avoided by any classifier because they rely on the datasets used, the strength of the attack, and how the perturbations are measured [156].

The authors in [157] investigated how the effectiveness of adversarial attacks could be transferred to models other than the targeted one and demonstrated that adversarial examples that are created to fool a specific model are likely to have an impact on all models that are trained in the same dataset. [150] concluded that adversarial-generated images are misclassified even when printed on paper and digitally re-acquired, thereby showing that the phenomenon is true in both the digital and the physical domains. Furthermore, [158] has shown that deep facial recognition learning methods can incorrectly identify faces when users wear ad-hoc adversarial glasses. Finally, [159] described the method for generating image patches to be placed on the input target image to cause the neural network to output the desired class. This type of attack is constructed and carried out without the awareness of the target image and theoretically enables the adversarial patch to be widely used for malicious intent after it is spread over the Internet.

Several papers have studied defense techniques against attacks. The authors of [157] suggest an input gradient regularization approach used during the training process to push the model to have smooth gradients. They believe that a gradient-trained model with less extreme values is more resistant to adversarial perturbations and that its behavior in response to such attacks is also easier to interpret. In addition, [160] calculates an instance-specific lower bound on the input perturbation level required to alter the classifier's decision, providing a formal characterization of its robustness. The article also introduces the Cross-Lipschitz regularization function, which forces the classifier's differences to be constant at the data points. Instead, Jakubovitz et al. [161] propose a low-complexity regularization technique that uses the Jacobian Frobenius norm of the network, which is applied to already trained models as post-processing, robustness-improving stage. In particular, while not being an active defense method, the proposed GCCS method ensures improved robustness against adversarial perturbations, as it is.

If standard approaches focus on learning the classification boundaries, the proposed GCCS approach instead learns to map the input classes to the latent space's target distributions. Specifically, the encoder maps each class's features to the Gaussian distributions on a simplex for an arbitrary number of classes, maximizing classes' separability. Other papers also suggest learning how to map to a regularized space, such as [162] and [163], which incorporate adversarial and variational autoencoder techniques. In [164], Stuhlsatz et al. present a feature extraction method that generalizes the classical Linear Discriminant Analysis (LDA) of neural networks. The authors in [165] nonlinearly extend LDA by placing it on top of a deep neural network and maximizing LDA's values in the last hidden representation.

The primary goal of the most discriminant analysis methods is the dimensionality reduction [166]. One of the drawbacks of these approaches is that they seek to increase the distance between classes that are already well-separated at the cost of poorly-separated neighboring classes, leading to a non-hierarchical pattern of inter-class separability. Another related work is RegNet [167], a deep biometric authentication learning technique that deals with the one-vs-all classification problem of distinguishing authorized users from non-authorized users. This technique regularizes two-dimensional latent space via a loss function based on a simplified Kullback-Leibler divergence equation; however, this method is not suitable for high-dimensional classification problems as addressed by GCCS.

6.2 Proposed Method

The proposed approach is based on the architecture shown in 6.1. Labeled training data $\mathbf{X}$ for a D-class classification problem is given as input to a neural network consisting of a feature extractor and a latent space mapper. The feature extractor aims to learn non-linear transformations from arbitrary data distributions and extract distinctive and highly separable features. The latent space mapper consists of one or more fully connected layers to map the output $\mathbf{z}$ to specific target distributions in the D-dimensional latent space (i.e., as many dimensions as the number of classes); no non-linear activation function is employed in the last layer of the mapper. It is worth noting that the proposed approach does not rely on a specific feature extraction architecture, so existing state-of-the-art architectures

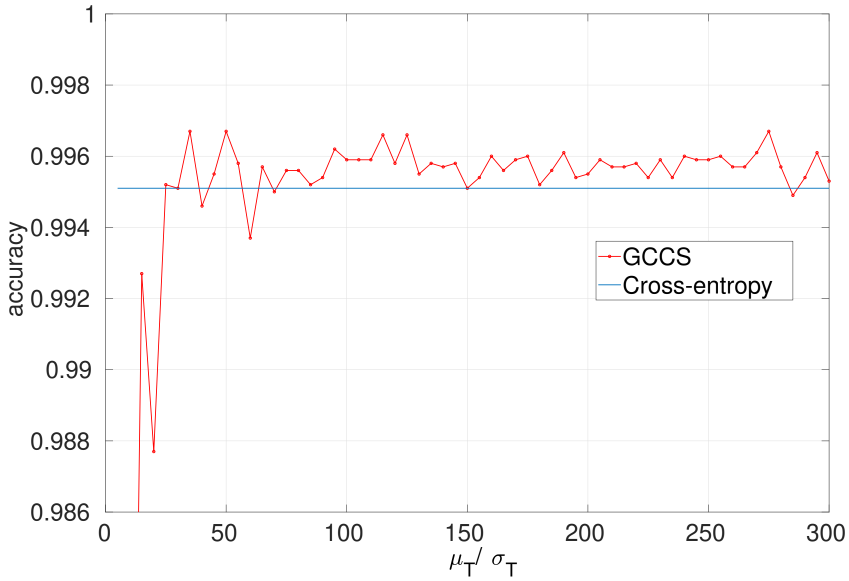


Figure 6.2: Classification accuracy (%) for GCCS and cross-entropy on MNIST with ResNet-18.

can be used for this task.

In order to achieve the desired purpose, the proposed method needs to define three key components: a target model for the distribution of features in a latent space; a loss function to achieve that distribution; and, finally, a decision rule. The specifics are the following.

6.2.1 Model for the target distributions

GCCS aims to learn the most discriminative features and maximize the inter-class separability by finding a nonlinear projection of high-dimensional observations onto a lower-dimensional space. This is obtained by regularizing the latent space to D different statistical distributions, where D is the number of classes the data belongs to. Let us first define the desired target distribution $\mathbb{P}_i$ for class $\mathcal{C}_i$, $i = 1, \dots D$, as a D-variate Gaussian distribution, i.e. $\mathbb{P}_i = \mathcal{N}(\boldsymbol{\mu}_{Ti}, \boldsymbol{\Sigma}_T)$, with $\boldsymbol{\mu}_{Ti} = \mu_T \mathbf{e}_i$ and $\boldsymbol{\Sigma}_T = \sigma_T^2 \mathbb{I}_D$, where $\mathbf{e}_i$ is the ith standard unit vector and $\mathbb{I}_D$ is the $D \times D$ identity matrix. μ_T and σ_T are user-defined parameters that are related to inter-class separation and are discussed later in the manuscript. Here, it should be noted that in order to have separable distributions we should have $\mu_T / \sigma_T > \sqrt{2D}$, otherwise as D grows the classes will inevitably mix.

Because each distribution $\mathbb{P}_i$ has a mean value proportional to $\mathbf{e}_i$, the statistical distributions are based on the vertices of the regular $(D-1)$-simplex at $\mu_T \mathbf{e}_i$, as shown in Fig 6.1. The target model has several advantages. Firstly, this choice ensures that each class is at the same distance from all other classes. Due to the uniformity of the latent space distributions and the consequent lack of a short

Method	MNIST ResNet-18	FMNIST ResNet-18	SVHN ResNet-18	CIFAR-10 ResNet-18	CIFAR-10 Shake-Shake-96	CIFAR-100 Shake-Shake-112
GCCS - regular training	99.58	92.69	94.20	82.97	96.19	76.53
GCCS - fine tuning	**99.64**	**93.83**	**95.58**	**81.52**	**97.06**	**77.48**
No Defense - cross-entropy	99.35	91.91	94.12	78.59	95.78	76.30
Jacobian Reg. - regular training [161]	98.99	91.79	94.11	70.09	-	-
Jacobian Reg. - fine-tuning[161]	98.53	92.43	93.54	82.09	-	-
Input Gradient Reg. - regular training [157]	97.98	88.45	93.77	78.32	96.50	74.89
Input Gradient Reg. - fine-tuning [157]	99.11	92.55	93.17	76.15	96.90	75.68
Cross Lipschitz regular training [160]	96.78	92.54	91.42	80.10	-	-
Cross Lipschitz - fine-tuning [160]	98.77	92.41	93.50	79.39	-	-

Table 6.1: Maximum test accuracy obtained through *regular training* vs *fine-tuning* over different benchmark datasets with different competing techniques in the case in which no adversarial attack is performed.

path, the attack strength must be much greater to generate a misclassification leading to improved robustness. Moreover, provided that the distributions are Gaussian, the decision boundaries are straightforward to compute. This contrasts with the standard behavior of neural networks, which appear to generate very complex boundaries, promoting accuracy and adversarial robustness.

6.2.2 Loss function

To train the network, we need to introduce a loss function that helps us minimize the appropriate distance metric between the distributions of latent output variables and target distributions.

Let us refer to the output of a encoding neural network as $\mathbf{z} = H(\mathbf{x})$, where $[z_1, \dots, z_D] \in \mathbb{R}^D$ indicates latent mapping, and $\mathbf{x} \in \mathbb{R}^n$ indicates input data belonging to D of different classes. The goal is to learn the encoding function of the input $\mathbf{z} = H(\mathbf{x})$ such that $\mathbf{z} \sim \mathbb{P}_i$ if $\mathbf{x} \in \mathcal{C}_i$.

During the training phase, the network inputs a batch of samples $\mathbf{X} \in \mathbb{R}^{b \times n}$, where b is the batch size and computes the encoded outputs $\mathbf{z} \in \mathbb{R}^{b \times D}$. We are interested in their first and second-order statistics, which can be estimated as a sample of $\boldsymbol{\mu}_{Oi}$ and a sample covariance of $\boldsymbol{\Sigma}_{Oi}$ for each class. Considering that the target statistics are known, and the sample statistics for the batch have been computed, we can establish an effective loss to calculate how far the distributions are from each other. More in-depth, we use the *Kullback-Leibler* divergence (KL).

For the sample distribution of any class $\mathcal{C}_i$, the difference between the KL and the Gaussian target distribution can be written as:

$$\mathcal{L}_i = \log \frac{|\boldsymbol{\Sigma}_T|}{|\boldsymbol{\Sigma}_{Oi}|} - D + \operatorname{tr}(\boldsymbol{\Sigma}_T^{-1} \boldsymbol{\Sigma}_{Oi}) + (\boldsymbol{\mu}_{Ti} - \boldsymbol{\mu}_{Oi})^\intercal \boldsymbol{\Sigma}_T^{-1} (\boldsymbol{\mu}_{Ti} - \boldsymbol{\mu}_{Oi}) \tag{6.1}$$

We consider the cumulative loss $\mathcal{L} = \sum_{i=1}^{D} \mathcal{L}_i$. This loss is minimized when the sample statistics of the D encoded distributions match the target distributions.

However in the case of a limited batch size, it may be difficult to control the behavior of the tails of the distributions obtained by relying solely on KL. Therefore, we also consider the $\mathcal{K}_{i,j}$, [168] of the jth component of the ith target distribution, specified as $\mathcal{K}_{i,j} = \left(\frac{z_{i,j}-\boldsymbol{\mu}_{Oi,j}}{\sigma_{Oi,j}}\right)^4$.

In the case of multiple i.i.d. univariate normal distributions, such as those we enforce during training, the target Kurtosis for each class is $\mathcal{K}_{i,j} = 3$. This can be applied to the cumulative loss, obtaining the loss $\mathcal{L}^{\text{GCCS}}$ as follows:

$$\mathcal{L}^{\text{GCCS}} = \sum_{i=1}^{D} \left[\mathcal{L}_i + \lambda(\mathcal{K}_i - 3)\right], \tag{6.2}$$

where $K_i = 1/D \sum_j K_{i,j}$ and λ determines the strength of the Kurtosis term and is set to $\lambda = 0.2$.

6.2.3 Decision Rule

Once the preconditions are fulfilled, GCCS allows defining optimal decision boundaries in the resulting latent space. For the given target distributions, the optimal boundaries are obtained by partitioning the space into Voronoi regions such that all points in a region are closer to the respective centroid (the mean vector $\boldsymbol{\mu}_{Ti}$) than to any other centroid in the $(D-1)$-simplex. The resulting decision rule consists of computing the distance of the feature point from all centers and choose the class with the minimum distance. To determine which class a test image belongs to, the following decision rule is employed:

$$\widehat{y} = \arg\max_i z_i, \tag{6.3}$$

which returns the index of the predicted class for the test image.

6.3 Experiments

6.3.1 Datasets and Training Parameters

The performance of classifiers trained using the GCCS loss was evaluated on MNIST [144], FMNIST [145], SVHN [146], IFAR-10 and IFAR-100 [147]. For less complex datasets such as MNIST, FMNIST, and SVHN, experiments were performed using ResNet-18 [55] as a feature extraction network. Shake-Shake-96 and Shake-Shake-112 [169] regularization networks have been used for for the more challenging CIFAR-10 and CIFAR-100 datasets, respectively, using a widen factor equal to 6 for the former and 7 for the latter. The encoder's last layer is preceded by a fully-connected layer that outputs a vector with dimension D. Each network was trained for a total of 1800 epochs. In order to boost network convergence, we used cosine learning rate decay [170] with an initial value of 0.01 as well as weight decay with a rate of 0.001. Finally, dropout regularization [171] with 0.8 keep probability value was applied to all fully connected layers in the network.

Target Distributions Parameters

In this section, we perform an experiment to explore the behavior of target distributions for different mean and variance values. Since we set the mean μ_T and the variance σ_T for the target distributions so that they are centered on the vertices of the regular $(D-1)$-simplex, the only parameter that affects our design is the ratio μ_T/σ_T, i.e., how far apart the distributions are from the chosen variance.

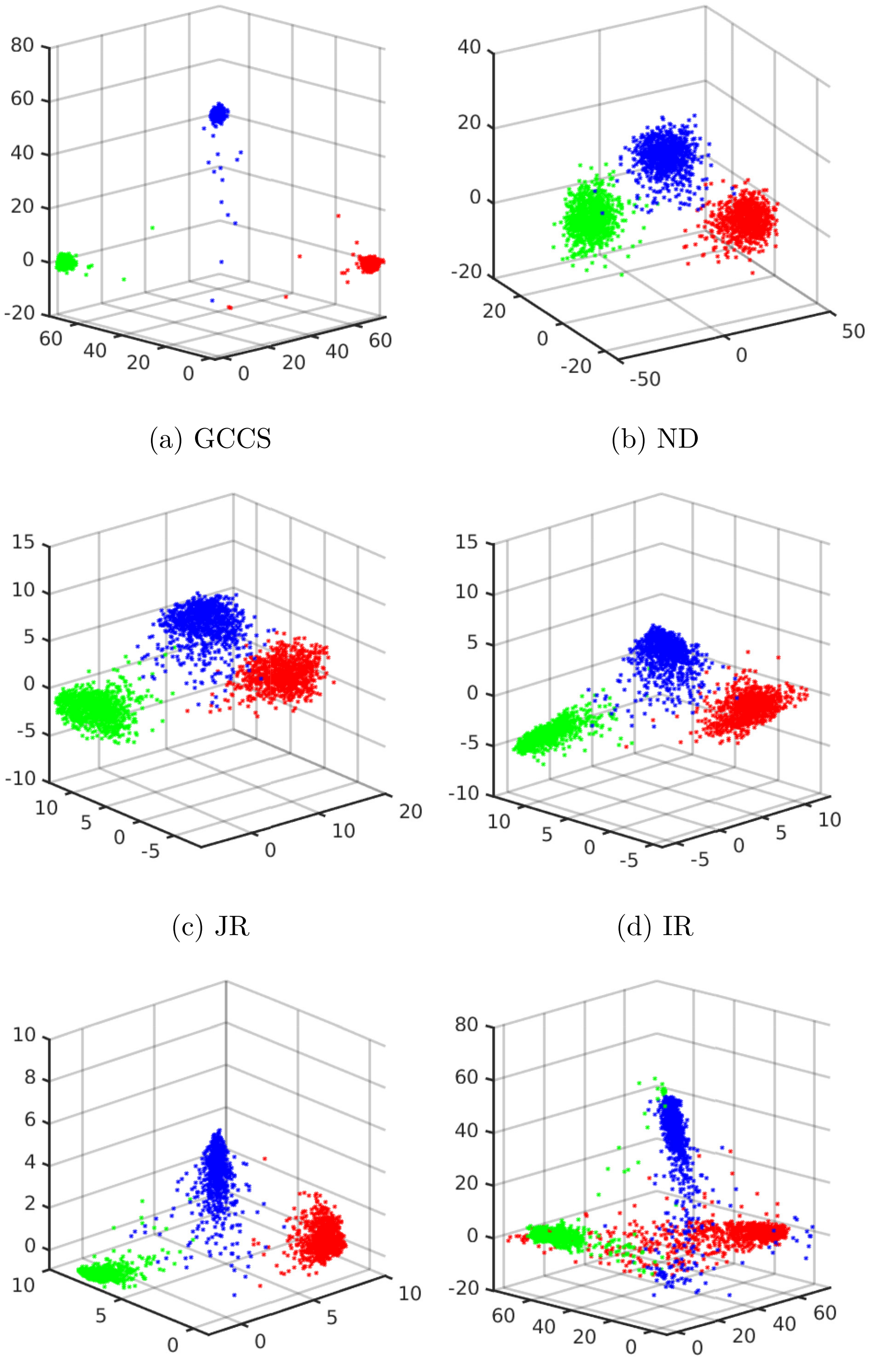
In this experiment, we set $\sigma_T = 1$, so that the target distributions are $\mathbb{P}i = \mathcal{N}(\mu_T \mathbf{e}_i, \mathbb{I}_D)$; then we compute the accuracy of the classification as a function of $\mu_T \in [0.5,300]$. Fig. 6.2 shows the accuracy as μ_T/σ_T for MNIST-10 dataset. It can be observed that the accuracy of the $\mu_T \geq 20$ region is even higher than that achieved with the traditional cross-entropy loss.

In the following, we will pick $\mu_T = 70$ and $\sigma_T = 1$. This choice ensures that we work in that region and that the target distributions are sufficiently distant from each other.

6.3.2 Classification accuracy

As a first experiment, we compared the classification accuracy of GCCS with that obtained by an equivalent network trained with cross-entropy loss (no defense) and state-of-the-art defense techniques such as Jacobian Regularization [161], Input Gradient Regularization [157], and Cross Lipschitz Regularization [160] in the case in which no adversarial attack is performed. As seen in Tab. 6.1, GCCS delivers high classification accuracy both when networks are trained from scratch (*normal training*) and when they are first trained using regular cross-entropy losses and then fine-tuned with either GCCS loss or other protection techniques (*fine-tuning*). In particular, Tab. 6.1 shows that the proposed method outperforms the standard cross-entropy loss and other existing approaches [161], [157] and [160] over the datasets considered. More comprehensive evidence can be drawn from the results; other techniques generally lead to a small decrease in accuracy with respect to the standard cross-entropy loss function, while GCCS improves accuracy, especially in challenging datasets CIFAR-10 and CIFAR-100.

In comparison to other methods, the highest classification accuracy yielded by GCCS is due to the high separability of the target distributions in the latent space. To illustrate this better, the output distribution for three different MNIST classes [0, 1, and 9] is shown in Fig 6.3. Looking at the Fig 6.3-a versus Fig 6.3-b, Fig 6.3-c, Fig 6.3-d, and Fig 6.3-e, the distributions of the output of the three classes are immediately less spread out and more separated than the other cases.



(a) GCCS

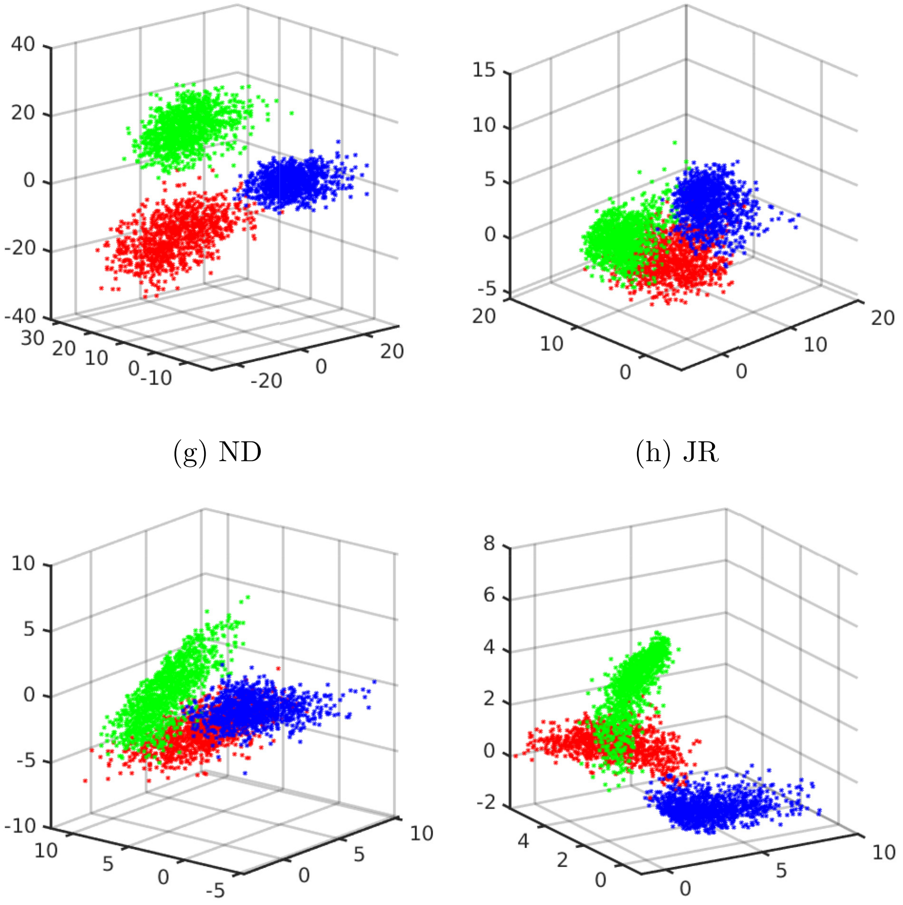
(b) ND

(c) JR

(d) IR

(e) CL

(f) GCCS



(g) ND

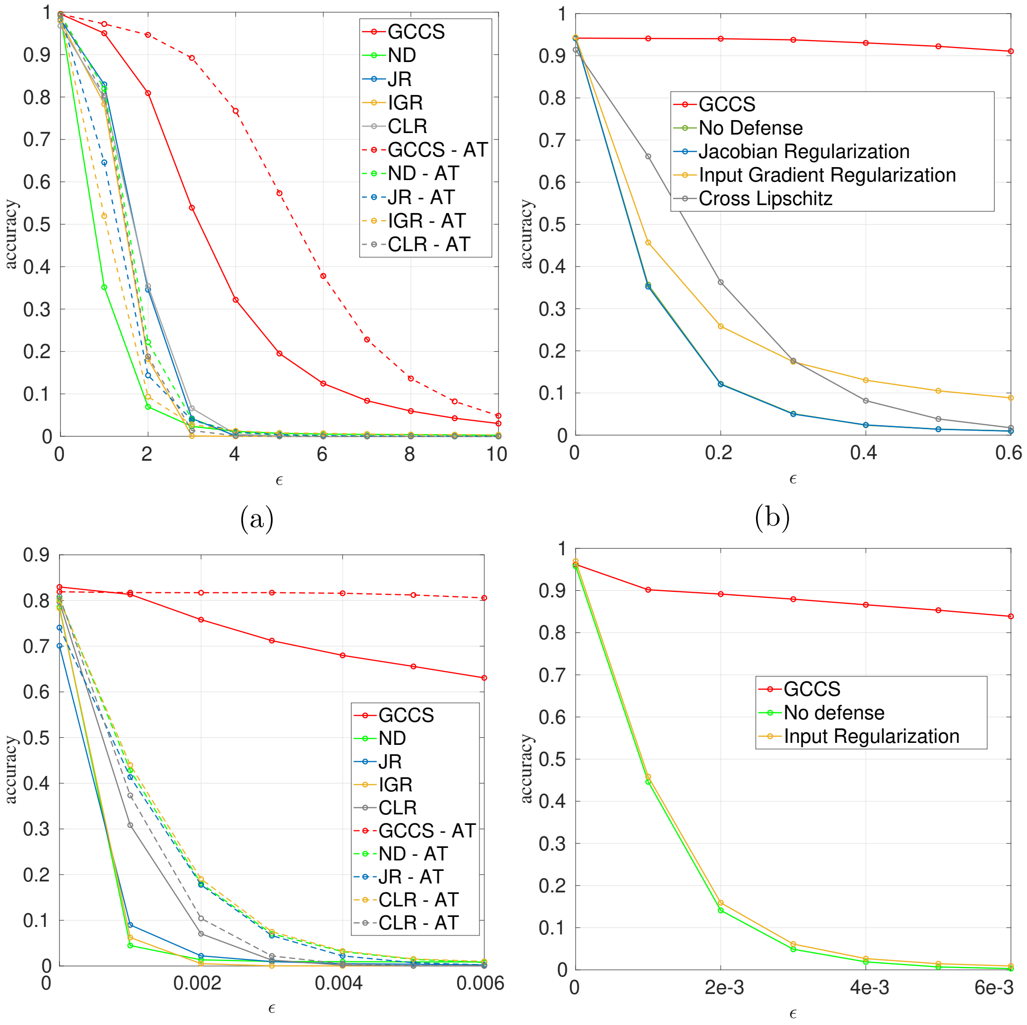
(h) JR

(i) IR

(j) CL

Figure 6.3: **(a-e)** Visual representation of latent space output distributions on MNIST for regular training in the case that no adversarial attack is applied. For better visualization of the separability, only three classes are shown, and an appropriate scale is used for each plot. (a) GCCS; (b) standard cross-entropy; (c) Jacobian Regularization [161]; (d) Input Gradient Regularization [157]; (e) Cross Lipschitz Regularization [160]. **(f-j)** Visual representation of latent space output distributions on MNIST for TGSM (5 steps, $\epsilon = 2e^{-3}$) is applied. For better visualization, only three classes are shown. (f) GCCS; (g) standard cross-entropy; (h) Jacobian Regularization [161]; (i) Input Gradient Regularization [157]; (j) Cross Lipschitz Regularization [160].

Fig 6.3 also demonstrates that GCCS provides lighter distribution tails than other methods.



(a) (b)

(c) (d)

Figure 6.4: Test accuracy for PGD (5 steps) attack on (a) ([MNIST, ResNet-18]); (b) ([SVHN, ResNet-18]); (c) ([CIFAR-10, ResNet-18]); (d) ([CIFAR-10, Shake-Shake-96]) for different values of ϵ.

6.3.3 Robustness Evaluation

This section evaluates how GCCS and other competing techniques degrade under both targeted attacks (TGSM, JSMA) and non-target attacks (PGD) in terms of classification accuracy. The accuracy is assessed as a function of the tunable parameter ϵ, which shows how strong the attack is. The noise vector namely $\mathbf{n}$ is added by the attack to the input signal $\mathbf{x}$ that satisfies $\|\mathbf{n}\|_\infty/\|\mathbf{x}\|_\infty \leq \epsilon$.

Non-targeted Attacks

We begin by evaluating the performance of all methods when subjected to the non-target PGD attack on MNIST, SVHN, CIFAR-10, and CIFAR-100 datasets. Projected Gradient Descent (PGD) [148], is an iterative FGSM variant that introduces noise in multiple steps. PGD is, in particular, the strongest adversarial attack that exploits first-order local information about the trained model. In this work, for PGD, we use 5-iterations attack, i.e., PGD-5 as done in [156, 172, 173].

We set $0 \leq \epsilon \leq 10e^{-2}$ for MNIST, and set $0 \leq \epsilon \leq 6e^{-3}$ for SVHN, CIFAR-10, and CIFAR-100, since the MNIST is a less challenging dataset, in general. As shown in Fig. 6.4, GCCS outperforms all competing approaches by a significant amount on all considered data sets. Our approach is much robust than the others, especially for stronger attacks. The performance gap is especially evident in PGD, which is indeed the strongest adversarial attack utilizing local first-order network information.

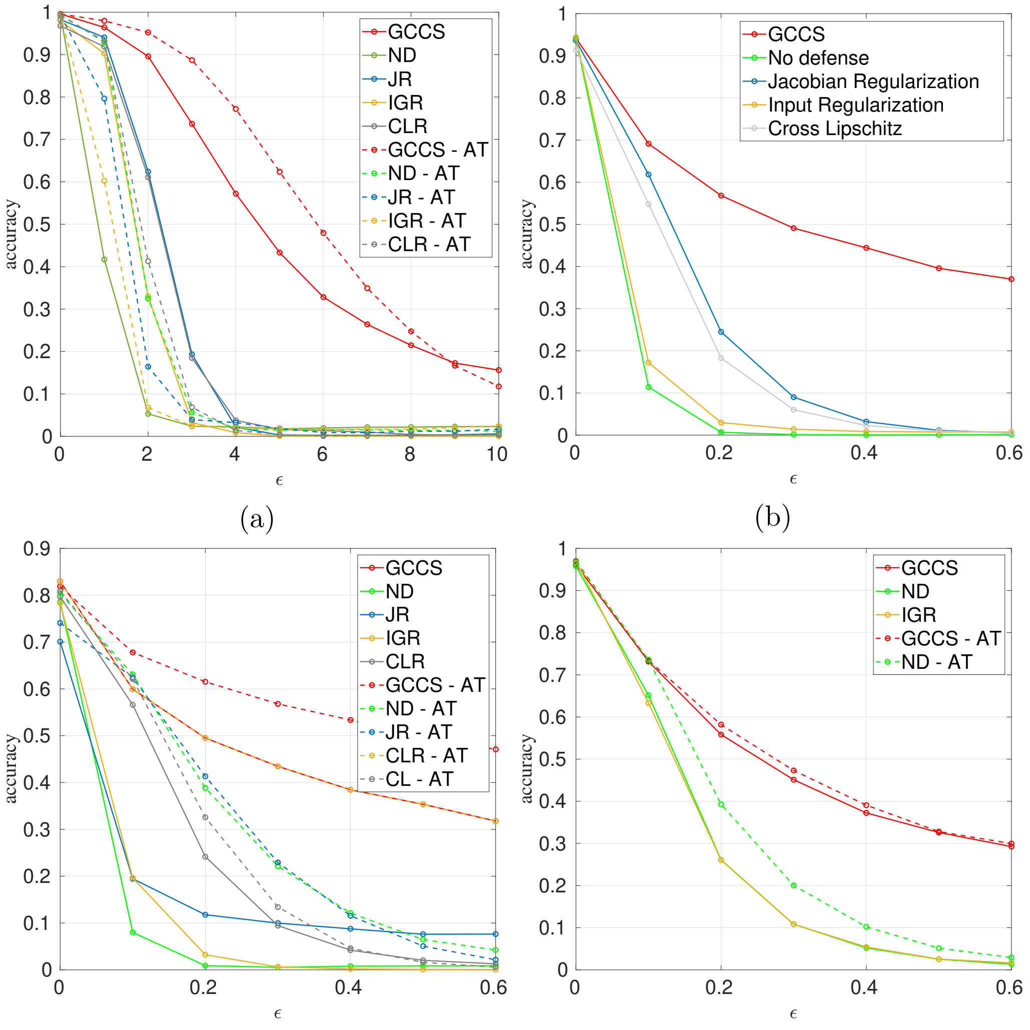
Targeted Attacks

Targeted adversarial attacks such as TGSM and JSMA are also considered. Similar to Sec 6.3.3, we present the classification accuracy curves against the attack strength ϵ.

TGSM Attack: In TGSM [150], the input samples are disrupted by introducing noise in the direction of the negative gradient with respect to the selected target class. Fig. 6.5 presents the results for TGSM-5, 5-iterations of TGSM attacks, over MNIST, SVHN, CIFAR-10, and CIFAR-100 datasets. In this attack, the target output class is y_{l+1} while the true class is y_l.

Fig. 6.5 shows that GCCS yields significantly higher robustness compared to other approaches, across various datasets, and with different attack strength ϵ. In order to gain a better understanding of why the proposed method performs better than the others in Fig. 6.3, a visual representation of the target distributions is illustrated in the latent space *after* TGSM-5 attack $\epsilon = 2e^{-3}$ has been applied.

Fig. 6.3-g clearly shows the effectiveness of the attack when no defense mechanism is being employed in the sense that the output distributions are shifted to replace the output distribution of the next class. Fig. 6.3-h, Fig. 6.3-i, and Fig. 6.3-j report the output distributions under TGSM in the case of Jacobian, Input Gradient, and Cross-Lipschitz regularizations, respectively showing that, despite the defense mechanism, the distributions still appear to shift their position in the latent space towards the adjacent classes, causing a significant drop in classification accuracy as seen in Fig. 6.5. Instead, in the GCCS case (Fig. 6.3-f), even if the tails of the output distributions become heavier, their positions are not swapped with the adjacent classes, allowing for better separability and thus improved classification accuracy and robustness.



(a)

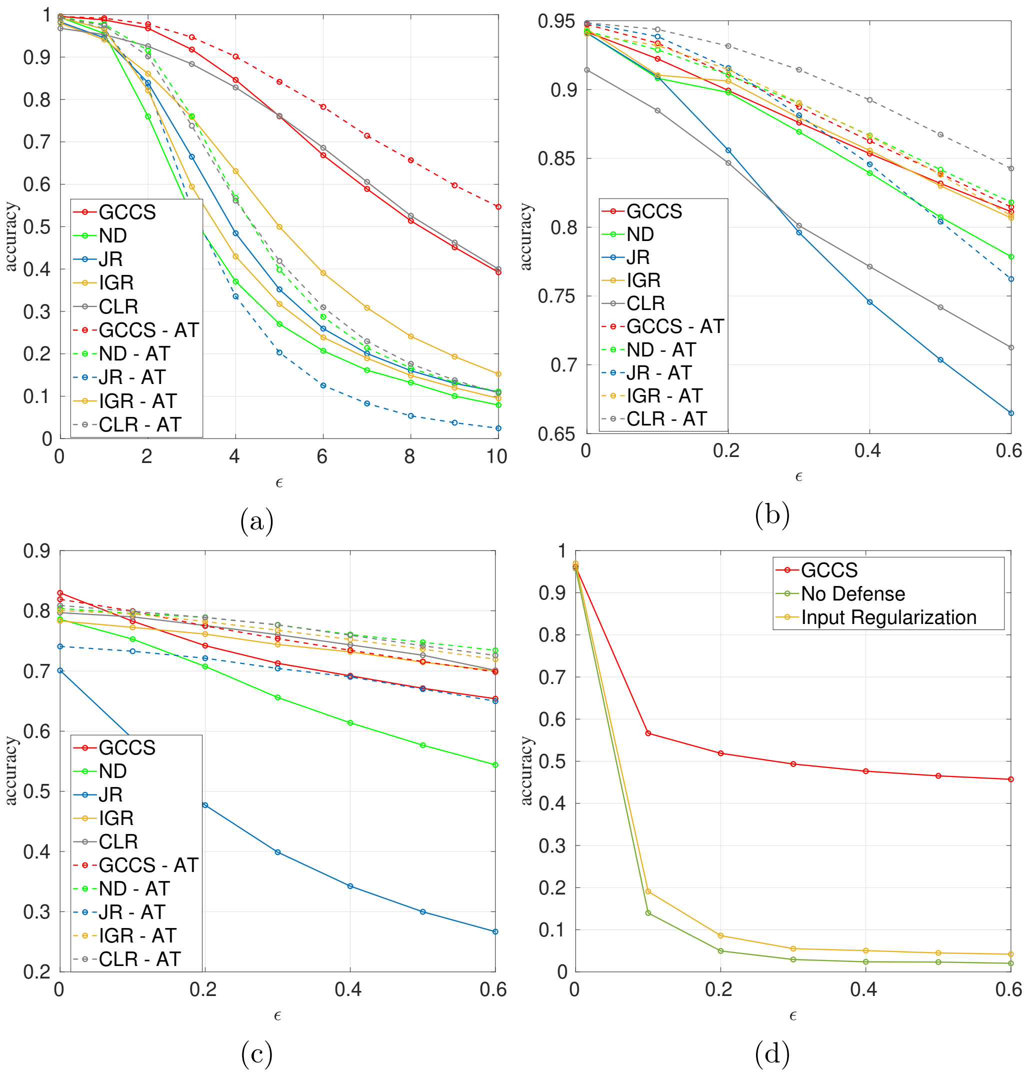
(b)

(c)

(d)

Figure 6.5: Test accuracy when applying the TGSM attack (5 steps) for (a) ([MNIST, ResNet-18]) ; (b) ([SVHN, ResNet-18]); (c) ([CIFAR-10, ResNet-18]) (d) ([CIFAR-10, Shake-Shake-96]), for different values of ϵ.

JSMA Attack: The other target attack we consider is JSMA [149], which consists of iteratively computing the Jacobian matrix of the network function to form a saliency map; this map is used at each iteration to pick which pixels to tamper with such that the probability of changing the output class to the selected one is increased. In our case, we consider the JSMA-200 with the 1-pixel saliency map. Similar to the TGSM case, Fig. 6.6 indicates the classification accuracy for increasing attack strength ϵ. The proposed method confirms its robustness even in the case of the JSMA attack, achieving higher accuracy than other methods, especially on the challenging CIFAR-10 dataset.



(a)

(b)

(c)

(d)

Figure 6.6: Test accuracy when applying the JSMA attack (200 steps, 1 pixel) on (a) ([MNIST, ResNet-18]); (b) ([SVHN, ResNet-18]); (c) ([CIFAR-10, ResNet-18]); (d) ([CIFAR-10, Shake-Shake-96]), for different values of ϵ.

6.4 Conclusions

We have presented an approach that goes beyond cross-entropy, using a loss function that promotes class separation and robustness by learning a mapping of the decision variables onto Gaussian distributions. Our work was inspired by the idea that mapping the centroids of the distributions on the vertices of the simplex could lead to the uniformity of the distributions of the features in the latent space and the lack of a short path to the adjacent decision-region. Experiments on different multi-class datasets show excellent performance of the GCCS-trained classifiers, both in terms of accuracy and robustness of the adversarial attack classifier, outperforming existing state-of-the-art approaches, both when used to train the network from scratch and when used as a fine-tuning step on pre-trained networks. Performance is evaluated for both targeted and non-target adversarial attacks. We have shown that regularizing latent space on target distributions significantly improves robustness against adversarial perturbations. Indeed an analysis of the latent distributions for the proposed GCCS method shows that the different classes tend to remain separate even in the presence of targeted attacks, although comparable attack strength inevitably mixes the distributions achieved by competing methods.

Chapter 7

Conclusion: summary and future work

7.1 Summary

The conclusions of the dissertation are summarized as follows:

- The dissertation consists of two parts; the first part aims to improve biometric authentication techniques' robustness under challenging and limited data environments. In particular, we presented novel methods for generic biometric authentication under the constraint of limited data and unconstrained facial verification. The second part of the dissertation focuses on a robust and accurate classification system using deep learning.

- Firstly, we presented a novel approach for generic biometric authentication based on either adversarial learning or statistical techniques. We show that the latent space regularization leads to improved accuracy and robustness. Our intuition behind this behavior is that the non-linear boundaries learned by standard deep learning classifiers indeed become very complex as they try to closely fit the training data, leaving room for misclassification. Conversely, the proposed methods enable much simpler boundaries to be used as it does not learn how to partition the space but rather how to map the input space into the latent space. With extensive experimentation on multiple large biometric datasets with several state-of-the-art benchmark methods, we showed that the proposed approach consistently outperforms other existing techniques. We further show that regularizing the latent space makes the architecture less vulnerable to targeted and nontargeted perturbations.

- Furthermore, we advanced the state-of-the-art in face verification by learning a latent representation in which matching and non-matching pairs are mapped onto clearly separated and well-behaved target distributions. The proposed solution, which does not impose a specific metric, but allows the network to learn the best metric given the target distributions, leads to improved accuracy compared to state of the art. In BioMetricNet, distances between input pairs behave more regularly. Instead of learning a complex partition of the input space, we learn a complex metric over it, which further enables the use of much simpler boundaries in the decision phase. With extensive experiments on multiple datasets with several state-of-the-art benchmark methods, we showed that BioMetricNet consistently outperforms other existing techniques.

- In the second part of the dissertation, we have presented an approach that goes beyond cross-entropy, employing a loss function that promotes class separability and robustness by learning a mapping of the decision variables onto Gaussian distributions. Our work was motivated by the idea that mapping the centroids of the distributions on the vertices of a simplex could lead to the uniformity of the feature distributions in the latent space and the lack of a short path towards a neighboring decision region. Experiments on different multi-class datasets show excellent performance of the classifiers trained using the GCCS loss both in terms of accuracy and robustness of the classifier against adversarial attacks, outperforming existing state-of-the-art methods, both when used to train a network from scratch and when applied as a fine-tuning step on pre-trained networks.

7.2 Future directions

Though contributions were offered across a range of challenges in biometric authentication and robust and accurate classification, the studies presented in this dissertation may lead into many new research challenges. Several problems related to the topics discussed in this thesis are still open and worth investigating. Based on the contributions of this thesis, the following research directions appear promising.

- Concerning the proposed methods for generic biometric authentication, i.e., AuthNet and RegNet adding new users to pre-trained networks and handling user revocation are still open.

- Regarding BioMetricNet, future work will consider BioMetricNet in the context of 3D face recognition and adversarial attacks. Moreover, considering the slight mismatch between metric distributions and target distributions, it is worth investigating if alternative parameter choices for the target distributions can lead to improved results.

Chapter 8

Glossary

- CNN: Convolutional Neural Network
- DNN: Deep Neural Network
- EER: Equal Error Rate
- FAR: False Acceptance Rate
- FRR: False Rejection Rate
- FF: Feed Forward
- GAN: Generative Adversarial Network
- GCCS: Gaussian Class Conditional Simplex
- JS divergence: Jensen-Shannon divergence
- KL divergence: Kullback-Leibler divergence
- MLP: Multi-Layer Perceptron
- NN: Neural Networks
- ROC: Receiver Operating Characteristics

Bibliography

[1] *Importance of user authentication in network security.* https://www.seqrite.com/blog/importance-of-user-authentication-in-network-security. Accessed: 2020-12-30.

[2] *Top ten mind blowing advantages of biometric technology.* https://www.m2sys.com/blog/biometric-hardware/top-ten-mind-blowing-advantages-of-biometric-technology/. Accessed: 2020-12-30.

[3] Brendan F Klare et al. "Face recognition performance: Role of demographic information". In: *IEEE Transactions on Information Forensics and Security* 7.6 (2012), pp. 1789–1801.

[4] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. "Multi-column deep neural networks for image classification". In: *arXiv preprint arXiv:1202.2745* (2012).

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems.* 2012, pp. 1097–1105.

[6] Li Wan et al. "Regularization of neural networks using dropconnect". In: *International conference on machine learning.* 2013, pp. 1058–1066.

[7] Yi Sun et al. "Deep learning face representation by joint identification-verification". In: *Advances in neural information processing systems.* 2014, pp. 1988–1996.

[8] Jun-Yan Zhu et al. "Toward multimodal image-to-image translation". In: *Advances in neural information processing systems.* 2017, pp. 465–476.

[9] Amjad Almahairi et al. "Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired Data". In: *arXiv preprint arXiv:1802.10151* (2018).

[10] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. "Image-to-image translation for cross-domain disentanglement". In: *Advances in neural information processing systems.* 2018, pp. 1287–1298.

[11] Michal Uricar et al. "Yes, we gan: Applying adversarial techniques for autonomous driving". In: *Electronic Imaging* 2019.15 (2019), pp. 48–1.

[12] Kevin Eykholt et al. “Robust physical-world attacks on deep learning visual classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018, pp. 1625–1634.

[13] Veit Sandfort et al. “Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks”. In: *Scientific reports* 9.1 (2019), pp. 1–9.

[14] Agisilaos Chartsias et al. “Disentangled representation learning in cardiac image analysis”. In: *Medical image analysis* 58 (2019), p. 101535.

[15] Samuel G Finlayson et al. “Adversarial attacks on medical machine learning”. In: *Science* 363.6433 (2019), pp. 1287–1289.

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”. In: *arXiv preprint arXiv:1412.6572* (2014).

[17] Ruitong Huang et al. “Learning with a strong adversary”. In: *arXiv preprint arXiv:1511.03034* (2015).

[18] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. “Deepfool: a simple and accurate method to fool deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 2574–2582.

[19] Nicholas Carlini and David Wagner. “Adversarial examples are not easily detected: Bypassing ten detection methods”. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security.* 2017, pp. 3–14.

[20] Seyed-Mohsen Moosavi-Dezfooli et al. “Universal adversarial perturbations”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 1765–1773.

[21] Tianhang Zheng, Changyou Chen, and Kui Ren. “Distributionally adversarial attack”. In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 33. 2019, pp. 2253–2260.

[22] Li Deng and Dong Yu. “Deep learning: methods and applications”. In: *Foundations and trends in signal processing* 7.3–4 (2014), pp. 197–387.

[23] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).

[24] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning.* Vol. 1. 10. Springer series in statistics New York, 2001.

[25] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics.* 2010, pp. 249–256.

[26] Delving Deep into Rectifiers. “Surpassing Human-Level Performance on ImageNet Classification”. In: *Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun* (2015).

[27] Günter Klambauer et al. “Self-normalizing neural networks”. In: *Advances in neural information processing systems*. 2017, pp. 971–980.

[28] Matthew D Zeiler. “Adadelta: an adaptive learning rate method”. In: *arXiv preprint arXiv:1212.5701* (2012).

[29] T Tieleman and G Hinton. “Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning”. In: *Technical Report.* (2017).

[30] Kingma Da. “A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[31] Timothy Dozat. “Incorporating nesterov momentum into adam”. In: (2016).

[32] Caglar Gulcehre, Marcin Moczulski, and Yoshua Bengio. “Adasecant: robust adaptive secant method for stochastic gradient”. In: *arXiv preprint arXiv:1412.7419* (2014).

[33] Paul J Werbos. “Applications of advances in nonlinear sensitivity analysis”. In: *System modeling and optimization*. Springer, 1982, pp. 762–770.

[34] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. “Rectifier nonlinearities improve neural network acoustic models”. In: *Proc. icml*. Vol. 30. 1. 2013, p. 3.

[35] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[36] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015).

[37] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[38] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[39] Davide Maltoni et al. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[40] Nalini K Ratha, Jonathan H Connell, and Ruud M Bolle. “An analysis of minutiae matching strength”. In: *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer. 2001, pp. 223–228.

[41] Anil Jain, Ruud Bolle, and Sharath Pankanti. “Introduction to biometrics”. In: *Biometrics*. Springer, 1996, pp. 1–41.

[42] Richa Jindal and Sanjay Singla. “An Optimised Latent Fingerprint Matching System Using Cuckoo Search”. In: ().

[43] Kasey Wertheim. “Embryology and morphology of friction ridge skin”. In: *The fingerprint sourcebook* (2011), pp. 103–126.

[44] Alessandra Aparecida Paulino. *Contributions to biometric recognition: matching identical twins and latent fingerprints*. Citeseer, 2013.

[45] Asker M Bazen et al. “A correlation-based fingerprint verification system”. In: *Proceedings of the ProRISC2000 workshop on circuits, systems and signal processing*. 2000, pp. 205–213.

[46] Takahiro Hatano et al. “A fingerprint verification algorithm using the differential matching rate”. In: *Object recognition supported by user interaction for service robots*. Vol. 3. IEEE. 2002, pp. 799–802.

[47] Brendan Klare and Anil K Jain. “On a taxonomy of facial features”. In: *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE. 2010, pp. 1–8.

[48] Ming-Hsuan Yang, David J Kriegman, and Narendra Ahuja. “Detecting faces in images: A survey”. In: *IEEE Transactions on pattern analysis and machine intelligence* 24.1 (2002), pp. 34–58.

[49] Xiaoyang Tan and Bill Triggs. “Enhanced local texture feature sets for face recognition under difficult lighting conditions”. In: *IEEE transactions on image processing* 19.6 (2010), pp. 1635–1650.

[50] Matthew Turk and Alex Pentland. “Eigenfaces for recognition”. In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.

[51] Peter N Belhumeur, João P Hespanha, and David J Kriegman. *Eigenfaces vs. fisherfaces: Recognition using class specific linear projection*. Tech. rep. Yale University New Haven United States, 1997.

[52] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[53] Alhussein Fawzi et al. “Classification regions of deep neural networks”. In: *arXiv preprint arXiv:1705.09552* (2017).

[54] A. Fawzi, S. Moosavi-Dezfooli, and P. Frossard. “The Robustness of Deep Networks: A Geometrical Perspective”. In: *IEEE Signal Processing Magazine* 34.6 (Nov. 2017), pp. 50–62. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2740965.

[55] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[56] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 4700–4708.

[57] Daniel L Swets and John Juyang Weng. “Using discriminant eigenfeatures for image retrieval”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 8 (1996), pp. 831–836.

[58] Weihong Deng, Jiani Hu, and Jun Guo. “Extended SRC: Undersampled face recognition via intraclass variant dictionary”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 34.9 (2012), pp. 1864–1870.

[59] John Wright et al. “Robust face recognition via sparse representation”. In: *IEEE transactions on pattern analysis and machine intelligence* 31.2 (2009), pp. 210–227.

[60] Baback Moghaddam, Wasiuddin Wahid, and Alex Pentland. “Beyond eigenfaces: Probabilistic matching for face recognition”. In: *fg*. IEEE. 1998, p. 30.

[61] Xiaofei He et al. “Face recognition using laplacianfaces”. In: *IEEE transactions on pattern analysis and machine intelligence* 27.3 (2005), pp. 328–340.

[62] Chengjun Liu and Harry Wechsler. “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition”. In: *IEEE Transactions on Image processing* 11.4 (2002), pp. 467–476.

[63] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. “Face description with local binary patterns: Application to face recognition”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 12 (2006), pp. 2037–2041.

[64] Yaniv Taigman et al. “Deepface: Closing the gap to human-level performance in face verification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014, pp. 1701–1708.

[65] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 815–823.

[66] Jiankang Deng et al. “Arcface: Additive angular margin loss for deep face recognition”. In: *arXiv preprint arXiv:1801.07698* (2018).

[67] Swami Sankaranarayanan et al. “Triplet probabilistic embedding for face verification and clustering”. In: *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)*. IEEE. 2016, pp. 1–8.

[68] Hongyu Xu et al. “Template regularized sparse coding for face verification”. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE. 2016, pp. 1448–1454.

[69] Yi Sun, Xiaogang Wang, and Xiaoou Tang. “Deeply learned face representations are sparse, selective, and robust”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2015, pp. 2892–2900.

[70] Yandong Wen et al. “A discriminative feature learning approach for deep face recognition”. In: *European conference on computer vision*. Springer. 2016, pp. 499–515.

[71] Rajeev Ranjan et al. “A fast and accurate system for face detection, identification, and verification”. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1.2 (2019), pp. 82–96.

[72] Rohit Kumar Pandey et al. “Deep secure encoding for face template protection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2016, pp. 77–83.

[73] Arun Kumar Jindal, Srinivas Chalamala, and Santosh Kumar Jami. “Face template protection using deep convolutional neural network”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2018, pp. 575–5758.

[74] Liu Chaoqiang, Xia Tao, and Li Hui. “A hierarchical Hough transform for fingerprint matching”. In:

[75] Nalini K Ratha et al. “A real-time matching system for large fingerprint databases”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 8 (1996), pp. 799–813.

[76] Yuliang He et al. “Image enhancement and minutiae matching in fingerprint verification”. In: *Pattern recognition letters* 24.9-10 (2003), pp. 1349–1360.

[77] Lifeng Sha and Xiaoou Tang. “Orientation-improved minutiae for fingerprint matching”. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 4. IEEE. 2004, pp. 432–435.

[78] Dongjae Lee, Kyoungtaek Choi, and Jaihie Kim. “A robust fingerprint matching algorithm using local alignment”. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 3. IEEE. 2002, pp. 803–806.

[79] JeongHee Cha et al. “Fingerprint matching based on linking information structure of minutiae”. In: *International Conference on Computational Science and Its Applications*. Springer. 2004, pp. 41–48.

[80] Kyung Deok Yu, Sangsin Na, and Tae Young Choi. “A fingerprint matching algorithm based on radial structure and a structure-rewarding scoring strategy”. In: *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer. 2005, pp. 656–664.

[81] Joshua Abraham, Paul Kwan, and Junbin Gao. “Fingerprint matching using a hybrid shape and orientation descriptor”. In: *State of the art in Biometrics*. InTech, 2011.

[82] Lu Jiang et al. “A direct fingerprint minutiae extraction approach based on convolutional neural networks”. In: *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE. 2016, pp. 571–578.

[83] Luke Nicholas Darlow and Benjamin Rosman. “Fingerprint minutiae extraction using deep learning”. In: *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2017, pp. 22–30.

[84] Ruxin Wang, Congying Han, and Tiande Guo. “A novel fingerprint classification method based on deep learning”. In: *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE. 2016, pp. 931–936.

[85] Dario Maio and Davide Maltoni. “Neural network based minutiae filtering in fingerprints”. In: *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*. Vol. 2. IEEE. 1998, pp. 1654–1658.

[86] WF Leung et al. “Fingerprint recognition using neural network”. In: *Neural Networks for Signal Processing Proceedings of the 1991 IEEE Workshop*. IEEE. 1991, pp. 226–235.

[87] Yao Tang, Fei Gao, and Jufu Feng. “Latent fingerprint minutia extraction using fully convolutional network”. In: *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2017, pp. 117–123.

[88] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[89] Hongyi Zhang et al. “mixup: Beyond empirical risk minimization”. In: *arXiv preprint arXiv:1710.09412* (2017).

[90] Ralph Gross et al. “Multi-pie”. In: *Image and Vision Computing* 28.5 (2010), pp. 807–813.

[91] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. “From few to many: Illumination cone models for face recognition under variable lighting and pose”. In: *IEEE transactions on pattern analysis and machine intelligence* 23.6 (2001), pp. 643–660.

[92] Raffaele Cappelli et al. “Fingerprint verification competition 2006”. In: *Biometric Technology Today* 15.7-8 (2007), pp. 7–9.

[93] Mohammad A Alsmirat et al. “Impact of digital fingerprint image quality on the fingerprint recognition accuracy”. In: *Multimedia Tools and Applications* 78.3 (2019), pp. 3649–3688.

[94] Mohammad Sabri, Mohammad-Shahram Moin, and Farbod Razzazi. “A New Framework for Match on Card and Match on Host Quality Based Multimodal Biometric Authentication”. In: *Journal of Signal Processing Systems* 91.2 (2019), pp. 163–177.

[95] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[96] Anastasia Borovykh. “A gaussian process perspective on convolutional neural networks”. In: *arXiv preprint arXiv:1810.10798* (2018).

[97] SDK VeriFinger. “Neuro Technology (2010)”. In: *VeriFinger, SDK Neuro Technology* ().

[98] Jia Xiang and Gengming Zhu. “Joint Face Detection and Facial Expression Recognition with MTCNN”. In: *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE. 2017, pp. 424–427.

[99] Dong Yi et al. “Learning face representation from scratch”. In: *arXiv preprint arXiv:1411.7923* (2014).

[100] Helala AlShehri et al. “A Large-Scale Study of Fingerprint Matching Systems for Sensor Interoperability Problem”. In: *Sensors* 18.4 (2018), p. 1008.

[101] Gary B Huang et al. “Labeled faces in the wild: A database forstudying face recognition in unconstrained environments”. In: *Workshop on faces in’Real-Life’Images: detection, alignment, and recognition*. 2008.

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

[103] Tianyue Zheng, Weihong Deng, and Jiani Hu. “Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments”. In: *arXiv preprint arXiv:1708.08197* (2017).

[104] Li Fei-Fei, Rob Fergus, and Pietro Perona. “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories”. In: *Computer vision and Image understanding* 106.1 (2007), pp. 59–70.

[105] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[106] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. “Deep Face Recognition.” In: *British Machine Vision Conference.* Vol. 1. 3. 2015, p. 6.

[107] Yi Sun et al. “Deep Learning Face Representation by Joint Identification-Verification”. In: *Advances in Neural Information Processing Systems.* 2014, pp. 1988–1996.

[108] Yi Sun et al. “DeepID3: Face Recognition with very Deep Neural Networks”. In: *arXiv preprint arXiv:1502.00873* (2015).

[109] Jian Wang et al. “Deep Metric Learning with Angular Loss”. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2017, pp. 2593–2601.

[110] Weiyang Liu et al. “Large-Margin Softmax Loss for Convolutional Neural Networks.” In: *International Conference on Machine Learning.* Vol. 2. 3. 2016, p. 7.

[111] Weiyang Liu et al. “Sphereface: Deep Hypersphere Embedding for Face Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2017, pp. 212–220.

[112] Jiankang Deng et al. “ArcFace: Additive Angular Margin Loss for Deep Face Recognition”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2019, pp. 4690–4699.

[113] Hao Wang et al. “CosFace: Large Margin Cosine Loss for Deep Face Recognition”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.* 2018, pp. 5265–5274.

[114] Evgeniya Ustinova and Victor Lempitsky. “Learning deep embeddings with histogram loss”. In: *Advances in Neural Information Processing Systems.* 2016, pp. 4170–4178.

[115] Matteo Testa et al. “Learning mappings onto regularized latent spaces for biometric authentication”. In: *Proceedings of the IEEE International Workshop on Multimedia Signal Processing.* 2019.

[116] Arslan Ali et al. “Authnet: Biometric Authentication Through Adversarial Learning”. In: *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP).* IEEE. 2019, pp. 1–6.

[117] Christian Szegedy et al. “Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning”. In: *Thirty-First AAAI Conference on Artificial Intelligence.* 2017.

[118] Kai Chen and Wenbing Tao. “Once for all: a two-flow convolutional neural network for visual tracking”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.12 (2017), pp. 3377–3386.

[119] David Held, Sebastian Thrun, and Silvio Savarese. “Learning to track at 100 fps with deep regression networks”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 749–765.

[120] Yann LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551.

[121] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. “Learning representations by back-propagating errors”. In: *Cognitive modeling* 5.3 (1988), p. 1.

[122] Martin Abadi et al. “TensorFlow: A System for Large-Scale Machine Learning”. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation*. 2016, pp. 265–283.

[123] Kaipeng Zhang et al. “Joint Face Detection and Alignment using Multitask Cascaded Convolutional Networks”. In: *IEEE Signal Processing Letters* 23.10 (2016), pp. 1499–1503.

[124] Dong Yi et al. “Learning Face Representation from Scratch”. In: *arXiv preprint arXiv:1411.7923* (2014).

[125] `http://http://trillionpairs.deepglint.com/overview`.

[126] Ira Kemelmacher-Shlizerman et al. “The megaface benchmark: 1 million faces for recognition at scale”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4873–4882.

[127] Brianna Maze et al. “Iarpa janus benchmark-c: Face dataset and protocol”. In: *2018 International Conference on Biometrics (ICB)*. IEEE. 2018, pp. 158–165.

[128] Gary B Huang et al. “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments”. In: 2008.

[129] Lior Wolf, Tal Hassner, and Itay Maoz. *Face Recognition in Unconstrained Videos with Matched Background Similarity*. IEEE, 2011.

[130] Tianyue Zheng, Weihong Deng, and Jiani Hu. “Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments”. In: *arXiv preprint arXiv:1708.08197* (2017).

[131] Tianyue Zheng and Weihong Deng. “Cross-Pose LFW: A Database for Studying Crosspose Face Recognition in Unconstrained Environments”. In: *Beijing University of Posts and Telecommunications, Tech. Rep* (2018), pp. 18–01.

[132] Soumyadip Sengupta et al. “Frontal to Profile Face Verification in the Wild”. In: *2016 IEEE Winter Conference on Applications of Computer Vision*. IEEE. 2016, pp. 1–9.

[133] Stylianos Moschoglou et al. “Agedb: the first manually collected, in-the-wild age database”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 51–59.

[134] Weiyang Liu et al. “Learning Towards Minimum Hyperspherical Energy”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 6222–6233.

[135] Yandong Wen et al. “A Discriminative Feature Learning Approach for Deep Face Recognition”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 499–515.

[136] Jingtuo Liu et al. “Targeting Ultimate Accuracy: Face Recognition via Deep Embedding”. In: *arXiv preprint arXiv:1506.07310* (2015).

[137] Xiao Zhang et al. “Range Loss for Deep Face Recognition with Long-tailed Training Data”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5409–5418.

[138] Jiankang Deng, Yuxiang Zhou, and Stefanos Zafeiriou. “Marginal Loss for Deep Face Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 60–68.

[139] Cecilia Summers and Michael J Dinneen. “Improved Adversarial Robustness via Logit Regularization Methods”. In: *arXiv preprint arXiv:1906.03749* (2019).

[140] Seyed-Mohsen Moosavi-Dezfooli et al. “Robustness via curvature regularization, and vice versa”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9078–9086.

[141] Yair Carmon et al. “Unlabeled data improves adversarial robustness”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 11190–11201.

[142] Rafael Pinot et al. “Theoretical evidence for adversarial robustness through randomization”. In: *Advances in Neural Information Processing Systems*. 2019, pp. 11838–11848.

[143] Nicholas Carlini et al. “On evaluating adversarial robustness”. In: *arXiv preprint arXiv:1902.06705* (2019).

[144] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[145] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv:1708.07747* (2017).

[146] Yuval Netzer et al. “Reading digits in natural images with unsupervised feature learning”. In: (2011).

[147] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer, 2009.

[148] Aleksander Madry et al. "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083* (2017).

[149] Nicolas Papernot et al. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[150] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world". In: *arXiv preprint arXiv:1607.02533* (2016).

[151] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv:1312.6199* (2013).

[152] Nilesh Dalvi et al. "Adversarial classification". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 99–108.

[153] Daniel Lowd and Christopher Meek. "Adversarial learning". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 641–647.

[154] Yanpei Liu et al. "Delving into transferable adversarial examples and black-box attacks". In: *arXiv preprint arXiv:1611.02770* (2016).

[155] Nicolas Papernot et al. "Distillation as a defense to adversarial perturbations against deep neural networks". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 582–597.

[156] Ali Shafahi et al. "Are adversarial examples inevitable?" In: *arXiv preprint arXiv:1809.02104* (2018).

[157] Andrew Slavin Ross and Finale Doshi-Velez. "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients". In: *Thirty-second AAAI conference on artificial intelligence*. 2018.

[158] Mahmood Sharif et al. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 1528–1540.

[159] Tom B Brown et al. "Adversarial patch". In: *arXiv preprint arXiv:1712.09665* (2017).

[160] Matthias Hein and Maksym Andriushchenko. "Formal guarantees on the robustness of a classifier against adversarial manipulation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 2266–2276.

[161] Daniel Jakubovitz and Raja Giryes. “Improving dnn robustness to adversarial attacks using jacobian regularization”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 514–529.

[162] Alireza Makhzani et al. “Adversarial autoencoders”. In: *arXiv preprint arXiv:1511.05644* (2015).

[163] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[164] Andre Stuhlsatz, Jens Lippel, and Thomas Zielke. “Feature extraction with deep neural networks by a generalized discriminant analysis”. In: *IEEE transactions on neural networks and learning systems* 23.4 (2012), pp. 596–608.

[165] Matthias Dorfer, Rainer Kelz, and Gerhard Widmer. “Deep linear discriminant analysis”. In: *arXiv preprint arXiv:1511.04707* (2015).

[166] Jieping Ye and Shuiwang Ji. “Discriminant analysis for dimensionality reduction: An overview of recent developments”. In: *Biometrics: Theory, Methods, and Applications. Wiley-IEEE Press, New York* (2010).

[167] Matteo Testa et al. “Learning mappings onto regularized latent spaces for biometric authentication”. In: *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2019, pp. 1–6.

[168] DN Joanes and CA Gill. “Comparing measures of sample skewness and kurtosis”. In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 47.1 (1998), pp. 183–189.

[169] Xavier Gastaldi. “Shake-shake regularization”. In: *arXiv preprint arXiv:1705.07485* (2017).

[170] Akhilesh Gotmare et al. “A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation”. In: *arXiv preprint arXiv:1810.13243* (2018).

[171] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[172] Tianhang Zheng, Changyou Chen, and Kui Ren. “Is pgd-adversarial training necessary? alternative training via a soft-quantization network with noisy-natural samples only”. In: *arXiv preprint arXiv:1810.05665* (2018).

[173] Todor Davchev et al. “An empirical evaluation of adversarial robustness under transfer learning”. In: *arXiv preprint arXiv:1905.02675* (2019).