computers

MDPI

Article

Utilizing Transfer Learning and Homomorphic Encryption in a Privacy Preserving and Secure Biometric Recognition System

Milad Salem, Shayan Taheri and Jiann-Shiun Yuan *

Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816-2362, USA; milad73s@gmail.com (M.S.); shayan.taheri@gmail.com (S.T.)
* Correspondence: yuanj@mail.ucf.edu; Tel.: +1-407-823-5719

Received: 4 December 2018; Accepted: 26 December 2018; Published: 29 December 2018

Abstract: Biometric verification systems have become prevalent in the modern world with the wide usage of smartphones. These systems heavily rely on storing the sensitive biometric data on the cloud. Due to the fact that biometric data like fingerprint and iris cannot be changed, storing them on the cloud creates vulnerability and can potentially have catastrophic consequences if these data are leaked. In the recent years, in order to preserve the privacy of the users, homomorphic encryption has been used to enable computation on the encrypted data and to eliminate the need for decryption. This work presents DeepZeroID: a privacy-preserving cloud-based and multiple-party biometric verification system that uses homomorphic encryption. Via transfer learning, training on sensitive biometric data is eliminated and one pre-trained deep neural network is used as feature extractor. By developing an exhaustive search algorithm, this feature extractor is applied on the tasks of biometric verification and liveness detection. By eliminating the need for training on and decrypting the sensitive biometric data, this system preserves privacy, requires zero knowledge of the sensitive data distribution, and is highly scalable. Our experimental results show that DeepZeroID can deliver 95.47% F1 score in the verification of combined iris and fingerprint feature vectors with zero true positives and with a 100% accuracy in liveness detection.

Keywords: biometrics; convolutional neural network; deep learning; fingerprint; homomorphic encryption; iris; privacy; transfer learning

1. Introduction

Biometrics is a tool to automatically distinguish subjects in a reliable manner for a target application based on the derived signals from physical or behavioral traits (such as fingerprint, iris, palm veins, face, DNA, voice pattern, facial pattern, and hand geometry). In comparison to the classical security methods (including PIN, password, key, and card), this technology provides several benefits such as being a unique identification of individuals, mobile, very hard to forge, always with the user (no external carrying), user friendly, and secure. The process of recognizing the objects/individuals in an automated manner using their biometric data is called biometric recognition system (BRS). A BRS has had applications in the law enforcement for decades in authentication of individuals; however, nowadays smartphones rely on biometrics for verification of the user as well. Traditionally, these systems include server-side database owner and users who submit candidate biometric records for verification of the identity profiles.

These data can be used to measure biological characteristics for identification and classification of entities. Two biometric data that have gained significant attention are iris and fingerprint. Fingerprint is the fundamental and traditional elements to use for identification of human beings. This element,

as shown in Figure 1, includes a pattern of ridge and valleys on the surface of a fingertip that is formed during the starting months of development of fetus. Even, twins from the same parents or the prints from the fingers of the same person are not the same. Multiple fingerprints of one person can provide additional information to allow higher level of recognition. Small cuts and bruises on the fingerprints or other factors such as aging or being exposed to environmental parameters can cause degradation of the recognition accuracy.

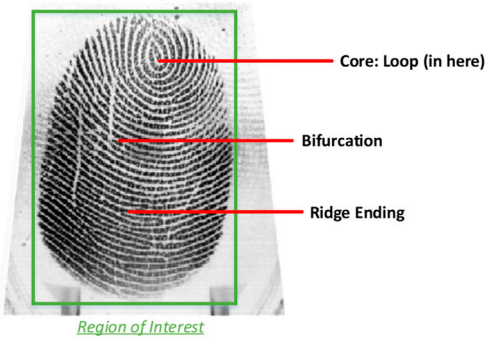


Figure 1. An authentic fingerprint.

Iris is another and more recent element for human recognition. As shown in Figure 2, it is the annular region of the eye, which is bounded by the pupil and the sclera on either side. Similar to fingerprint, its texture and structure are formed during fetal development. They are stabilized after few years from its formation. The iris texture holds unique information for recognition and identification purposes which can be leveraged to provide high system accuracy. As opposed to fingerprint, changing or tampering the iris pattern does not happen easily. The preprocessed format of iris that is ready for feature (i.e., the unique information) extraction is shown in Figure 3.

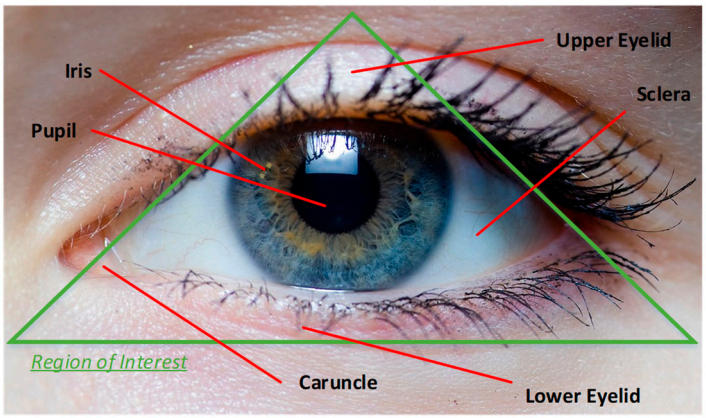


Figure 2. A human eye.

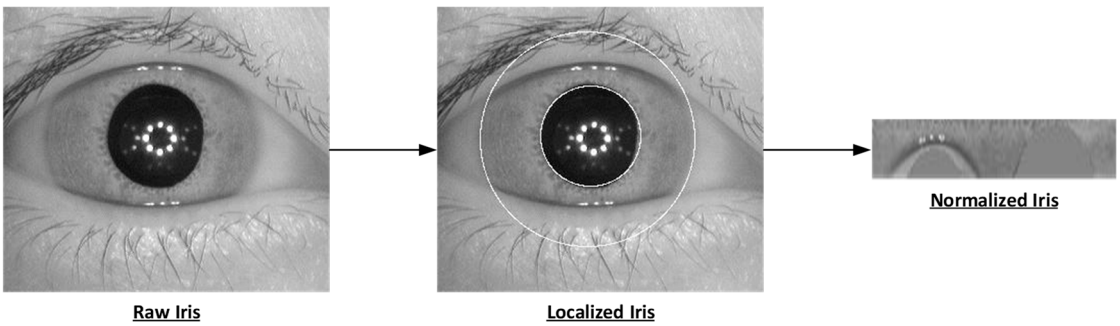


Figure 3. The raw, localized, and normalized version of an iris.

In a modern biometric recognition system, the biometric data of the users collected during registration are stored in the cloud and are accessed during verification. Storing the biometric data in the cloud enables the verification process, performing numerous computations, and storing a large volume of data. Despite all these benefits, this technology is not without its side effects. The security and privacy of information heightens when biometric data is being shared, since biometric data, unlike passwords and PINs, cannot be reset and one leak of information can have inevitable consequences. These data should not be exposed to third party and should be under control anywhere outside the user side. To address this problem, the biometric data stored in the cloud should be secured through encryption. By having encrypted data within the cloud, the privacy of entities are preserved.

However, simply encrypting the data makes it unusable with the computing processes in the cloud. Therefore, it is required to provide a framework for applying mathematical functions on the encrypted data on the cloud side. This is a mandatory requirement; otherwise the plain and unencrypted biometric information is at risk and can be changed or leaked. Additionally, encrypting the data by any arbitrary encryption scheme is not possible since applying a minor change on the plain information can lead to major change in the ciphertext. This major change leads to unrecognizable data after decryption.

This challenge can be solved through encrypting the sensitive data using a scheme that has homomorphism. When the data is homomorphically encrypted, certain operations can be carried out using the ciphertext, and leveraging this homomorphism enables cloud computing. In this way, the data is processed for recognition purposes without sharing the plaintext information. Therefore, a homomorphically encrypted data is protected against the attackers and honest-but-curious servers since it is never decrypted.

Another challenge in this domain relates to the recognition process. A successful recognition of biometric data highly depends on the development of efficient feature extractors that are capable of obtaining meaningful information from the data. Traditionally hand-crafted, manual, or trial-and-error-based feature extractors were used to represent the biometric data and help in the recognition task. In the recent years, with the advent of deep learning, neural networks have been deployed in this domain they have been proven to extract useful features and information from the data and achieve high accuracy. A recognition system that takes advantage of these networks, given enough data, can surpass human-made features and human-level accuracy in many applications. However, compared to other similar image recognition tasks, gathering a large amount of data is arduous in this domain. Moreover, a trained model on the biometric data holds information about the data distribution and can be a point of vulnerability. A solution to these challenges is applying transfer learning and transferring knowledge from another task to the biometric verification task.

In this work a deep learning-based biometric recognition system within the privacy preserving domain is developed. In this system a pre-trained deep neural network is used to extract features from the biometric data, alleviating the problems of small data and eliminating the need to train a model and the possibility of information leakage through the model. Moreover, the biometric data is encrypted using a partially homomorphic encryption and stays safe within the system while verification is being carried out. There are multiple parties in our computation model, including client, authentication server, database, and matcher that provide an added level of security and privacy. We also show that the pre-trained network is useful not only for recognition of different entities, but also for detection of true and fake biometric data.

The system extracts the features of biometric data from a deep neural network (known as off-the-shelf features). To preserve the privacy of extracted features and make them safe and protected, they are masked and also encrypted using Paillier Chunkwise. Then, the encrypted features are recognized on the cloud-side without being decrypted, leveraging the additive homomorphism property of the applied encryption method. To further increase the system security and make its recognition functionality more accurate, a true/fake detector is implemented on the user side.

This detector utilizes the extracted deep features by inputting them into a Support Vector Machine (SVM) for distinguishing the true and the fake biometric data.

Our main contribution can be stated as:

1. Proposing DeepZeroID system, which makes a bridge between deep features, homomorphic encryption, and biometric security. Moreover, the running protocol among all parties within the system that takes an encrypted and masked data for query computations has been demonstrated. In this way, accessing, computing, and storing the data by the agents and parties inside the framework become more secure. This system has the capability of having zero information leakage for two reasons. Firstly biometric data stays encrypted in the system. The encryption prevents the attackers from gaining access to any sensitive data or the contents of the individual queries. Secondly the neural network used as a feature extractor is not trained on biometric data and has no knowledge of the data distribution. This lack of knowledge enables the scalability of the proposed system as well, since new user can be added without the need for changing the feature extractor.
2. Development of CNNOptLayer, which is an algorithm that performs an exhaustive search operation among all layers of the convolutional neural network under process. It is capable of finding the optimal layer for feature extraction.
3. Inclusion of a single Convolutional Neural network (CNN) as the feature extractor for multiple tasks within the system (namely iris/fingerprint recognition and true/fake detection). The feature extraction is performed based on leveraging the CNNOptLayer algorithm.
4. Improving the encryption speed of a CNN-based privacy preserving biometric recognition system by utilization of Paillier Chunkwise.
5. Presentation of new attacks and malicious scenarios for deep learning-based biometric recognition system and demonstrating the weaknesses and deficiencies of the system under these attacks.

The rest of this paper is structured as: the background of the work is presented in Section 2. In the background section, the general information about transfer learning, homomorphic encryption, the leveraged deep neural networks within the biometric system, and true/fake detection of biometric data (which is a part of our system) are discussed. Section 3 discusses the related works. Section 4 describes our proposing system and methodology. In this section, the general overview of our privacy preserving biometric recognition system is explained. The processing flow of biometric data from the user-side perspective is described as well. At its end, the process of deciding on a biometric data input is denoted. Section 5 centers around experimental approach and the results, discusses the experimental setup and the process of selecting dataset, shows the final results, and analyzes security aspect of the system. Moreover, a discussion and limitations of the system are provided in its last part. The work is concluded in Section 6.

2. Background

In this section, the fundamental concepts and techniques that are utilized in this work are discussed. These concepts and techniques include transfer learning, homomorphic encryption, leveraged pre-trained deep neural networks (which are DenseNet and AlexNet), and the process of true/fake detection of the biometric data.

2.1. Transfer Learning

The process of making predictions on the future data is done using the statistical models that are trained on previously collected labeled or unlabeled training data within the context of traditional data mining and machine learning algorithms. If there is sufficient labeled data for gaining knowledge, then a fair classifier can be built. Within this context, the semi-supervised classification addresses the problem of insufficient labeled data for building a good classifier through utilizing a large amount of unlabeled data and a small amount of labeled data. Most of the techniques in this domain assume that

the distributions of the labeled and the unlabeled data are the same. Also, each task is learned from the scratch in these techniques.

However, in the cases where there is noticeable insufficiency in the labeled data, transfer learning helps to overcome this issue base on allowance of difference among the domains, tasks, and distributions used in training and testing. This technique helps to apply knowledge learned previously (from source tasks) to solve new problems (target tasks) faster and/or with higher quality solutions. According to this technique, the knowledge and patterns extracted from certain data can be helpful in representing another data distribution.

In this work, we use a deep neural network pre-trained on a thousand classes of ImageNet dataset, consisting of many different classes such as objects and animals. The biometric data is fed to this neural network and the output of different layers is considered as the feature representation of the input data.

2.2. Homomorphic Encryption

Homomorphism is mapping of a mathematical set into another set or onto itself in such a way that applying mathematical operations to the elements of the source set is mapped into the elements of the target set. Using this property, a dataset can be transformed into another while preserving the relationships among their elements. Leveraging homomorphism during the encryption process helps to perform certain types of computations on the ciphertext. The result of operations on the ciphertext is also a ciphertext, which if decrypted results in the same outcome of applying the mapped operations on the initial plain information. This is different from other types of encryption according to which applying any change on the ciphertext causes damage of the plain information when it is decrypted. This type of encryption is called Homomorphic Encryption.

Nowadays, industries, companies, organizations, and any other private institutes allow storing their information in a public cloud to access their computing and analytics services. Theoretically, a fully homomorphic encryption scheme [1] allows the computing and analytics services to be done in the cloud in a protected and secure way. Therefore, cloud computing platforms can perform complex and complicated computations on homomorphically encrypted data without ever having access to the unencrypted data. As a result, arbitrary computations can be applied on the encrypted data, while the features of the functions and the format of the encrypted data remain preserved. However, the efficiency and speed of these computations, at the moment, are drastically low, causing hindrance of leveraging fully homomorphic encryption by its full capacity.

Within this context, the Paillier cryptosystem is a probabilistic asymmetric algorithm for public key cryptography and a partially homomorphic encryption. This system is additively homomorphic, meaning it supports addition and is semantically secure. Its probabilistic property causes randomness in production of ciphertext. Due to the decisional composite residuosity assumption (DCRA) property of the Paillier encryption scheme, an encrypted data using this system is secure against honest but curious servers and users. The properties of homomorphic Paillier encryption system can be stated as: (a) the encrypted numbers can be multiplied by a non-encrypted scalar; (b) the encrypted numbers can be added together; (c) the encrypted numbers can be added to non-encrypted scalars. So, these operations hold for a Paillier encryption system: (i) the product of two ciphertexts is decrypted to the sum of their corresponding plaintexts. (ii) the multiplication of a ciphertext and a random number with the power of a plaintext decrypted to the sum of the corresponding plaintexts. (iii) an encrypted plaintext with having another plaintext as its power is decrypted to the product of the two plaintexts.

2.3. Leveraged Deep Neural Networks: DenseNet and AlexNet

The algorithms within the deep learning domain learn the complex, representative, and discriminative features in a hierarchical way from the high dimensional data. These architectures of these algorithms are usually constructed as multi-layer networks in a way to have more computation of abstract features as nonlinear functions of lower-level features. They are used to build a model

that relates the inputs to the outputs based on modeling complex non-linear relationships in both supervised and unsupervised settings.

These algorithms have applications in a variety of domains ranging from image processing, computer vision, speech recognition, natural language processing, communication patterns, pixel-based classification, and target recognition, to high-level semantic feature extraction. A deep learning method can be categorized as supervised, semi-supervised, or unsupervised. Two deep neural networks are introduced in this section due to their utilization in our system: AlexNet and DenseNet. In the ImageNet Large Scale Visual Recognition Challenge (in 2012), the AlexNet (i.e., the challenge winner) was introduced that is a convolutional neural network written in the CUDA platform. The network is usually made of five convolution layers, max-pooling layers (with local response normalization), dropout layers, and three fully connected layers. A softmax layer at the end classifies the input data. It showed more than 10% accuracy higher than the second-ranked network in the competition and outperformed all its predecessors in the challenge.

The other network utilized in this work is Dense Convolutional Network (DenseNet). This network is a stack of dense blocks followed by transition layers. According to its architecture, each layer is connected to every other layer in a feed-forward format (within each dense block). This means each layer is connected to the entire earlier layers (which provide feature re-use). Each block is made from a series of units, which each packs two convolutions, batch normalization, and ReLU activations. The output of each unit is a fixed number of feature vectors. According to this parameter, the flow of information through the layers is controlled. For each layer, the feature maps of all preceding layers are treated as separate inputs whereas its own feature maps are passed on as inputs to all subsequent layers. DenseNets have many persuading advantages namely, reducing the vanishing-gradient problem, strengthening feature propagation, encouraging feature reuse, and reducing the number of parameters substantially. This network architecture is not only efficient, but also has the big advantage of improved flow of information and gradients throughout the network. The dense connections in this network have a regularizing effect that reduces over-fitting on tasks with smaller training set sizes. Moreover, due to allowance of feature reuse among the Dense units, its structure tends to be more compact in comparison to its counterparts. The most noticeable trend in the network behavior is its easy training and higher accuracy in comparison to the other state-of-the-art networks with less number of network parameters. The architectures of these networks are shown in Figure 4. In this work a pre-trained DenseNet on the ImageNet dataset is used as a feature extractor. Moreover, AlexNet is used to demonstrate the vulnerabilities of deep neural networks against attacks.

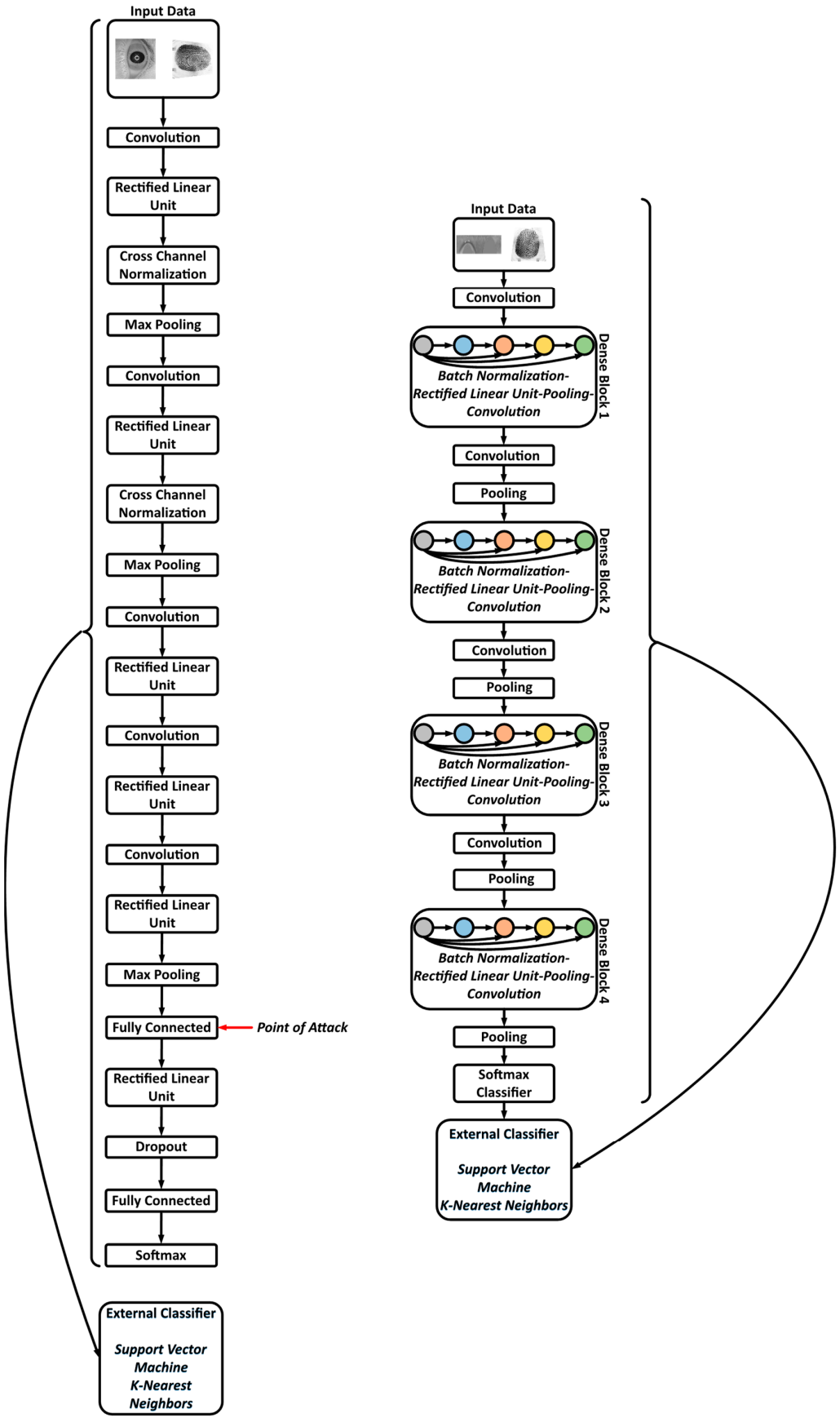


Figure 4. The AlexNet CNN architecture (**left**) and the DenseNet CNN architecture (**right**).

2.4. True/Fake Detection of Biometric Data

Biometric data presents several benefits over classical data (i.e., password, key, card, and so forth) for provision of security. In order to attack a biometric recognition system, the easiest way is to deliver a fake biometric data. This attack is called spoofing and is of great importance to the research community. This attack can be used to find the vulnerabilities against a recognition system of the iris, the fingerprint, the face, and the signature. The main strength of this attack is its defensive capability against digital protection mechanisms, including encryption, digital signature, or watermarking. A strong and protected biometric recognition system can distinguish authentic fingerprint or iris from fake ones. A number of true and fake biometric data (i.e., fingerprint and iris) employed in this work are shown in Figure 5.

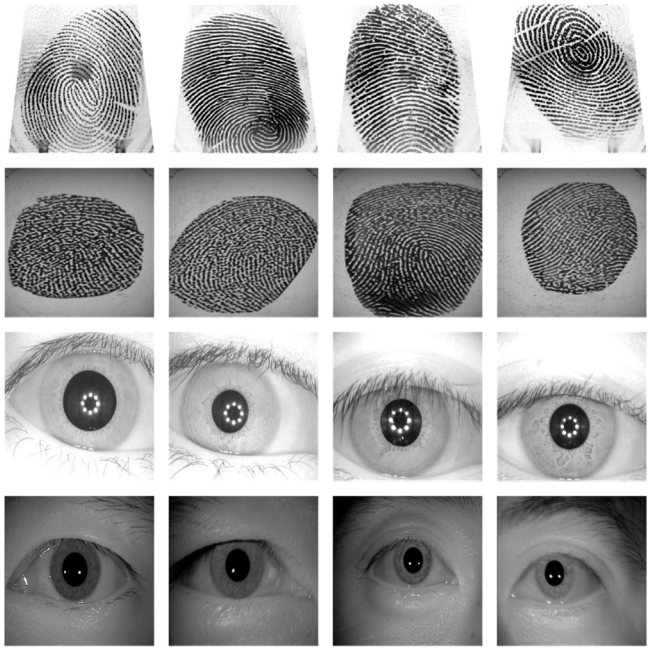


Figure 5. The samples of true fingerprint (first row), fake fingerprint (second row), true iris (third row), and fake iris (fourth row).

The systems for discriminating between true and fake biometric data can be classified into two parts: (a) hardware; and (b) software. In the hardware-based systems, a specific device is added to the sensor in order to detect particular properties of a living trait such as blood pressure or skin distortion. The hardware-based systems display higher detection rate. In the software-based systems, fake traits are detected once the sample has been acquired with a standard sensor. These systems are less expensive and less intrusive and can be integrated in any part of the recognition system. Using software-based modules, the system can be protected against external injection of malicious data samples.

The traits for distinguishing the real and the fake data are extracted from the image instances of fingerprint or iris. There are a number of ways to extract features: (ix) the manual descriptors including usage of local amplitude contrast (spatial domain) and phase (frequency domain) for formation of a bi-dimensional contrast-phase histogram, (iy) local phase quantization (LPQ) for texture derivation, and local binary pattern (LBP) with wavelet. (iz) convolutional neural network. The countermeasures for this attack are stated as: (a1) utilization of multi-biometrics; (a2) using challenge-response methods; and (a3) liveness detection techniques. The last technique has shown significant performance in recent years and uses different physiological properties to distinguish the real and fake traits.

The protection methods based on the liveness assessment need to satisfy certain requirements: (i1) being non-invasive; (i2) being low cost, which implies the possibility of its wide usage if it is affordable;

(i3) delivering high performance, which means the detector needs to demonstrate a good accuracy, while it does not cause any degradation on the system recognition performance. Finally, development of a robust liveness assessment system which satisfies these requirements can improve the integrity and correctness of the overall biometric recognition system.

3. Related Work

In this section, the related works to the application of deep learning in recognition systems as well as the applications of homomorphic encryption schemes are described. While traditionally recognition systems use manually extracted features to represent the input data [2], the state of the art systems have proven that end-to-end systems that allow the neural network to perform the feature extraction autonomously have shown higher accuracies [3]. Moreover, homomorphic encryption has been involved in these recognition systems to preserve the privacy. A. Ene et al. [4] proposed implementing speech recognition system and preserving the identity of users (or speakers) through leveraging homomorphic operations and usage of large amount of plaintext space. In [5], the authors proposed executing computationally intensive biometric recognition system by offloading the recognition process to the cloud. In this technique, the recognition-based operations as well as bulk enrollment operations are divided into multiple tasks, to be executed on a set of servers in the cloud. In order to further improve the privacy and security of biometric data, it is offered to make them cancelable when they are stored in the cloud. The work [6] presented a secure and privacy-preserving mechanism for authentication of users based on their biometric data in a distributed framework. In order to improve the security and privacy, three modalities are combined based on a weighted score level fusion to determine the final multimodal data. To protect biometric data storage, they proposed processing the data in a multi-party framework that enhances security in all stages of authentication. Therefore, attacking a single database does not significantly jeopardize the security of the data. This framework not only provides security, but also improves usability, execution time, and efficiency.

In [7], a framework is shown that has the duty of protection and privacy provision within the context of having a large amount of information. This framework consists of two layers of protection, the first layer of which provides robust hash values as queries and the second layer provides an ability for the client to modify certain bits in a hash value to prevent original content or features from being revealed. This scaling of information helps make computations more difficult on the server based on the interest of the client. This interaction of client and server within a protected environment helps to preserve their privacy. A secure system for multi-biometric data has been proposed in [8] that uses deep neural networks and error-correction coding. The multi-biometric data is generated by a feature-level fusion framework with the input of multiple biometric data. Via making the multi-biometric data cancelable, they further secure the privacy and confidentiality of the users. The PassBio has been proposed in [9] according to which a user-centric biometric authentication scheme is offered that gives this ability to users to encrypt the biometric templates with a light weight encryption scheme. The encrypted data stay in the server and will never be accessed directly. In this framework, the privacy and protection are catered through running the "compute-then-compare" computational model coupled with the threshold predicate encryption.

The authors in [10] provided good answers for characterization of biometric designs based on privacy enhancing technologies. Through answering these questions, the regulations for the protection of biometric information are presented and the cryptographic techniques for design of a secure biometric system are analyzed and compared. In addition, a privacy-preserving approach for authentication of biometric data within the context of mobile applications is proposed. The proposed model uses a mechanism according to which pseudonymous biometric identities are used for securing the registration and authentication of biometric identities. In [11], a basic fusion model blueprint for preserving the privacy of cloud-based user verification/authentication is proposed. It is considered that the three modalities of biometric data are located in different databases of semi-honest providers. They are combined based on their performance parameters (i.e., weighted score). It was proposed

in [12] that a distributed setting of clients, cloud server, and service provider with verifiable interactions (to be executed on top of a homomorphic encryption scheme) can help improve security against malicious servers. Taheri et al. in [13] showed the biometric recognition systems (specifically for fingerprint and iris) are not secure due to possible presence of hardware and software Trojans inside the system. In their work, they proposed how hardware Trojans can manipulate the image instances of the iris and the fingerprint, leading to denial of service in many of the existing biometric recognition systems. Accordingly, a cross-layer recognition system is developed that performs security-based data analysis of biometric data in two levels and is strong enough in confronting the designed hardware Trojans.

The authors in [14] proposed using a privacy-preserving biometric identification for face recognition based on eigen-face approach. In their technique, Paillier cryptosystem is used as an additive homomorphic encryption unit. For finding the difference between the face image vector from the client and the server's database, Euclidean distance is employed. Inside their framework, a matcher is used to compare the information within the encrypted domain in order to avoid revealing any information.

A privacy-preserving face identification has been proposed in [15] according to which the facial images are presented by binary feature vectors. In its implementation, additive homomorphic encryption and oblivious transfer have been used. In order to measure the similarity between the images, the Hamming distance has been used. An efficient matching protocol has been proposed in [16] with having application in many privacy-preserving biometric identification systems inside a semi-honest setting. A more efficient protocol is proposed by the authors that computes the Euclidean distances for improving the privacy and security of the matching system. A novel privacy-preserving biometric identification scheme was proposed in [17] that achieves efficiency through exploitations of the cloud computing power. The scheme provides outsourcing of biometric data to the cloud servers. The identification of biometric data occurs through generation of a credential for the candidate biometric trait and its submission to the cloud. On the cloud side, the identification happens over the encrypted data using the credential. This identification has the advantage of real time computation, low communication cost, and secure outsource of data to the cloud. In addition, the problem of training high quality word vectors over large-scale encrypted data within the context of privacy-preserving is tried to be solved by designing a suite of arithmetic primitives on encrypted data.

A privacy-preserving identification mechanism for mobile sensing is proposed in [18] that selects sensed data dynamically in order to protect the sensitive information of participants. This mechanism solves the contradiction between the protection of user privacy and performing the task of the identification. The privacy and sensitivity of the data are catered by letting the users to define their sensitivity and selecting the sensed data dynamically. The identification part is given by training a two-layered neural network and learning the user behavior in order to generate an identity for it. In [19] an efficient and privacy-preserving identification system for fingerprint data was presented using cloud systems. Within this context, the cloud has the duty of exploiting the computation power for extensive mathematical computations. [20] proposed a privacy-preserving identification system that outsources the encrypted biometric data into the cloud and is efficient in computations. All the identification operations within the cloud are executed on the encrypted data and they are returned to the database owner. A complete security analysis shows that this scheme is secure even if attackers forge requests.

In [21] proposed a secure face verification scheme using a specifically trained neural net. They extracted the features from the last layer of the network. In comparison to that work, we eliminate training on sensitive data, use a faster encryption scheme, and find the optimum layer to train on using CNNOptLayer. It has been proven that transfer learning can increase the accuracy when small data is presented, and in this work, the case is similar with the presence of small data and the need for feature extractors. Therefore, using the same concept of deep features can help the verification task. Moreover, in this work an algorithm is presented to assist in finding the optimum layer for feature extraction.

Furthermore, by adding homomorphic encryption and eliminating the need for training, this work leverages a pre-trained DenseNet to preserve the security of the biometric data.

4. Methodology of the Proposed System

The proposed system in this work caters privacy-preserving capability to a deep learning-based biometric recognition system that receives queries from users. In this system four elements are involved, namely client (from cellphone/computer), matcher (from cloud), database (different means can be used for this purpose), and authentication server (from cloud). The data stored on the database is encrypted personal records which prevents attackers from gaining access to the sensitive information of the enrolled users. The DeepZeroID system uses only fingerprint and iris, but it can be extended to other biometric images as well. For every query, the region of interest within the biometric data needs to be localized and processed before it is sent to the recognition system.

4.1. General Overview of Privacy-Preserving Biometric Recognition System

The network architecture and the associated biometric recognition system (including their elements) are shown in Figure 6. The details of this system are depicted in Figure 7. As it can be observed from the figure, there are four sides in the network namely client, matcher, database, and authentication server.

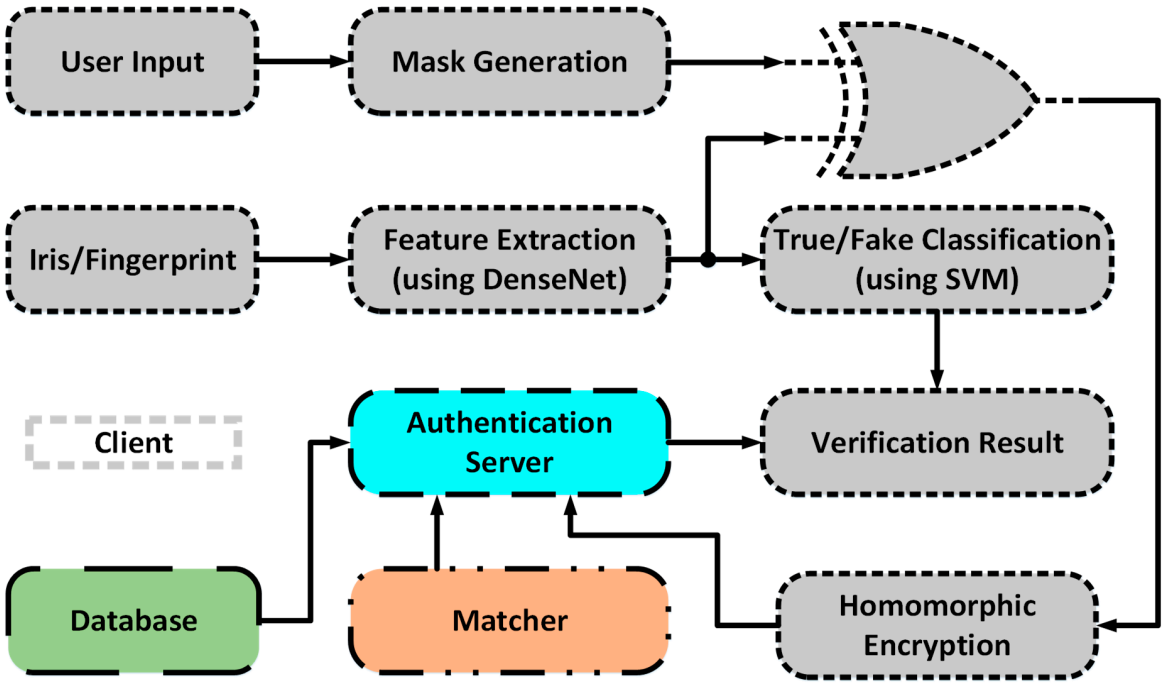


Figure 6. The proposed deep learning-based privacy preserving biometric recognition system.

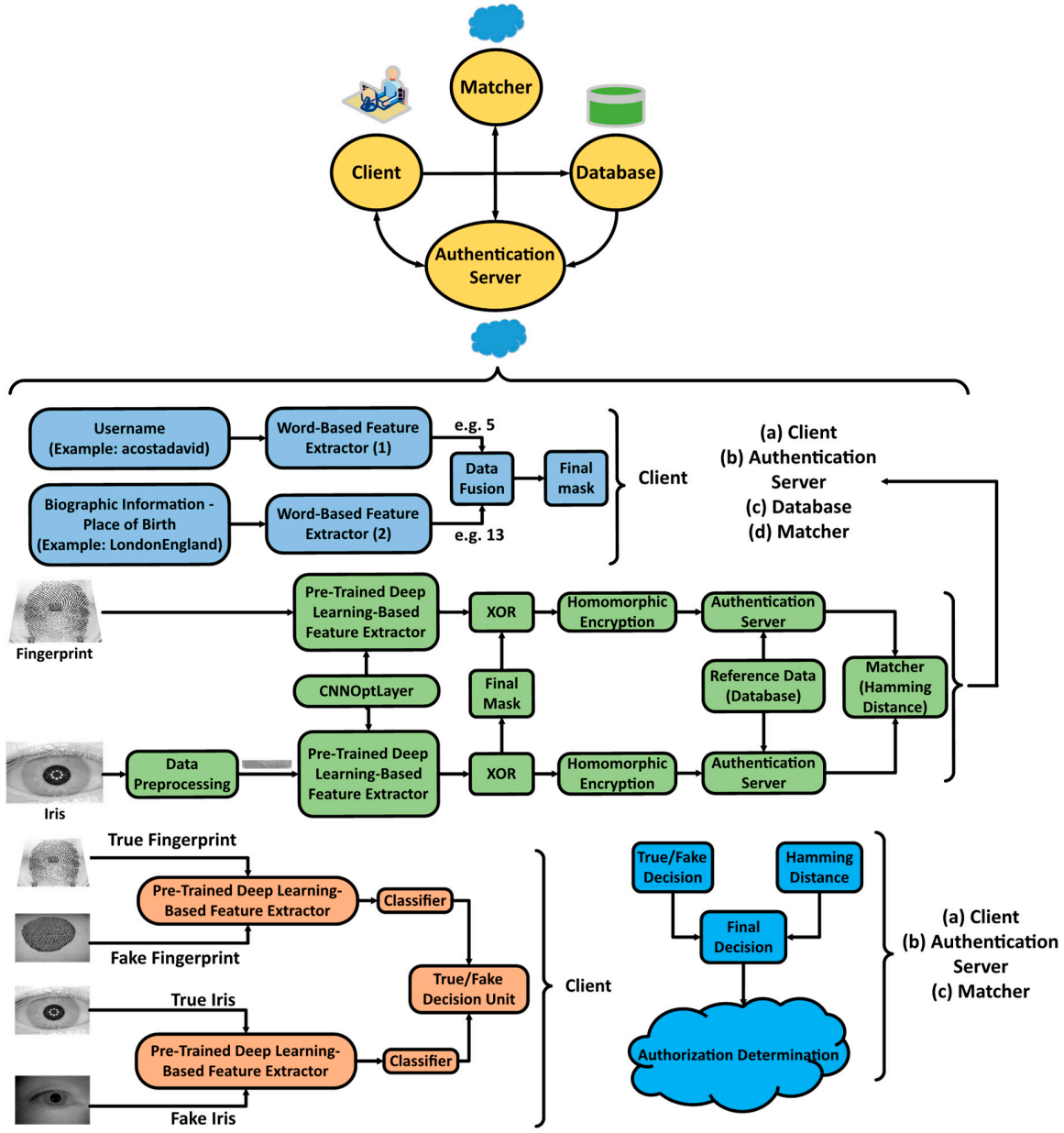


Figure 7. The details of the proposed system.

On the client-side, a person provides his/her fingerprint, iris, username, and biographic information (which is the place of birth in here). The inputted iris image is segmented in order to find the region of interest within the eye and then normalized; however, the fingerprint stays in its raw format. These images are then fed to a pre-trained deep neural network (DenseNet) and the outputs of specific layers of this network are extracted as the feature vectors for the inputs. An algorithm namely CNNOptLayer is developed to find the optimal layer for feature extraction. In parallel to this process, the username and biographic information that was gathered from the user are delivered to two word-based feature extractors. After extraction, these two word-based features, are concatenated and replicated to create a binary mask for the user. Having done so, the extracted features from the fingerprint or iris are binarized and bit-masked, and then up-sampled and encrypted using a partially homomorphic encryption scheme. On the other side, two SVM classifiers trained on the true and fake fingerprint and iris features are used to determine the type of the input biometric data in terms of being true or fake.

Outside of the user side, the encrypted features are stored in a database for future matching. The future queries are compared to the reference features by a matcher within the cloud. In order to verify an input, their encrypted binary vector is sent to the authentication server and is compared to the reference vector with the XOR operation being carried out. The results of the XOR operation are sent to the matcher unit, where the hamming distance is calculated. If the hamming distance is smaller than a certain threshold, the vectors match. The input is verified if the results of the verification system as well as the authenticity system are positive. The flow and protocol for this system is shown in Algorithm 1.

Algorithm 1: The protocol and overall scheme of deep learning-based privacy preserving bi-modal biometric recognition system.

01: **Input Parameters:** True and Fake Biometric Data, Username, and Biographic Information
02: **Output Parameters:** Output: Authorization Determination
03: Client:
04: BiometricData ← DenseNet Features & CNNOptLayer (TrueFingerprint or Preprocessed Iris)
05: BioInfoData1 ← WordBasedFeatureExtractor1 (Username)
06: BioInfoData2 ← WordBasedFeatureExtractor2 (BioInfo)
07: FinalMask ← WordDataFusion(BioInfoData11, BioInfoData2)
08: Client-AuthenticationServer-Database-Matcher:
09: PlainData ← XOR (BiometricData, Final Mask)
10: Ref / TestEncryptedData ← Homomorphic Encryption Scheme (PlainData)
11: Matcher ← XOR (Ref EncryptedData1, TestEncryptedData)
12: MatchingDecision ← Hamming Distance Threshold (Matcher)
13: TrueFakeBiometricData ← DenseNet Features (TrueFakeFingerprint/Iris)
14: DetectedTrueFakeData ← SVM (TrueFakeBiometricData)
15: Client-AuthenticationServer-Matcher:
16: AuthorizationDetermination ← FinalDecisionUnit (Matching Decision, Detected TrueFakeData)

4.2. Flow of Biometric Data at the Client-Side

The inputted data passes through five stages on the client side before being sent out to the cloud. These stages are discussed in details hereunder.

1. Data Preprocessing: The area of interest inside the image taken from the eye, i.e., the iris, needs to be extracted. In this work circular Hough transformation is used to localize the iris and extract it. The segmented iris is then normalized. Fingerprint images remain unchanged.
2. Feature Extraction: The images are fed to a DenseNet that is pre-trained on millions of images from the ImageNet dataset. This massive amount of images included a thousand various classes such as chairs, zebras, apples, monitors, and etc. The concept of transfer learning aids us to use the patterns learnt from these images for the task of biometric verification. Each layer within this deep network contains many patterns that might be useful in representing the inputted image. The output of these layers, also known as off-the-shelf features [22], is taken as the representation of the input, i.e., the feature vector. However, the task of finding the right layer to extract the features from can be arduous. In [23], layers are chosen randomly in order to extract features. In this work the pre-trained DenseNet is coupled with the CNNOptLayer algorithm to find the most optimal layer for feature extraction for each task. This algorithm performs an exhaustive search on the convolutional layers within the network, and uses their output as features. The acquired features result in a verification output and their performance can be measured using the F1 score. The layer with the highest F1 score is chosen as the optimal layer for that specific task. After extraction, the feature vectors are binarized based on the mean of each feature. This binarization allows us to perform hamming distance and use the encryption scheme.

3. Masking the Data: The username and the biographic information (or place of birth) are given to two word based feature extractors. The first extractor finds the index of the first, the middle, and the last element of its word from the dictionary of letters. Then, the ceiling of the index of the first element to the power of the index of the third element is divided by the multiplication of the index of the second element to the power of two on one side and the addition of the index of the first element and the index of the second element on the other side. The output of this function is BioInfoData1. The other feature extractor finds three elements: the length of the birth place word, the frequency of the most repeated character, and the difference between the highest and the lowest indices among the characters in the word. The operation to be performed on these elements is described as the round of the addition of the first element, the second, and the third element divided by three as the base and the ceiling of the first element divided by the third element as the power. The output of this unit is BioInfoData2. These two data are concatenated and repeated until the lengths of the image feature vectors are reached. After getting the final mask, it is XORed with the biometric feature vectors to create the plain data for the encryption. The reason for XORing the mask with the feature vector lies within the fact that feature vectors are binarized and later XORed for comparison. Since the mask generation outputs the same mask for the same individual each time they request for verification, the result of comparing two masked feature vectors of the same individual is equal to that of the comparison of two plain feature vectors. Therefore, masking the binary feature vectors does not change the results of the comparison unit for the same individuals, but highly affects the cases where the vectors come from different individuals.
4. Encryption: The two plain vectors data go into a Paillier Chunkwise encryption scheme [24]. This scheme first up-samples the data, and then encrypts chunks of it using Paillier Encryption. This scheme has two advantages; firstly, Paillier encryption is partially homomorphic and supports addition, which enables us to perform the XOR operation on the authentication server. Secondly, the up-sampling allows the matcher to calculate the hamming distance and recognize if the two feature vectors match or not.
5. True/Fake Detection: The last operation on the client-side to identify the liveness of the presented biometric data. The CNNOptLayer is used to find the optimal layer for feature extraction for this task and a SVM, which is trained on these feature vectors from true and fake datasets, identifies the liveness of the data.

4.3. Decision Making Process

The decision for a claimed identity is made based upon two elements: (1) the result of matcher, and (2) the result of true/fake detection unit. In this work the authentication/matcher architecture which is pivotal for the performance of Paillier Chunckwise is used. The encrypted feature vector is sent to the authentication server along with the username. The original biometric data for that user name is retrieved from the database and is sent to the authentication server. Using the additive homomorphism of Paillier scheme, the two encrypted vectors are XORed. The results of this XOR and tell us how similar these binary vectors were before encryption. This result is sent to the Matcher, which calculated the hamming distance by looking at the up-sampled results. If the resulted hamming distance is smaller than a pre-defined threshold, which will be determined in the next section, the two vectors come from the same user. If the true/fake detection unit identifies the biometric data as live data, and the hamming distance is smaller than the threshold, the user is verified.

5. Experimental Approach and Results

5.1. Experimental Setup

In order to perform the feature extraction, a pre-trained DenseNet is acquired. The used DenseNet is a Keras implementation of this network in Python that supports the TensorFlow backend.

This network has 161 layers with the input size of 224 × 224. The iris images are segmented using Hough transformation. The images are resized to match the input size and are fed to the DenseNet. Using CNNOptLayer, the optimum layer for the tasks of biometrics verification and true/fake detection are identified. The features extracted from these layers are binarized and up-sampled and masked. The final vectors are encrypted using Paillier Chunkwise Encryption written in Matlab. The true/fake detection classifier is written using Scikit-learn's SVM.

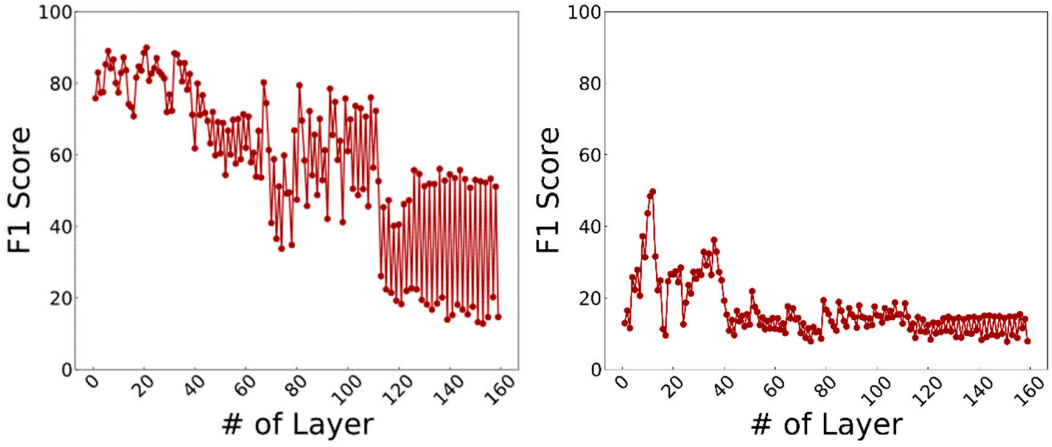
5.2. Dataset Selection

The datasets used in this work are the CASIA fingerprint and iris datasets [25]. Both of these biometric data also have fake version, which were used in the training of the true/fake detection unit. Overall, 165 users were selected with each having five fingerprint and five iris images. The right thumb and the right iris are used for each user and a unique username and place of birth is given to them. Therefore, there are 825 images for each biometric input that can be compared to each other. This yields 339,900 different cases of comparisons for each biometric data type. In the end, the iris and fingerprint vectors are merged to see how well our system works in a bi-modal environment.

The true/fake classifier is trained on data from CASIA's true/fake dataset, with ten-fold cross-validation. Therefore, this system's data is different from the ones used for verification.

5.3. Final Results

Each of the nearly 340,000 comparisons yields in a hamming distance computed by the matcher. If the input vectors match, this distance should be low. After all of the distances are calculated, a threshold that maximizes the F1 score is found and the last F1-score is computed. The F1 score results of choosing each layer of the DenseNet as feature extractor is shown in Figure 8.



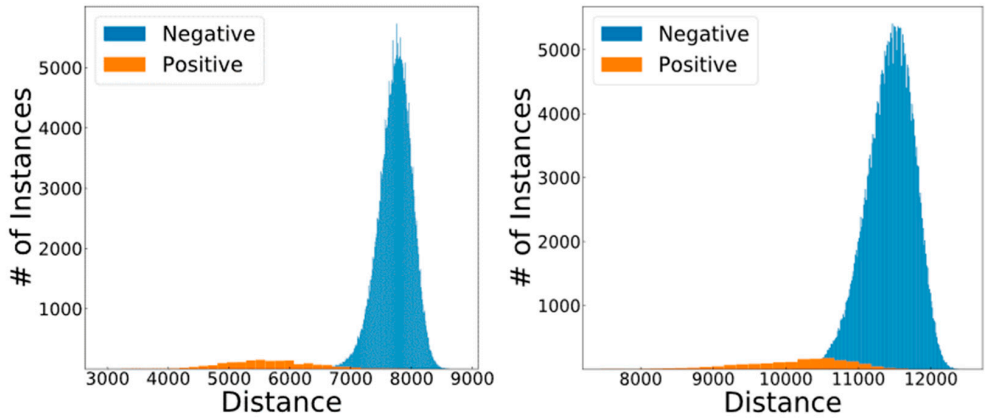
(**a**) (**b**)

Figure 8. The results of each layer of the DenseNet as feature extractor for (**a**) iris and (**b**) fingerprint.

As it is visible from Figure 8, different layers have different capabilities in extracting features. The optimum layer for feature extraction from the iris images is the 21th layer, while the 12th layer gives the best representation for the fingerprints. This is due to the fact that each of these layers holds patterns learnt from the ImageNet dataset and different patterns suit different types of data. Moreover, it can be seen that iris verification achieves a higher F1 score than fingerprint verification, showing that this pre-trained network is more suitable for feature extraction from normalized iris images.

Another important observation is the fact that the performance seems to become better as the layers increase but falls after certain layers. While the first layers hold a simpler abstraction of the input data, the higher layers hold a more complex abstraction. If this abstraction becomes too complex, information is lost and performance is downgraded.

Having found the optimum layers using CNNOptLayer, we take a closer look at the results from these two layers. After the iris features are extracted from the 21th layer and fingerprint features are extracted from the 12th layer, they are binarized. Having done so, the hamming distances between all possible and unique pairs of input images are calculated. In order to do so, one image (e.g., an iris) is taken as the reference, and the rest of the images are compared to it and the hamming distances are recorded. Since there are 5 images taken from each user in the dataset, only 4 other image should ideally match this image and the 820 other users should have a higher hamming distance. In this work, the collection of the distances that are gathered from match cases are called "Positive", while the collection of the distances that are not from the same person's data are called "Negative". This naming can also be thought of as their ground truth verification results. These two distributions are shown in Figure 9.

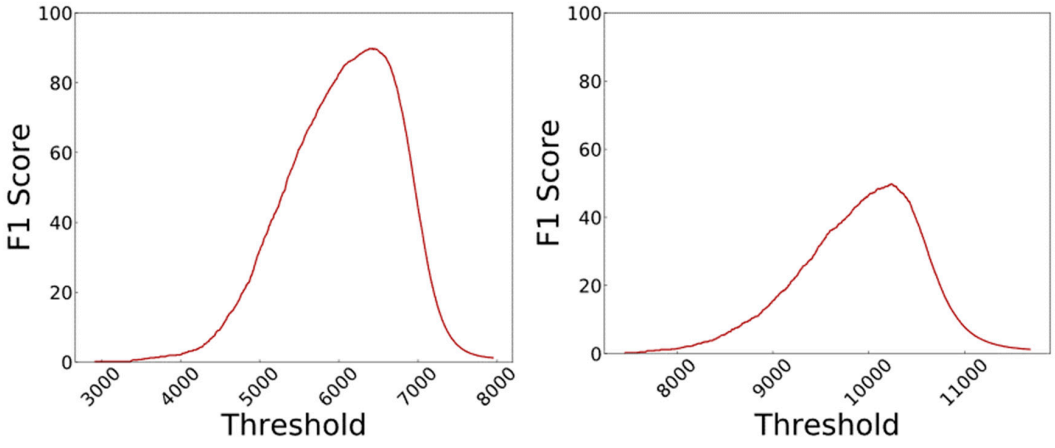


(**a**) (**b**)

Figure 9. The optimum distance distributions of (**a**) iris and (**b**) fingerprint.

As it is visible from Figure 9, the negative instances in average have a higher hamming distance than the positive ones. This is a testament to the fact that the feature extraction is performed correctly. However, these two distributions have less overlap in the iris images than they do in the fingerprint images, showing that the verification of irises is easier for the system that verification of fingerprints.

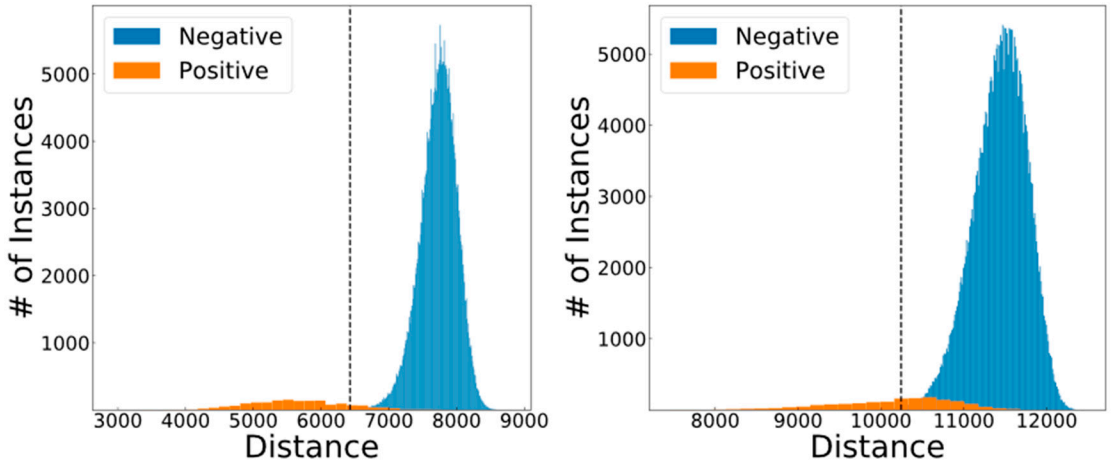
In order to find an optimum threshold, this value is swept across the minimum to the maximum range of the distances and the one with the highest F1 score is chosen. This is reflected in Figure 10.



(**a**) (**b**)

Figure 10. The results of changing the threshold in (**a**) iris and (**b**) fingerprint verifications.

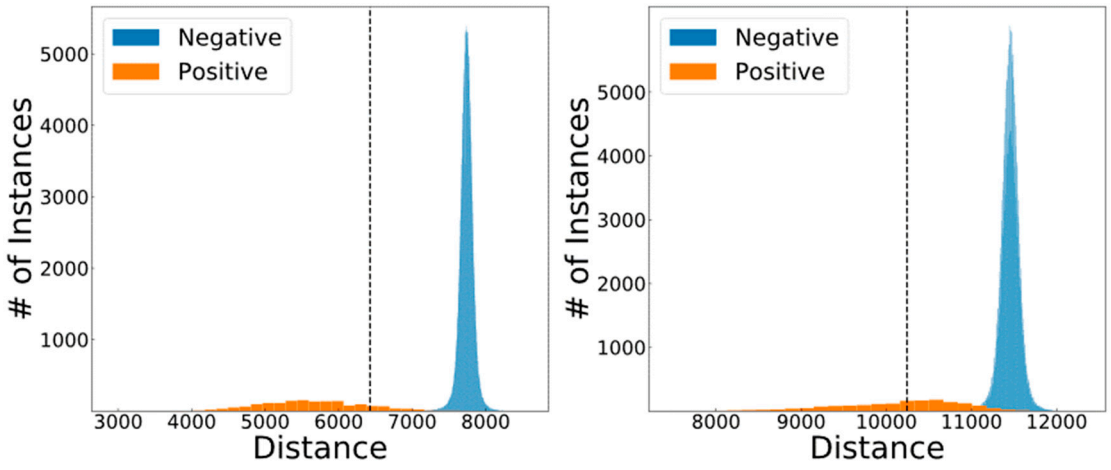
Now that the thresholds are found, we can depict how the decision boundary would look like in this one dimensional space in Figure 11.



(**a**) (**b**)

Figure 11. The optimum distance distributions of (**a**) iris and (**b**) fingerprint with the optimum thresholds.

As it can be seen from Figure 11, there are misclassifications in both cases of verification. In order to alleviate this problem, we take the action of bit-wise masking the data with the mask created from the user input. This masking of the data takes place before up-sampling and encryption. After the masks are applied, the changes in the distributions are depicted in Figure 12.



(**a**) (**b**)

Figure 12. The optimum distance distributions of (**a**) iris and (**b**) fingerprint with the optimum thresholds after masking the data.

As it can be observed from Figure 12, the positive distribution remains unchanged due to having the same masks; however, the negative distribution is pushed further in the distance. This lowers the false positives greatly. In order to not affect the security of the system, the threshold remains unchanged, so that if anyone can gain access to the personal data of a user, they would have no advantage in gaining access to the system.

In order to evaluate the performance of this system in a bi-modal environment, the features of iris and fingerprint images are concatenated and fed to the system as the input features. The result of this combination is visible in Figure 13.

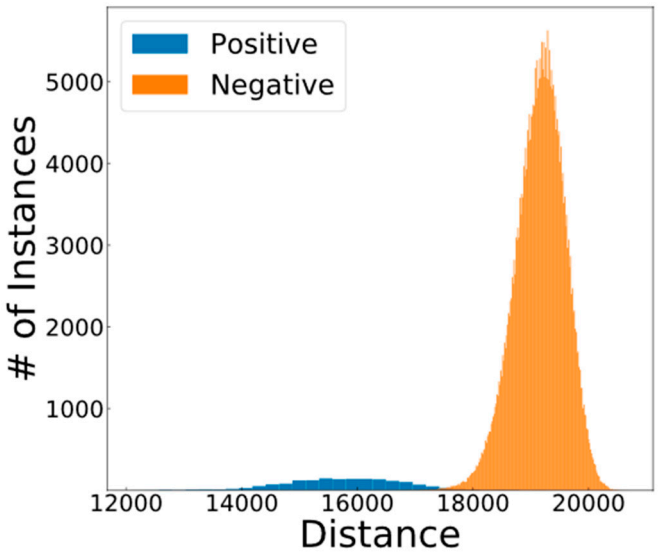
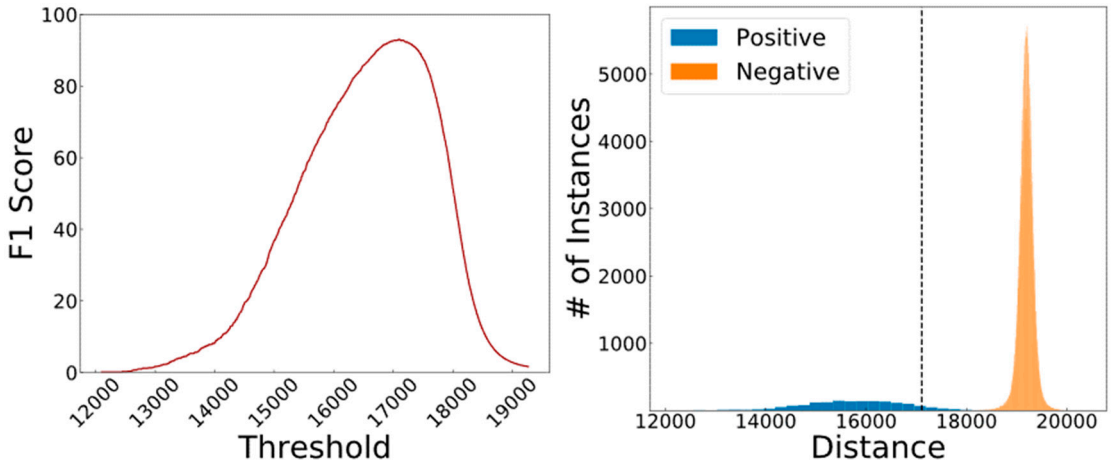


Figure 13. The distance distributions of the combined iris and fingerprint features.

The combined distributions showed in Figure 13 shows that this combination has decreased the overlap between the negative and positive distribution. Therefore, combining the fingerprint and iris data makes the system more accurate. The optimum threshold for this case is found and shown in Figure 14.



(**a**) (**b**)

Figure 14. The results of (**a**) threshold sweeping and (**b**) the optimum distance distributions of the combined iris and fingerprint features.

As for the true/fake detection system, the CNNOptLayer was similarly used in order to find the best features. It proved an easy task for the DenseNet since the 2nd layer's features for the iris and the 69th layer's features for fingerprint yielded 100 percent classification accuracy.

The results are shown in Table 1.

Table 1. The overall results of the verification system with iris and fingerprint inputs (TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative respectively).

Data Type	Masked	Layer	Threshold	TP	TN	FP	FN	F-Score
Fingerprint	No	12	10,243	812	337,450	800	838	49.79
Fingerprint	Yes	12	10,243	812	338,246	4	838	65.86
Iris	No	21	6427	1414	338,165	85	236	89.81
Iris	Yes	21	6427	1414	338,248	2	236	92.24
Combined	No	12 + 21	17,108	1507	338,168	82	143	93.05
Combined	Yes	12 + 21	17,108	1507	338,250	0	143	95.47

As is observed in Table 1, the best result was obtained when the fingerprint and iris features were concatenated. This table also shows the reason behind masking the data. While the data is homomorphically encrypted and does not need masking for privacy, the masking helps the accuracy of the system. Via masking, the false positives that are detrimental to the goal of a verification system are lowered significantly.

5.4. Security Analysis

In order to evaluate the security of the proposed system, a number of locations within the system that the data is unprotected (or plain) are targeted. The points that are targeted in here are the CNN-based feature extractor and the SVM-based classifier within the true/fake detection module. These attacks are carried out on AlexNet which is different from our system and serve the sole purpose of finding vulnerabilities within CNNs.

Besides the aforementioned privacy concerns, there are other possible threats and attacks that can target a biometric recognition system. They can be stated as: (1) attacks on the sensing devices, which are known as direct attacks. These attacks can cause impersonation or evasion of identity. The countermeasure for these attacks operate based on the liveness detection according to which it is assessed that the biometric data is fake or alive. This operation is done based on specific patterns (such as the ones remained from sweating or blinking eyes). (2) attacks on the channels that connect different modules. An attack from this type is called man-in-the-middle attack according to which an original image is replaced with a new synthetic image. (3) attacks to the processing modules and algorithms. (4) attacks to the template database. (5) fabricating a fake biometric trait to mimic an enrolled client, which is called spoofing attack. (6) Feeding stolen data of the victim to the feature extractor. (7) attacks on the feature extractor. (8) attacks in the matcher. (9) attacks on the template database. (10) suffering of deep neural networks within the system from unexpected instabilities and performing misclassification on data instances created by adversaries through adding invisible and small disorder to the originally recognized data. Also, the extracted features from them may be vulnerable to mimicking and synthetically image production. Another security issue is leakage of essential information from a trained network model.

The proposing attacks for the feature extractor point to the fully connected layer, shown in Figure 15.

The attacks are: (AX) substituting every "odd" element of the Bias vector of the seventh fully connected layer with its next "even" element in the vector ($b[2n+1] = b[2n]$, for "n" starting from zero). (BX) substituting every thirty-two elements of the Bias vector of the seventh fully connected layer with their average value ($b[n : n+31] = Average(b[n : n+31])$, n = 1, 33, 66, ...). (CX) substituting every thirty-two elements of the Bias vector of the seventh fully connected layer with their minimum value ($b[n : n+31] = Minimum(b[n : n+31])$, n = 1, 33, 66, ...). (DX) substituting every "odd" element in eight columns of the Weight matrix (with the interval size of 512) of the seventh fully connected layer with its next "even" element in the vector ($W[n, K] = W[2n, K]$, for "n" starting from zero and "K" = 1, 513, 1026, etc.). (EX) substituting every element of the Weight matrix of the seventh fully connected layer with their mean/median/minimum/mode value ($W[:] = Average/Median/Minimum(W[:])$). (FX) Flipping

the Weight matrix of the seventh fully connected layer up-side and down-side ($W = FlipUpDown(W)$). (GX) changing the layer for feature extraction from the seventh fully connected layer to the second convolution layer ($y_{FC7} = y_{CONV2}$).

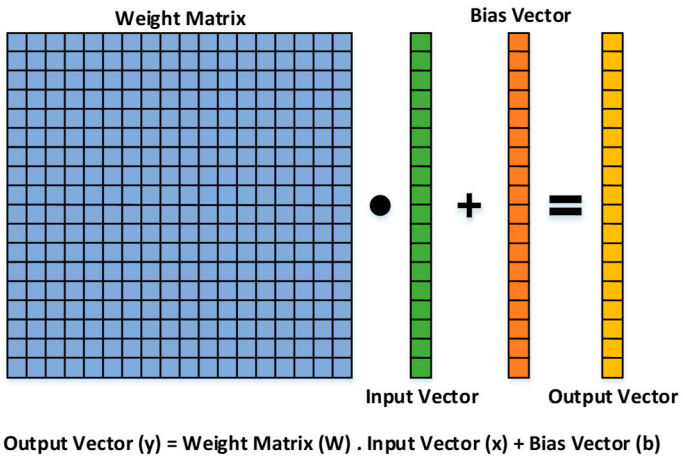


Figure 15. The architecture and formula for the fully connected layer.

(HX) performing an insider attack by targeting the classifier and through: (i) manipulation of 12.5% of the Training labels; (ii) manipulation of 25% of the Training labels; (iii) manipulation of 50% of the Training labels. The equation of the SVM classifier is shown to the following. According to this equation, s represents the support vectors, α represents the weights, b represents the bias, x represents the input vector, k represents the kernel function (i.e., it can be a dot product for a linear kernel), c represents the group type. If $c \geq 0$, then the input vector belongs to the first class, otherwise it belongs to the second class. According to the mentioned attack, the support vectors are manipulated to cause misclassification of the input data.

$$c = \alpha_i \times k \times (s_i, x) + b$$

(IX) manipulation of neurons according to which: (I) positively rectify the outputs of a number of neurons during the verification phase (e.g., $|y[i]|$, $i \in random\ numbers$); (II) forcing the outputs of a number of neurons to zero value (power gating neurons) during the verification phase (e.g., $y[i] = 0$, $i \in random\ numbers$); and (III) negatively rectify the outputs of a number of neurons during the verification phase (e.g., $-|y[i]|$, $i \in random\ numbers$).

Among the proposed attacks, only a number of them were applicable and led to observable negative impact on the system. The attack number 7 causes degradation of TPR from 1.0 to 0.17 (-83%) and TNR from 1.0 to 0.79 (-21%) when the fingerprint is input into the system. This attack means the unauthorized access and denial of service. The attack number 5 causes degradation of TPR from 1.0 to 0.0 and FNR when the substitution value is mean or median. On the other hand, when the substitution value is minimum or mode then TNR is degraded from 1.0 to 0.0. The first case means denial of service and the second case means unauthorized access. Regarding the iris biometric data, the attack number 7 causes a complete denial of service. The attack number 5 delivers the same results as what were delivered for the fingerprint data. All these attacks are performed after the user enrollment process.

There are three components in our system that collectively defend against these attacks. Firstly, the matching unit calculates the hamming distance between the original and the new feature vectors and enables the task of verification to take place. Secondly, the masking unit, which requires input from the user and only works correctly if the inputted words are the same as the original words. Lastly the true and fake detection unit checks if the input if received from a live individual or if it is synthetized. As we have shown, this unit is the most vulnerable unit is the system and can be manipulated to detect fake inputs as true. If the true/fake detection unit is compromised at the user side, this manipulation is only effective on the output of the SVM in the user side and is not effective on the results of the matcher

in the cloud. Therefore, in the worst case scenario where the attacker has the username and the place of birth of the user (i.e., can generate the correct mask), has fake biometric data, and manipulates the true/fake detection unit to output true, the result of the matcher in the cloud remains the same and the attacker is not verified.

In the literature, a number of defenses have been proposed for the attacks mentioned in (1) to (10) [26–28]. In [26,27], it is proposed to combine cryptography and biometric security in order to design a stronger authentication system. The authors in [28] discussed many different defenses for biometric recognition systems, including risk-based approach, systems and security architecture, defensive measures, challenge/response, retention of data, randomizing input biometric data, liveness detection, multiple biometrics, multi-modal biometrics, multi-factor authentication, soft biometrics, signal and data integrity and identity, cryptography and digital signatures, template integrity, cancellable biometrics, hardware integrity, network hygiene, physical security, activity logging, policy, and compliance checking.

There is another option for countering this attack that is protecting the deep neural network. Meanwhile, a number of techniques have been proposed for protection of neural networks may help in correcting the operation of the true/fake detection system. It means protecting the neural network-based feature extractor. In [29], an effective defense against backdoor attacks on neural networks has been proposed. The defense is called fine-pruning that is a combination of pruning and fine-tuning. This defense is capable of weakening or even eliminating the backdoors (with a specified success rate). Another work [30] proposes a novel approach for backdoor detection and removal from neural networks. This method is able to detect poisonous data as well as repairing the model. A robust and generalizable detection and mitigation system for detection of backdoor attacks for neural networks has been presented in [31]. This technique can identify backdoors and reconstruct possible triggers. The technique includes input filtering, neuron pruning and unlearning.

The SVM classifier can be defended and protected as well. The following techniques have been proposed to defend a classifier against possible attacks, which can be integrated into our model. In [32], an optimization framework has been proposed that is able of finding the label flips for the purpose of maximizing the classification error. The authors in [33] presented a strategy for improving the SVM robustness in front of input data manipulation based on a simple kernel matrix correction. [34] shows an adversary-aware design of SVMs based on real-world security problems. A method has been proposed in [35] according to which the classification model as well as the training procedure are not modified and it can be used to defend against many attacks. Using this defense, the distribution of clean and manipulated features can be modeled in order to enhance the SVM performance in classification.

5.5. Discussion and Future Research

The usage of one pre-trained neural network as feature extractor for multiple tasks in this work showed the flexibility of deep learning. While DenseNet was trained on images of everyday objects, the patterns learnt within proved to be useful in extracting features from both iris and fingerprint images in both tasks of verification and liveness detection. Observing the performance delivered by different layers' features in Figure 9 gave us insight about how the information that is valuable for the given task propagates through the network and at what layer the abstracted information becomes the most valuable. This Figure which is the heart of the CNNOptLayer algorithm can be derived in other tasks that contain transfer learning. While it is common practice to use the last layers of a pre-trained neural network, in this work we observed that it can be detrimental to do so, and used the algorithm to find the best layer.

One of the downsides of our work was the low F1 score on fingerprint data. This is due to the fact that the needed patterns might not exist or be dominant in the pre-trained neural network. For future research, one can train a specified neural network to learn verification of iris and fingerprint data in a multitasking manner. This network then might yield better results, having seen a more similar data distribution. The reason we avoided doing so was to not save any data from the biometric data

distribution inside the model, gaining a zero knowledge system that is scalable with no need to train on sensitive data. However, one can also train a network on generic fingerprint/iris dataset and test it on the sensitive data. Overall, a trade-off was observed between preserving the security of the data and the performance of the system.

6. Conclusions

In this work, a privacy-preserving cloud-based and multiple-party biometric verification system has been proposed which relies on one pre-trained deep neural network to perform feature extraction. Via using transfer learning, the achieved system was able to extract features from iris and fingerprint images for the tasks of biometric verification and true/fake detection. This enabled usage of a neural network that required no knowledge on sensitive data and scalability of the system when new users are added. Optimization of this process was done using a novel algorithm called CNNOptLayer which found the optimum layer for each task and input data type. The biometric features were bit-masked and encrypted using Paillier Chunckwise. This homomorphic encryption allowed the biometric data to remain encrypted outside of the user side and preserved the privacy of the user. Overall, the system was able to achieve a verification F1 score of 95.47% when verifying the combined features of iris and fingerprint inputs with zero false positives.

Author Contributions: M.S. contributed in coming up with the ideas, running the experiments, and writing the manuscript. S.T. contributed in coming up with the ideas and writing the manuscript. J.-S.Y. provided technical feedbacks and reviewed the manuscript. All authors read and confirmed the final version of manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gentry, C.; Boneh, D. *A Fully Homomorphic Encryption Scheme*; Stanford University: Stanford, CA, USA, 2009.
2. Khan, M.A.; Akram, T.; Sharif, M.; Javed, M.Y.; Muhammad, N.; Yasmin, M. An implementation of optimized framework for action classification using multilayers neural network on selected fused features. *Pattern Anal. Appl.* **2018**. Available online: https://doi.org/10.1007/s10044-018-0688-1 (accessed on 30 November 2018). [CrossRef]
3. Mahmood, Z.; Muhammad, N.; Bibi, N.; Ali, T. A Review on State-of-the-Art Face Recognition Approaches. *Fractals* **2017**, *25*, 1750025. [CrossRef]
4. Ene, A.; Togan, M.I.; Tom, S.-A. Privacy Preserving Vector Quantization Based Speaker Recognition System. *Proc. Rom. Acad. Ser. A* **2017**, *158*, 371–380.
5. Bommagani, A.S.; Valenti, M.C.; Ross, A. A Framework for Secure Cloud-Empowered Mobile Biometrics. In Proceedings of the 2014 IEEE Military Communications Conference, Baltimore, MD, USA, 6–8 October 2014; pp. 255–261.
6. Toli, C.-A.; Aly, A.; Preneel, B. Privacy-Preserving Multibiometric Authentication in Cloud with Untrusted Database Providers. *undefined* 2018. Available online: https://www.semanticscholar.org/paper/Privacy-Preserving-Multibiometric-Authentication-in-Toli-Aly/d37472309ba3a28b66646e92d239ffedb85f2abb (accessed on 30 November 2018).
7. Weng, L.; Amsaleg, L.; Morton, A.; Marchand-Maillet, S. A Privacy-Preserving Framework for Large-Scale Content-Based Information Retrieval. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 152–167. [CrossRef]
8. Talreja, V.; Valenti, M.C.; Nasrabadi, N.M. Multibiometric secure system based on deep learning. In Proceedings of the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, 14–16 November 2017; pp. 298–302.
9. Zhou, K.; Ren, J. PassBio: Privacy-Preserving User-Centric Biometric Authentication. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 3050–3063. [CrossRef]
10. Toli, C.-A.; Preneel, B. Privacy-preserving Biometric Authentication Model for e-Finance Applications. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Portugal, 22–24 January 2018; pp. 353–360.

11. Toli, C.-A.; Aly, A.; Preneel, B. A Privacy-Preserving Model for Biometric Fusion. In *International Conference on Cryptology and Network Security*; Springer: Cham, Switzerland, 2016; pp. 743–748.
12. Abidin, A. On Privacy-Preserving Biometric Authentication. In *International Conference on Information Security and Cryptology*; Springer: Cham, Switzerland, 2017; pp. 169–186.
13. Taheri, S.; Yuan, J.-S. A Cross-Layer Biometric Recognition System for Mobile IoT Devices. *Electronics* **2018**, *7*, 26. [CrossRef]
14. Erkin, Z.; Franz, M.; Guajardo, J.; Katzenbeisser, S.; Lagendijk, I.; Toft, T. Privacy-Preserving Face Recognition. In *International Symposium on Privacy Enhancing Technologies Symposium*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 235–253.
15. Huang, Y.; Malka, L.; Evans, D.; Katz, J. Efficient Privacy-Preserving Biometric Identification. Available online: http://mightbeevil.com/secure-biometrics/ndss-talk.pdf (accessed on 30 November 2018).
16. Osadchy, M.; Pinkas, B.; Jarrous, A.; Moskovich, B. SCiFI—A System for Secure Face Identification. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, USA, 16–19 May 2010; pp. 239–254.
17. Yuan, J.; Yu, S. Efficient privacy-preserving biometric identification in cloud computing. In Proceedings of the 2013 Proceedings IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 2652–2660.
18. Wang, Q.; Du, M.; Chen, X.; Chen, Y.; Zhou, P.; Chen, X.; Huang, X. Privacy-Preserving Collaborative Model Learning: The Case of Word Vector Training. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 2381–2393. [CrossRef]
19. Niu, X.; Ye, Q.; Zhang, Y.; Ye, D. A Privacy-Preserving Identification Mechanism for Mobile Sensing Systems. *IEEE Access* **2018**, *6*, 15457–15467. [CrossRef]
20. Zhu, L.; Zhang, C.; Xu, C.; Liu, X.; Huang, C. An Efficient and Privacy-Preserving Biometric Identification Scheme in Cloud Computing. *IEEE Access* **2018**, *6*, 19025–19033. [CrossRef]
21. Ma, Y.; Wu, L.; Gu, X.; He, J.; Yang, Z. A Secure Face-Verification Scheme Based on Homomorphic Encryption and Deep Neural Networks. *IEEE Access* **2017**, *5*, 16532–16538. [CrossRef]
22. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features off-the-shelf: An Astounding Baseline for Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 806–813.
23. Nguyen, K.; Fookes, C.; Ross, A.; Sridharan, S. Iris Recognition with off-the-Shelf CNN Features: A Deep Learning Perspective. *IEEE Access* **2018**, *6*, 18848–18855. [CrossRef]
24. Penn, G.M.; Pötzelsberger, G.; Rohde, M.; Uhl, A. Customisation of Paillier homomorphic encryption for efficient binary biometric feature vector matching. In Proceedings of the 2014 International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany, 10–12 September 2014.
25. Biometrics Ideal Test. Available online: http://biometrics.idealtest.org/dbDetailForUser.do?id=7 (accessed on 4 December 2018).
26. Popa, D.; Simion, E. Enhancing security by combining biometrics and cryptography. In Proceedings of the 2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Targoviste, Romania, 29 June–1 July 2017; pp. 1–7.
27. Tams, B. Attacks and Countermeasures in Fingerprint Based Biometric Cryptosystems. *arXiv* **2013**, arXiv:1304.7386.
28. Roberts, C. Biometric attack vectors and defences. *Comput. Secur.* **2007**, *26*, 14–25. [CrossRef]
29. Liu, K.; Dolan-Gavitt, B.; Garg, S. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. *arXiv*, 2018; arXiv:1805.12185.
30. Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; Srivastava, B. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. *arXiv*, 2018; arXiv:1811.03728.
31. Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; Zhao, B.Y. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. Available online: https://people.cs.vt.edu/vbimal/publications/backdoor-sp19.pdf (accessed on 30 November 2018).
32. Xiao, H.; Xiao, H.; Eckert, C. Adversarial label flips attack on support vector machines. In Proceedings of the 20th European Conference on Artificial Intelligence, Montpellier, France, 27–31 August 2012; pp. 870–875.
33. Biggio, B.; Nelson, B.; Laskov, P. Support Vector Machines Under Adversarial Label Noise. In Proceedings of the Asian Conference on Machine Learning, Taoyuan, Taiwan, 13–15 November 2011; pp. 97–112.

34. Biggio, B.; Corona, I.; Nelson, B.; Rubinstein, B.I.; Maiorca, D.; Fumera, G.; Giacinto, G.; Roli, F. Security Evaluation of Support Vector Machines in Adversarial Environments. In *Support Vector Machines Applications*; Springer International Publishing: Cham, Switzerlane, 2014; pp. 105–153.
35. Samangouei, P.; Kabkab, M.; Chellappa, R. Defense-GAN: Protecting Classifiers against Adversarial Attacks Using Generative Models. *arXiv* **2018**, arXiv:1805.06605.