T.C.

ALTINBAŞ UNIVERSITY

Information Technologies

TAMPER DETECTION IN MULTIMODAL BIOMETRIC TEMPLATES USING FRAGILE WATERMARKING AND ARTIFICIAL INTELLIGENCE

Fatima Ismail Ali ABU SIRYEH

Master Thesis

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2021

TAMPER DETECTION IN MULTIMODAL BIOMETRIC TEMPLATES USING FRAGILE WATERMARKING AND ARTIFICIAL INTELLIGENCE

by

Fatima Ismail Ali ABU SIRYEH

Information Technologies

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “Tamper Detection in Multimodal Biometric Templates Using Fragile Watermarking and Artificial Intelligence” prepared and presented by “Fatima Ismail Ali ABU SIRYEH” was accepted as a Master of Science Thesis in Department.

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Dr. Muhammad ILYAS	School of Engineering and Natural Sciences, Altinbas University	_____
Asst. Prof. Dr. Sewale Musadaq TAHA	Faculty of Communication Beykent University	_____

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Fatima Ismail Ali ABU SIRYEH

.

ACKNOWLEDGMENTS

I would like to acknowledge my indebtedness and render my warmest thanks to my supervisor, Asst. Prof. Dr. Abdullahi Abdu IBRAHIM, who made this work possible. His friendly guidance and expert advice have been invaluable throughout all stages of the work.

Special thanks are due to my parents as well as my brother, for their continuous support and understanding and their constant encouragement.

I would like to thank all my friends who supported me to achieve my goal and gave me valuable suggestions which have contributed the improvement of the thesis.

I would like to thank all the staff of the Computer Engineering Department, Altinbas University, for their ideas and advice that helped me to complete my thesis.

ABSTRACT

TAMPER DETECTION IN MULTIMODAL BIOMETRIC TEMPLATES USING FRAGILE WATERMARKING AND ARTIFICIAL INTELLIGENCE

ABU SIRYEH, Fatima Ismail Ali

M.Sc., Information Technologies, Altınbaş University

Supervisor: Ass. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: June/2021

Pages: 58

Significant attention has been brought towards biometric authentication in recent years, according to the rapidly growing demand for secure and reliable authentication methods. To improve the accuracy of biometric authentication systems, multiple templates are collected from each candidate, for different biometrics. Accordingly, any false positives or false negatives that may occur when matching one of the templates are rectified when matching the second one, and vice versa. Moreover, to reduce the overhead in the size of the data required to represent these templates, as well as secure these templates, recent studies use digital watermarking techniques to embed one of the templates, or its features, into the other. Fragile watermarking is widely used for this purpose, according to the ability of detecting any tampering when the watermark information is lost. However, this fragility denies the ability to further reduce the size of the data by compressing the image. Thus, a new watermarking approach is proposed in this study, in which the features extracted from the iris template are embedded in the face template of the candidate. The features of the iris are extracted using an autoencoder Artificial Neural Network (ANN), then, embedded in the DCT coefficients of the face image, after being quantized and before being compressed using Huffman Coding, based on the Joint Photographic Experts Group (JPEG) standard. The watermark information is embedded using the Least-Significant-Bit (LSB) watermarking method, where each value is calculated based on the corresponding value in the iris template

and the DCT coefficient in the face template, and encrypted using Arnold Transform (AT). Accordingly, the watermark information is lost when the compressed watermarked image is attacked. Additionally, as the watermark information does contain similar patterns, but never identical even for the same candidate, an ANN is used to investigate the existence of these patterns in the extracted watermark, instead of comparing it to a static array. The proposed method has achieved 100% tamper detection rate with 0.05% reduction in iris recognition accuracy while maintaining the face recognition accuracy intact.

Keywords*:* Biometric Authentication, Artificial Neural Networks, Digital Watermarking, Image Compression, Artificial Intelligence.

ÖZET

KIRILGAN DAMGALAMA VE YAPAY ZEKA KULLANAN ÇOK MODLU BIYOMETRIK ŞABLONLARDA KURCALAMA TESPITI

ABU SIRYEH, Fatima Ismail Ali

Yüksek Lisans, bilgi Teknolojisi, Altınbaş Üniversitesi

Danışman: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: Haziran/2021

Sayfalar: 58

Güvenli ve güvenilir kimlik doğrulama yöntemlerine yönelik hızla artan talebe göre, son yıllarda biyometrik kimlik doğrulamaya büyük ilgi gösterildi. Biyometrik kimlik doğrulama sistemlerinin doğruluğunu artırmak için, farklı biyometrikler için her adaydan birden fazla şablon toplanır. Buna göre, şablonlardan biri eşleştirilirken oluşabilecek yanlış pozitifler veya yanlış negatifler, ikincisi eşleştirilirken düzeltilir ve bunun tersi de geçerlidir. Ayrıca, bu şablonları temsil etmek için gereken verilerin boyutundaki ek yükü azaltmak ve bu şablonları güvenceye almak için, son çalışmalar, şablonlardan birini veya özelliklerini diğerine gömmek için dijital damgalama tekniklerini kullanır. Kırılgan damgalama, filigran bilgisi kaybolduğunda herhangi bir kurcalamayı algılama yeteneğine göre bu amaç için yaygın olarak kullanılmaktadır. Bununla birlikte, bu kırılganlık, görüntüyü sıkıştırarak verilerin boyutunu daha da küçültme yeteneğini reddeder. Bu nedenle, iris şablonundan çıkarılan özniteliklerin adayın yüz şablonuna gömüldüğü bu çalışmada yeni bir damgalama yaklaşımı önerilmiştir. İrisin özellikleri, bir otomatik kodlayıcı Yapay Sinir Ağı (ANN) kullanılarak çıkarılır, daha sonra, ortak Fotoğraf Uzmanları Grubuna (JPEG) dayalı olarak, nicelleştirildikten sonra ve Huffman Kodlaması kullanılarak sıkıştırılmadan önce yüz görüntüsünün DCT katsayılarına gömülür. standart. Filigran bilgisi, her değerin iris şablonundaki karşılık gelen değere ve yüz şablonundaki DCT katsayısına göre hesaplandığı ve Arnold Dönüşümü (AT) kullanılarak şifrelendiği Least-Significant-Bit (LSB) filigran yöntemi kullanılarak gömülür. Buna göre, sıkıştırılmış filigranlı görüntü saldırıya uğradığında filigran bilgisi kaybolur. Ek olarak, filigran bilgileri benzer desenler

içerdiğinden, ancak aynı aday için bile asla aynı olmadığından, statik bir diziyle karşılaştırmak yerine, çıkarılan filigrandaki bu kalıpların varlığını araştırmak için bir YSA kullanılır. Önerilen yöntem, yüz tanıma doğruluğunu bozulmadan korurken, iris tanıma doğruluğunda %0.05 azalma ile %100 kurcalama algılama oranı elde etmiştir.

Anahtar Kelimeler: Biyometrik Kimlik Doğrulama, Yapay Sinir Ağları, Dijital Filigran, Görüntü Sıkıştırma, Yapay zeka.

TABLE OF CONTENTS

Page

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

ML : Machine Learning

SVM : Support Vector Machine

JPEG : Joint Photographic Experts Group

PIN : Personal Identification Number

LSB : Least-Significant-Bit

DCT : Discrete Cosine Transform

AT : Arnold Transform

Tanh : Hyperbolic Tangent

ReLU : Rectified Linear Unit

SSE : Sum of Squared Error

IIT : Indian Institute of Technology

GPU : Graphical Processing Unit

SSIM : Structural Similarity Index

PSNR : Peak Signal to Noise Ratio

1. INTRODUCTION

The need for access management schemes to restrict access to certain digital systems and information is growing rapidly, according to the growing importance of such systems and information. The vulnerability of traditional secret-based methods, which requires the definition of a secret per each user, e.g., a password or a pattern, towards simple attacks, such as shoulder sniffing and guessing, has limited the application of such methods in restricting access to sensitive information and services [1, 2]. Alternatively, biometric authentication has emerged as a substitute to these methods, according to the higher difficulty of replicating the features that are extracted from biometric templates [3].

With their two main categories, physiological and behavioral, biometric authentication features are extracted either from the physical characteristics of a certain body part, e.g., fingerprint, or from the certain way a human executes a certain task, e.g., running. The difficulty of replicating these features, even if they are compromised, especially with the existence of anti-spoofing techniques, such as liveness detection, these features have shown better security when used to authenticate or recognize users. However, according to the higher robustness of the features collected from physiological features, i.e., the same features are extracted from the same candidate per each authentication process, has brought more attention towards this type of biometric features, compared to the use of behavioral features.

Despite the robustness of such features, and according to the high sensitivity of the services and information being protected using biometric authentication systems, multimodal biometric authentication, which relies of features that are extracted from multiple biometric templates, are being used to reduce the false positives and false negatives that may occur when using a single template [4, 5]. However, such an addition requires additional overhead to the size of the data being communicated to authenticate a candidate, as two templates, or features extracted from two different templates, are required for the authentication. For this purpose, recent studies rely on using digital watermarking techniques to embed one of the templates, or its features, over the other. The use of such approach allows the reduction of the size of the transferred data, as the information of the second template are embedded in the first one, in addition to the ability of securing the transmitted data during transportation, to avoid any manipulation [6-9].

Digital watermarking techniques are mainly categorized into two main categories, robust and fragile. In robust methods, the embedded information is maintained in the watermarked image even when the image is attacked. Hence, this type of watermarking is widely used to prove ownership of the watermarked image, in addition to its use to detect the regions in which attacks are executed against an image. Alternatively, the information that is embedded in an image using fragile watermarking techniques normally disappear as soon as the watermarked image is manipulated, which allows the use of these techniques for tamper detection. Both types of digital watermarking techniques are being used for multimodal biometric authentication, where robust methods are used to prove the authenticity of the image, by proving the owner of the image, which is the sending device, whereas fragile techniques are being used for tamper detection, by investigating the existence of these watermarks in the images [10-12].

Normally, the use of fragile watermarking for tamper detection requires the use of a static image, i.e., a logo, as a watermark, so that, the extracted watermark is compared against the predefined watermark and consider the image to be tampered with if they do not match. However, when one of the biometric templates is used as the watermark, and according to the changing nature of these templates, tamper detection has become a more complex task, which requires the use of Machine Learning (ML) techniques. The extracted watermarked is analyzed using ML to detect patterns that create the intended template, or its features, in the extracted watermark. For instance, if the face image is used as a watermark, the method proposed in [13] uses a Support Vector Machine (SVM) classifier to detect the existence of a face image only, i.e. does not investigate whose face is this. If a face is detected, then the received data is legit and the authentication process can be started. Accordingly, the use of such an approach allows maintaining the required reduction in the data size while maintaining the ability to detect tampering, unlike the use of such an approach with robust watermarking, as these watermarks are maintained or partially lost when a part of the image is attacked.

Despite the reduction in the size of the data being used to represent the biometric templates, as one of the templates is watermarked over the other, the watermarked templates that are produced using fragile watermarking cannot be further compressed, as the watermark information is lost. Further reduction in the size of the data can be achieved by using compression techniques, lossy and lossless. Although the visual features in the templates, or

images, are maintained intact in lossless techniques, the use of lossy compression techniques, such as the Joint Photographic Experts Group (JPEG) format, can provide significant reduction in the size of the data required to represent the visual features in the image, while maintaining high similarity of these features with the original, uncompressed, image. However, the loss of the watermark information, as a result of the fragility of the watermarking technique being used, and as lossy compression is considered as manipulation to the exact pixel values of the image, the use of image compression techniques for further improvement of the data efficiency is limited.

1.1. PROBLEM STATEMENT

The use of multimodal biometric authentication has brought attention towards the employment of watermarking techniques to embed one of the templates over the other, to reduce the overall size of the data required to represent these templates. With the use of fragile watermarking, the biometric templates are also secured against any tampering, during storage or communications. However, the use of such an approach eliminates the ability to further compress the image, using lossy compression techniques, which has the ability to significantly reduce the size of the data required to represent the watermarked image while maintaining the visual feature similar to the original, uncompressed, image. Additionally, the amount of distortion that the watermark imposes to the cover image depends on the size of the data that represents the watermark information. Hence, the reduction of the size of the watermark information can result in improving the quality of the biometric template that is being stored in the cover image. However, it is important to extract robust and efficient features from the biometric template that is being used as the watermark, in order to ensure less distortion to the cover image and maintain the ability to detect any tampering in the watermarked image, in addition to the use of the watermark information for the multimodal authentication stage. Addressing these issues can significantly improve the performance of multimodal biometric authentication systems.

1.2. AIM OF THE STUDY

In order to improve the efficiency of compressing the biometric templates, a new watermarking technique is proposed in this study, in which the watermark information is added to the compressed cover image, instead of the image before being compressed. This approach ensures to further reduce the size of the data required for multimodal authentication, while maintaining the ability to detect any tampering with the templates. The sizes of the produced files are then investigated in order to illustrate the improvement in the compression rate, compared to communicating the templates in full size, as well as evaluating the accuracy of detecting tampering in the received images. This tampering is detected using an artificial neural network, as the patterns in the watermark information are consistent among a certain type of biometric template but neve identical, even for the same individual. Additionally, the influence of the proposed method on the visual feature in the cover images is investigated by measuring their influence on the recognition accuracy of the biometric authentication system.

1.3. THESIS LAYOUT

Following this chapter, the thesis is organized as follows:

i. In chapter two, existing techniques that are being employed in the proposed method are described in details, in addition to their applications in existing methods.

ii. Chapter three describes, in details, the methods that are proposed in this study and how they utilize the existing techniques to meet the aim of the thesis and how they can be integrated in existing multimodal biometric authentication systems.

iii. In chapter four, the experiments that are conducted to evaluate the proposed method and illustrate its benefits are described in details, in addition to the results collected from these experiments. This chapter also discusses the results and illustrated the findings in the conducted experiments, in addition to comparing the performance of authentication systems when using the proposed method to existing state-of-the-art methods.

iv. The conclusion of the study and recommendations for future works are presented in chapter five.

2. LITERATURE REVIEW

2.1. AUTHENTICATION

Authentication is the process of distinguishing legitimate users from intruders, by comparing the authentication credentials provided by the user to those stored for the legitimate users [7]. Authentication techniques are categorized into different categories, based on the credentials provided by the user to authenticate into the device, or the service, which are [14]:-

i. **Something you know:** it is known as well, secret based authentication, its usage involves predefined secrets for the legitimate users, such as Personal Identification Number (PIN) or patterns. PINs are numerical secrets predefined by the user, so that if the number provided to the authentication technique matches the one stored for the legitimate user, the user is considered as a legitimate user and authenticated into the device. Passwords are alphanumerical values, which may consist of number, letters or both **[15]**. The main drawback of secret-based authentication is its vulnerability to simple attacks, such as shoulder surfing, especially that users tend to use easy-to-remember secrets, where such secrets are easy to predict as well [16].

ii. **Something you have:** in this method of authentication, the authenticity of the legitimate user is proven by providing physical objects belonging to the legitimate user such as an ATM card [26], Smartcards or electronic tokens.

iii. **Something you are:** this authentication technique rely on biometric features of the user, such as the fingerprint and handwriting [17, 18], which can also categorized into two categories:

   a. Physiological biometric: Physical biometrics features are extracted from the physical characteristics of a certain part of the human body, such as the iris and fingerprint [9]. Although physical biometric features have shown higher resistance to simple attacks, which has encouraged many manufacturers to embed sensors that extract these features from the users, many of the users have shown concerns about their privacy while using such authentications techniques. Some of the widely used physical

biometric features are the fingerprint, iris and facial recognition, where even knowing the type of features used by the authentication technique does not allow the intruder to access the device, as these features are unique per each person

b. Behavioral biometric: behavioral biometric features do not rely on physical characteristics of the user; they are extracted by collecting information about the unique way that a user performs a certain task. Although the behavior of one individual is different from the behavior on another, the behavior of a certain individual is not as static as the characteristics of the body of that individual, which are used to extract physiological biometric features. Thus, direct comparisons of the features extracted from the individual's behavior and the template features, extracted for the same individual upon setting up the authentication, are not applicable as they are in physiological features. For this reason, it is important to use machine learning techniques in order to make the decisions whether the features extracted from the authentication attempt are for the same individual that registered the template features during the setup of the authentication method [19, 20].

2.2. FACE RECOGNITION

Many of the recent real-life applications, such as entertainment, law enforcement, information security and criminal investigation, have started to require the recognition of humans based on the visual features in their faces, similar to the human approach of recognizing people. These methods are required to provide accurate decision despite the changing environmental conditions, such as lighting, and the changing nature of human faces, such as the use of accessories and differences in hair styles. Accordingly, for efficient face recognition, it is important to produce descriptors that are irrelevant to these conditions, so that, the produced values are always similar for the same individual [21-23].

In early stages of human recognition based on their facial features, the distances between the key points are measured and used to produce a descriptor, manually, even before the

employment of computers in such a task [24]. This method relies on generating a 16-value descriptor for each face, which is then used to match it with any other descriptor by simply measuring the Euclidean distance between the descriptors. These descriptors are considered for the same individual if the calculated distance is found less than a threshold value, otherwise, the templates are considered from different individuals, i.e. do not match.

The method proposed by Wiskott et al. [25] uses Gabor wavelet transform to generate a labeled graph for each face image, based on the key parts of the face image in the template. Compared to other methods from the same era, e.g. [26], this method has shown significantly better performance by producing very accurate descriptors. However, with the rapid developing in computers and the use of computer vision techniques, the need for manual annotation for each template, i.e. positioning of the key parts, has become a huge limitation to this method, according to the ability of modern computer-vision techniques to automatically locate these parts.

Several computer-vision-based techniques have been proposed in the literature for face recognition. Different researchers have evaluated different computer vision methods for face recognition, such as Local Binary Pattern Histogram (LBPH) [27], Speeded-Up Robust Feature (SURF) and Scale-Invariant Feature Transform (SIFT) [13, 28-30]. However, the use of hand-crafted features imposes the production of several false features that are irrelevant to the recognition task. On the other hand, the methods that are proposed based on neural networks have achieved significantly better performance, with accuracy of 99.13% by [31], using a convolutional neural network.

Schroff et al. [32] propose and train Siamese neural network that is used for face recognition, denoted as FaceNet. The structure of the neural network that is used in this method, shown in Table 2.1, illustrates that each face image is assigned with a descriptor that contains 128 values. Although the values in these descriptors do not reflect any interpretable information, the values provided for a candidate are expected to be similar, regardless of the variable conditions that the captured image can contain. This behavior is ensured by including a huge number of training images that include all possible parameters and their variations, so that, the neural network learns and emphasizes the distinctive features in a face template. In addition to outstanding performance of FaceNet in face recognition, compared to other neural

networks and techniques, this neural network has shown the ability to achieve remarkable performance when used with fingerprint recognition, i.e. using transfer learning, in several researches [33, 34]. Based on the size of the input template, several versions of FaceNet exist in the literature, which can be used based on the size of the extracted face template [34].

Table 2.1: Structure of the FaceNet [32].

type	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj (p)	params	FLOPS
conv1 (7×7×3, 2)	112×112×64	1							9K	119M
max pool + norm	56×56×64	0						m 3×3, 2		
inception (2)	56×56×192	2		64	192				115K	360M
norm + max pool	28×28×192	0						m 3×3, 2		
inception (3a)	28×28×256	2	64	96	128	16	32	m, 32p	164K	128M
inception (3b)	28×28×320	2	64	96	128	32	64	L_2, 64p	228K	179M
inception (3c)	14×14×640	2	0	128	256,2	32	64,2	m 3×3,2	398K	108M
inception (4a)	14×14×640	2	256	96	192	32	64	L_2, 128p	545K	107M
inception (4b)	14×14×640	2	224	112	224	32	64	L_2, 128p	595K	117M
inception (4c)	14×14×640	2	192	128	256	32	64	L_2, 128p	654K	128M
inception (4d)	14×14×640	2	160	144	288	32	64	L_2, 128p	722K	142M
inception (4e)	7×7×1024	2	0	160	256,2	64	128,2	m 3×3,2	717K	56M
inception (5a)	7×7×1024	2	384	192	384	48	128	L_2, 128p	1.6M	78M
inception (5b)	7×7×1024	2	384	192	384	48	128	m, 128p	1.6M	78M
avg pool	1×1×1024	0								
fully conn	1×1×128	1							131K	0.1M
L2 normalization	1×1×128	0								
total									7.5M	1.6B

2.3. DIGITAL IMAGE WATERMARKING

Watermarking an image indicates the embedding of additional information to the visual features of the image. The embedded information can be visible to the human eye, e.g., a logo, or invisible. Different types of watermarking techniques, mainly fragile and robust, are being used for different purposes, e.g., tamper detection and ownership proof. The ownership of the watermarked image can be investigated by investigating the existence of the watermark information. Hence, robust techniques are normally used for this application, as the watermark information is maintained even if the image is manipulated. On the other hand, fragile techniques are normally employed for tamper detection, as the watermark information is usually completely lost whenever an attack is executed against the watermarked image. Some hybrid methods combine both approaches to achieve the intended task, such as localizing the region that has been attacked in the image [11, 12].

As shown in the generic digital image watermarking system in Figure 2.1, the watermark information is embedded to the cover image before the watermarked image is stored or

transmitted. Additionally, such a system is required to be able to retrieve the original information of both the cover image and the watermark from the watermarked image. Moreover, to ensure the integrity of the image, the watermark, cover or watermarked image may be encrypted, so that, the original information is never retrieved without the decryption key. Hence, if the watermarked image is compromised or manipulated, it is impossible for the attacker to regenerate the watermarked image, as the encryption key is missing.

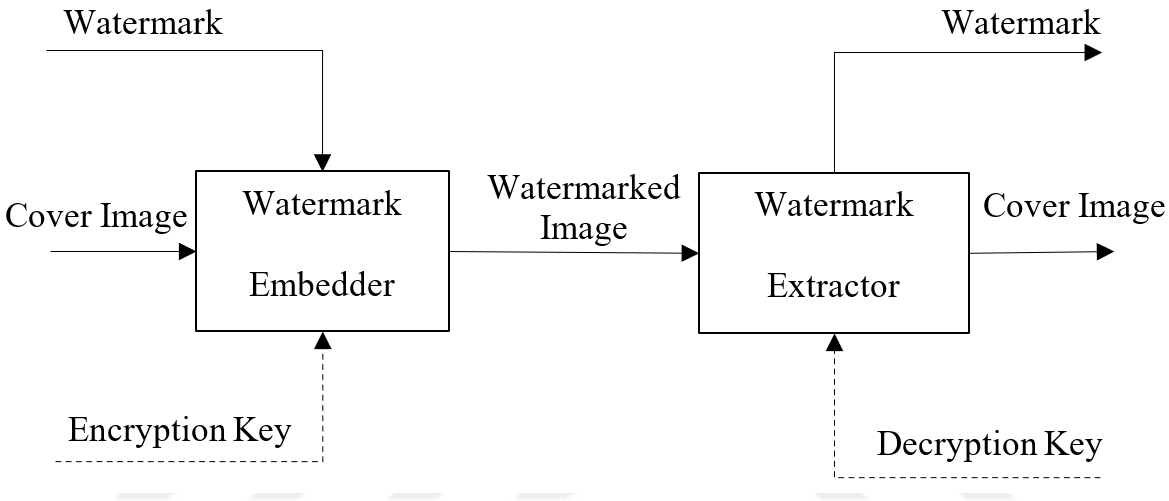


Figure 2.1: Watermark embedding and extraction [12].

2.3.1. Least-Significant-Bit (LSB) Watermarking

This watermarking technique relies on manipulating the value of the Least-Significant-Bit (LSB) of each value in the cover image, based on the value of the watermark information. As this bit has the least visual influence on the pixel value, this manipulation ensures the least possible distortion. Additionally, according to the high probability of changing the LSB when the pixel value is changed, LSB watermarking is considered a fragile technique, as such change in the value results in the loss of the watermark information, as this information is embedded in the LSBs. Moreover, the number of bits that the cover image can hold is equal to the total number of values that the image contains, which is based on the number of pixels and the number of color channels that represent each pixel.

When compressed using a lossy compression method, such as JPEG, the watermark information is lost when using the LSB method, as this value does not contain significant weight towards the pixel value. Accordingly, when such a watermarking technique is used

with the spatial domain of the image, the watermarked image cannot be compressed, to maintain the watermark information intact. Moreover, despite the ability of compressing the watermarked image, using the LSB method, using lossless compression techniques, these techniques do not provide significant reduction in the size of the data that represent the watermarked image.

2.3.2. Discrete Cosine Transform (DCT)

Per each set of values that are collected from 8×8 fixed-size non overlapping regions in the image, the DCT aims to define these values using a similar 8×8 frequency values for cosine waveforms that replication the same values when summed up together. The DCT coefficients are calculated as shown in Equation 2.1 [35].

$$B_{pq} = a_p a_q \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} A_{hw} cos \frac{\pi(2h+1)p}{2H} cos \frac{\pi(2w+1)q}{W}, \begin{matrix} 0 \le p \le H-1 \\ 0 \le q \le W-1 \end{matrix} \quad (2.1)$$

where

$$a_{qp} = \begin{cases} \frac{1}{\sqrt{H}}, p = 0 \\ \sqrt{\frac{2}{H}}, 1 \le p \le H-1 \end{cases}$$

and

$$a_q = \begin{cases} \frac{1}{\sqrt{W}}, q = 0 \\ \sqrt{\frac{2}{W}}, 1 \le q \le W-1 \end{cases}$$

Using these coefficients, the pixel values of the original image can be retrieved using the formula shown in Equation 2.2.

$$A_{hw} = \sum_{p=0}^{H-1} \sum_{q=0}^{W-1} a_p a_q B_{pq} cos \frac{\pi(2h+1)p}{2H} cos \frac{\pi(2w+1)q}{W}, \begin{matrix} 0 \le h \le H-1 \\ 0 \le w \le W-1 \end{matrix} \quad (2.2)$$

where

$$a_{qp} = \begin{cases} \frac{1}{\sqrt{H}}, p = 0 \\ \sqrt{\frac{2}{H}}, 1 \leq p \leq H - 1 \end{cases}$$

and

$$a_{q} = \begin{cases} \frac{1}{\sqrt{W}}, q = 0 \\ \sqrt{\frac{2}{W}}, 1 \leq q \leq W - 1 \end{cases}$$

Although the number of values that are produced by calculating the DCT coefficients is identical to the number of values in the original image, the resulting coefficients can be categorized into three frequency bands, low medium and high. The lower frequency bands are responsible for generating the DC components of the pixel values in the regions, e.g., the average intensity of the pixels in that region. These low frequency bands are very important for the human eye, as the change in these values has significant influence on the pixel values in that region. As the frequency bands are increased, the significance of the features that the patterns defined by these bands become less noticeable by the human eye. Accordingly, several watermarking techniques rely on manipulating the DCT coefficients, especially those in lower bands, to embed robust watermarks to the image.

2.3.3. Arnold Transformation

To secure the watermark information, several of the watermarking techniques rely on encrypting this information, so that, it is impossible for an attacker to regenerate the watermark information, as the encryption key is unknown. Arnold Transform (AT) is one of the efficient encryption methods that is widely used with digital watermarking, in which the positions of the values in the array are scrambled based on the encryption key. When received, the original positioning of the values is retrieved by using the decryption keys [36]. Accordingly, if the decrypted watermark information does not match the expected

information, then the received image is considered to be tampered with, regardless to the reason of the loss of the watermark information.

For each value in a $D \times D$ image, the new position of that value, x and y, are calculated using a key with two secret values, a and b, as shown in the formula in Equation 2.3, where i is the iteration number.

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix} mod(D) \quad (2.3)$$

Hence, restoring the original position of a value, that is scrambled using AT, is achieved by using the same key values and number of iterations, as shown in Equation 2.4 [30].

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} ab+1 & -a \\ -b & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix} mod(D) \quad (2.4)$$

2.4. ARTIFICIAL NEURAL NETWORKS

The mathematical operations in artificial neural networks are inspired by the human brains, in which an enormous number of nerve cells exchange electrical signals based on information collected from the external world. In average, a human's brain contains 10^{11} biological neurons that are interconnected to each other to create complex networks. As shown in Figure 2.2, the body of the neuron's cell receives adjusted electrical signals through its dendrites. Each dendrite may collect thousands of inputs, which may be collected from other neurons or other parts of the body, such as vision nerves in the eyes and smell sensors in the nose. The influence of the input on the output of the body cell is adjusted by adjusting the conductivity of the electrochemical solution in the synapsis correspondent to that input. However, the type of the influence that an input can cause, i.e., excitatory or inhibitory, is decided by the body cell, where excitatory inputs induce output from the body cell and such an output is prevented by inhibitory inputs. The output of a neuron can finally be provided as input to another neuron or interpreted into a decision in the human body, e.g., a movement [37-39]. Accordingly, the magnitude of the electrical signal, known as membranes, which are short-lived impulses, is controlled by the conductivity of the synapsis and its influence on the output is decided by the cell body.

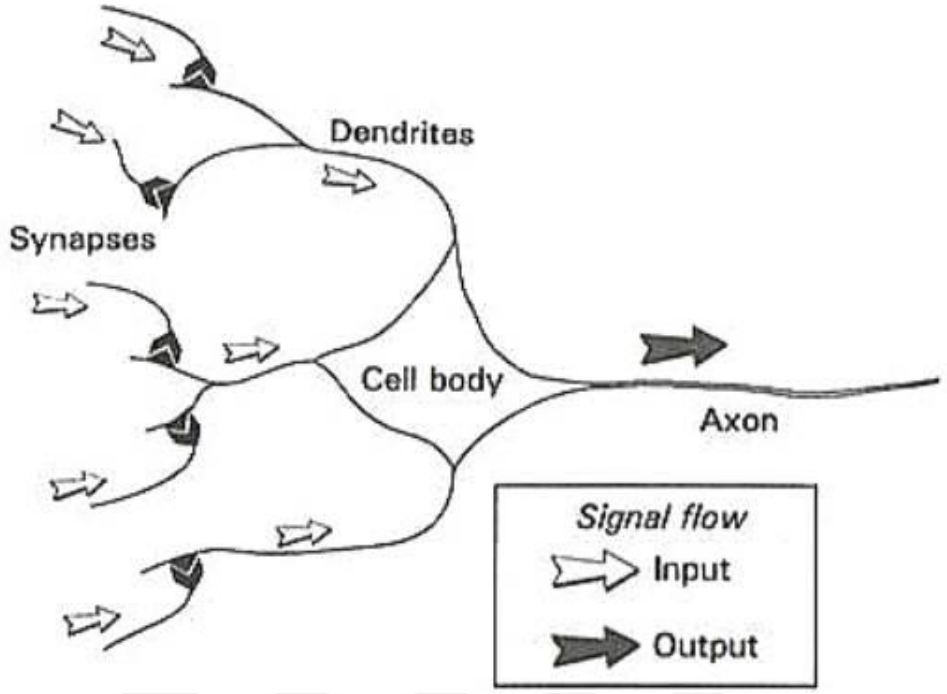


Figure 2.2: A neuron's Essential Components [37].

To mathematically represent these neurons and the networks that they create, the basic component of an artificial neural network, the artificial neuron, also adjusts the influence of each input on its output by using a parameter known as the weight. The value of the weight adjusts the influence of the corresponding input, in terms of how significant is that input, whereas the sign of that weight adjusts the type of the influence, i.e., excitatory or inhibitory. Then, the output of the neuron is calculated by summing the weighted inputs, as shown in Equation 2.5, where S is the summation result of the inputs x, each weighted by the corresponding weight value, w. Additionally, the neuron uses a bias value b to improve the flexibility of computations [40]. These computations are also illustrated visually in Figure 2.3, which shows another important component of the neurons, which is the activation function.

$$S_j = \sum_i x_i w_{ij} + b_j \tag{2.5}$$

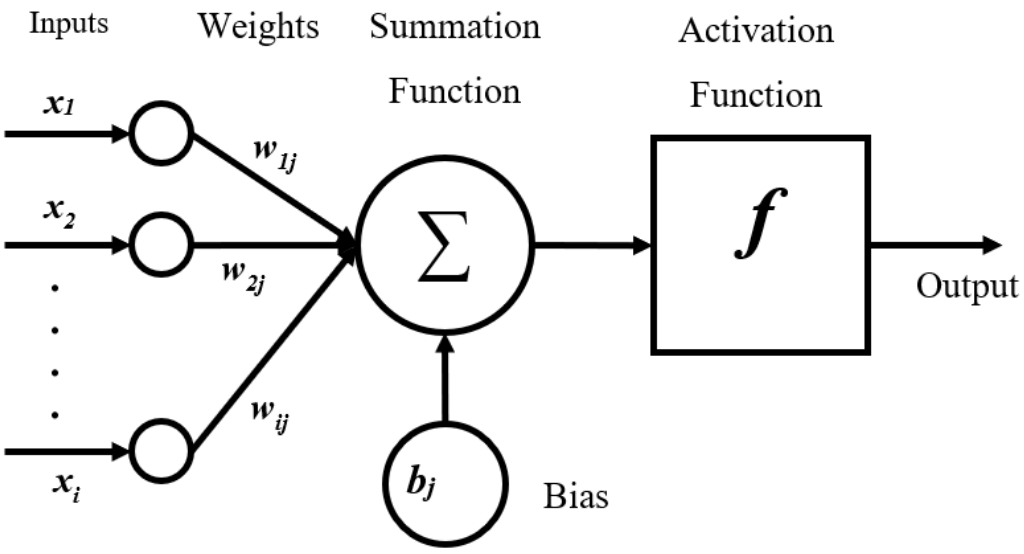


Figure 2.3: Visual Illustration of the Neuron's Computations [40].

According to Equation 2.5, the output of a neuron is linear to its inputs, regardless of the values of the weights and bias. To allow these networks to handle and solve non-linear problems, activation functions are being applied to the output of the neurons, after summing the weighted inputs and the bias value. This non-linearity provides the neuron with better ability to craft the features it can detect, which is a certain pattern in the input values. As shown in Figure 2.4, there are several activation functions that are being used with artificial neural networks, which has been able to improve the performance of these networks. Additionally, limiting the values that flow in, or outputted by, the neural network is a significant feature of activation functions, where for instance, the Hyperbolic Tangent (TanH) function limits the output in the interval [-1, 1], whereas sigmoid limits it in the interval [0, 1]. These functions are widely used in the output layer, based on the range of values required from the neural network [41, 42], whereas the use of the Rectified Linear Unit (ReLU) in hidden layers has shown significant improvement in the performance of the artificial neural networks in different applications [43, 44].

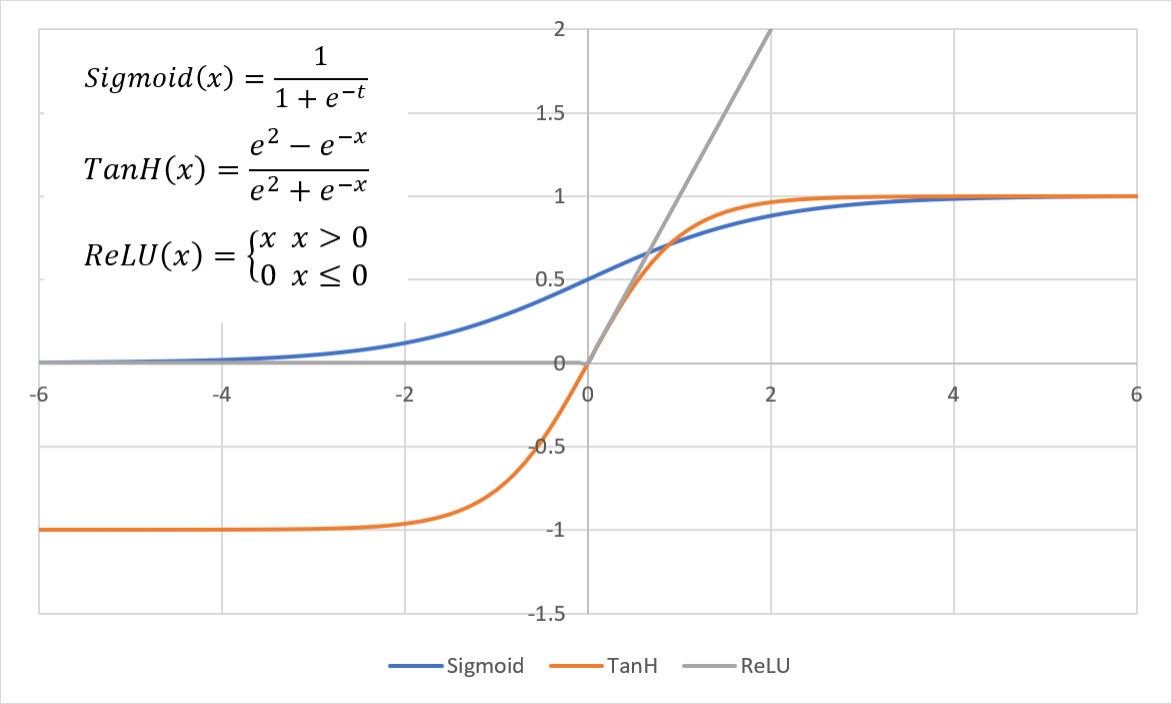


Figure 2.4: Activation Function for Neurons [43, 45].

Despite the ability to produce complex artificial neural networks, the neurons in a standard neural network are distributed in layers, so that, the neurons in a certain layer process the outputs of the ones in the previous layer. This distribution allows the neurons to detect complex features by combining the features detected in the previous layer, which may also be a combination of the features detected in a previous layer of detected directly in the input. In addition to the layers that are used to detect these features, which are known as hidden layers, every artificial neural network has an input layer, which gathers the input from the external environment, and an output layer that presents the computed values to the environment [46]. Figure 2.5 presents a sample feedforward neural network with four inputs, two outputs, which govern the number of neurons in the input and output layers, in addition to two hidden layers. To increase the complexity of the patterns, or features, that the neural network can detect, the number of hidden layers is increased. However, this increment increases the complexity of the required computations, according to the increased number of weights, biases and neurons. Moreover, to increase the number of patterns that the neural network is capable of detecting at a certain level of complexity, the number of neurons at the

corresponding layer is increased, which also increases the complexity of the computations. Hence, a balanced topology is required to detect the features that allow the neural network to achieve the required task with minimum possible complexity.

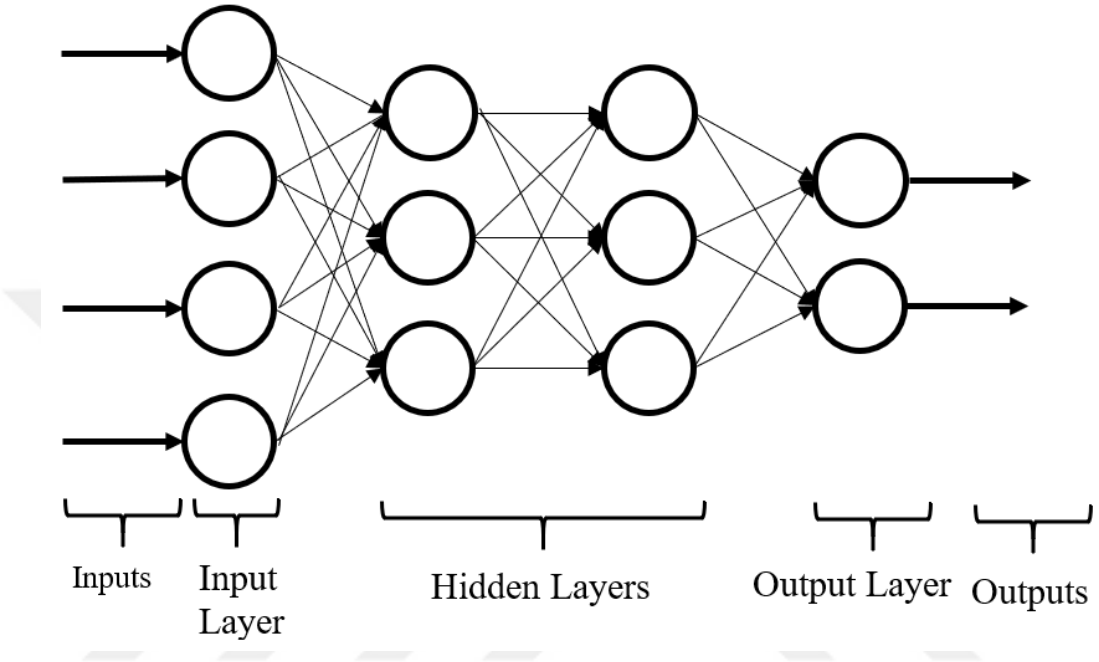


Figure 2.5: Sample Feed-Forward Artificial Neural Network [47].

According to the computations that are being executed by each neuron and the distribution of the neurons in an artificial neural network, the patterns that each neuron can detect, in order to calculate the output required from the neural network, are governed by the weights and bias value of each neuron. These values are optimized to minimize the difference between the values outputted from the neural network and the values required from the network per each input, which is known as the loss. Any optimization algorithm can be used for this purpose, however, the enormous number of parameters to be optimized in the neural network has imposed the need to use efficient optimization algorithms that has the ability to optimize such a number of parameters shortly. Accordingly, gradient descent optimization algorithms are being used for this purpose, which require a convex loss function, as shown in Figure 2.6. This type of optimization algorithms relies on calculating the partial derivative of each parameter to the loss, which indicates the direction in which changing the value increases the loss. Hence, the optimization algorithm changes the value in the opposite

direction, to minimize the loss, which represents the difference between the required and actual values. Accordingly, this type of optimization algorithms can get easily stuck in local minimums, which non-convex loss functions are used [48, 49].

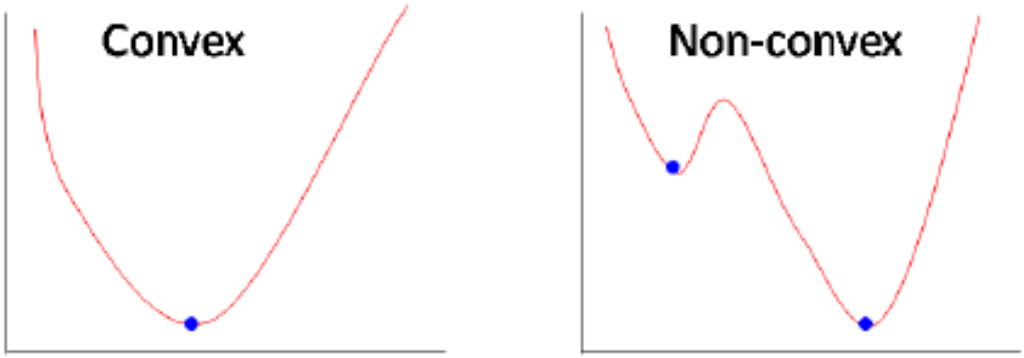


Figure 2.6: Convex versus non-convex function [50].

One of the widely used loss, also known as cost, functions in training neural networks for regression applications is the Sum of Squared Errors (SSE), shown in Equation 2.6. The output value *y* is calculated by passing the input values, which are beyond the control of the neural network, through the neurons in the layers of the neural network. Then, the output values are provided to the loss function, as well as the actual values $\hat{y}$, which are required from the neural network. Accordingly, the neural network has the only option of adjusting the weights and biases of the network, in order to minimize the loss and provide the values required from the neural network [51].

$$L = \sum_i (y_i - \hat{y}_i)^2 \tag{2.6}$$

There are two passes of computations in the neural network, a forward pass to calculate the output of the neural network for a certain input, and backpropagation, in which the weights and biases are updated. The use of reverse order to update the weights and biases of the neural network is to eliminate any redundancy in computations, as the partial derivatives are being

calculated with respect to the loss, which is at the output of the neural network. Hence, the computations of the partial derivatives for the weights and biases in the neurons of the output layer, when being calculated with respect to the loss, are used to calculate the partial derivatives of the weights and biases of the neurons in the last hidden layer, instead of repeating the same computations. Considering the bias as a weight or an input with a constant value of one, the same results can be collected from the neuron and the bias value can be updated alongside all other weight values [52].

The new weight value $\widehat{w}$ is calculated as shown in Equation 2.7, where *L* is the value of the loss, and *E* is the value of the learning rate. As the value of $\frac{\partial w}{\partial L}$ indicates the gradient ascent, this value is subtracted from the original weight value to reduce the loss, i.e., match the required output values. Moreover, the learning rate E is used to avoid huge updates to the weight values, which can cause the optimizer to miss the global minimum of the loss and limits the performance of the neural network. Accordingly, this process is repeated for a predefined number of time, i.e., epochs, or until a certain condition is met, e.g., reaching a certain loss value [53-55].

$$\widehat{w} = w - \frac{\partial w}{\partial L} \times E \times L \tag{2.7}$$

2.4.1. Autoencoders

As mentioned earlier, the inputs of a neuron in a certain layer are the outputs of the neurons in the previous layer. Accordingly, it is possible to retrieve the values in previous layers, as the ones in this layer still maintain the information of the features that exist in the input. Autoencoders exploit this property of ANNs in order to reduce the amount of information that is used to represent the inputs. The input is processed by a set of convolutional layers before being passed through the bottleneck layer, which is the layer that has the lowest number of neurons in the neural network, as shown in Figure 2.4. Then, by expanding the values in the bottleneck layer to reach the same size as the input, the neural network is trained to produce the same image as the inputs, i.e. the output required from the ANN is identical to its input. Accordingly, the neural network learns to retrieve the original information based

only on the values in the bottleneck layer. Thus, after training the neural network, it is split into encoding and decoding layers, where the image is encoded using the encoding network, then the values are sent to the receiver which retrieves the original image by passing the values through the decoding part of the ANN [56, 57].

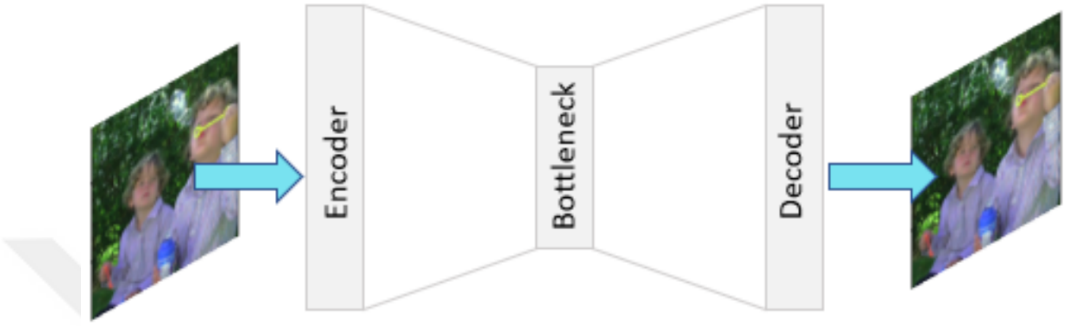


Figure 2.7: Structure of an autoencoder [58].

2.4.2. Classification Using ANNs

Based on the type of the input, a set of layers is used to extract the features that are required to reach the output from the provided input. In classification tasks, the required output represents the category that the input belongs to. Accordingly, a number of neurons is placed in the output layer based on the number of categories that exist in the classification domain, where the output of each neuron represents the probability of the input to be in the corresponding class. However, in binary classification, in which there are only two classes in the domain, a single neuron can be used to represent the probability of the input to be from one of these categories, as the increased probability of being in one of the classes reduces the probability to be in another. The transfer function that is used by the neurons in the output layer is governed by the type of classification the ANN is being used for, e.g. soft-max function for multi-class and sigmoid for binary classification.

2.5. JPEG COMPRESSION

To reduce the size of data required to represent the visual features in an image, JPEG standard [59] computes the Discrete Cosine Transform (DCT) coefficients of every 8×8 pixels block in the image using 8×8 frequencies, i.e. the number of values produced from this step is equal to the number of pixels in the image. Then, a quantization table is used to adjust the values

based on the significance of the frequency in the image, where normally lower frequencies have more influence on the image. Hence, the quantization table normally has higher values at higher frequencies, so that, most of the values produced for those frequencies are zero because the JPEG standard uses the integer values of the division result. This step represents the lossy compression in the JPEG standard, as the actual values of the DCT coefficients are being manipulated to reduce the number of unique values in the quantized value, based on the quantization table, as shown in Figure 2.8. This reduction allows more efficient lossless compression using Huffman Encoding, which is applied to the series of values extracted from the blocks [59-62], as shown in Figure 2.9.

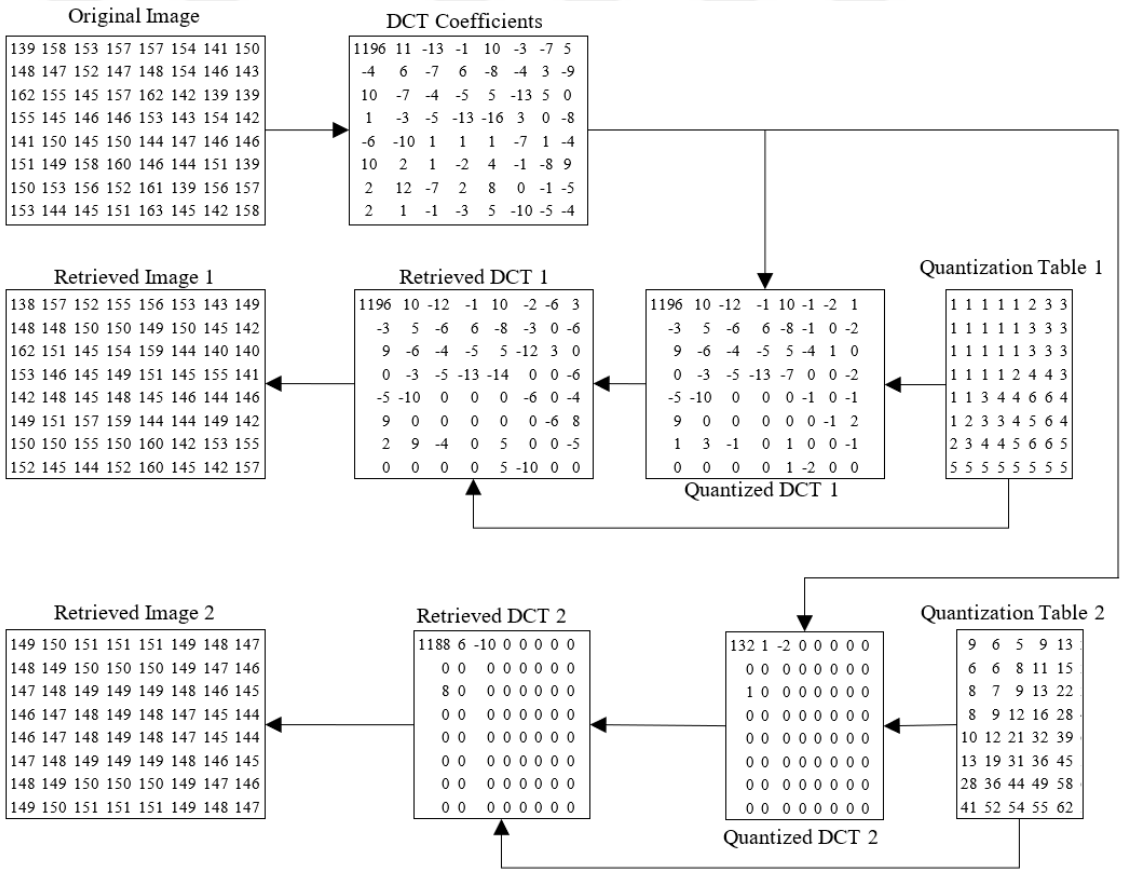


Figure 2.8: DCT coefficients extraction and quantization in the JPEG compression standard [59].

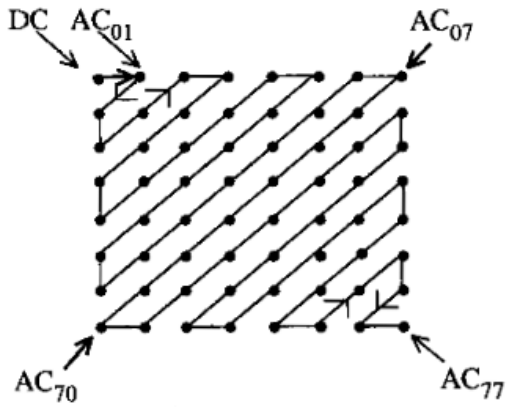


Figure 2.9: Converting an array of values into a series in the JPEG compression standard [63].

3. METHODOLOGY

The method proposed in this study uses a neural network to extract a set of distinctive values from the iris template and use them as a watermark over the face image. According to this approach, the integrity of the received data can be investigated by investigating the existence of iris information in the watermark and the matching can be completed using the extracted iris information and face templates, if and only if the data is recognized to be intact. For watermarking, the proposed method embeds the values extracted from the iris template into the DCT coefficients of the face template, i.e. produced a compressed watermarked template. Upon arrival, the watermark information is extracted and another neural network is used to detect whether the extracted iris information is legit or not, i.e. detect tampering.

3.1. IRIS FEATURES EXTRACTION

To reduce the size of the data that is to be used as a watermark over the face template, a neural network is used to extract and compress the data that represent the significant information in the iris. However, the extracted features must contain information about the entire iris image, instead of only defining the distinctive features that are used for the authentication process. This requirement is defined by the use of the iris template as a watermark to protect the communicated templates, so that, deep features are required for the entire image to investigate the patterns that appear in the retrieved, i.e. decompressed, version of the image. Additionally, to reduce the distortion imposed on the face image, a binary transfer function is used in the bottleneck layer, so that, the value outputted by each neuron in that layer is either zero or one. Hence, each value can be represented by manipulating a single bit in the cover image, i.e. the face image. Accordingly, a fixed-size descriptor is generated for the iris template, in which the values are binary.

3.2. WATERMARK EMBEDDING

To embed the values extracted from the iris template, the proposed method follows the JPEG standard, according to the popularity and high efficiency of that standard. As shown in Algorithm 1, the DCT coefficients are calculated for the cover image. Then, based on the

size of the descriptor generated for the iris, the bits of the DCT coefficients are adjusted to embed this information. However, to avoid the extraction of these bits, modification of cover image data and re-embedding the watermark, the proposed method includes information of the original image, extraction from the remaining bits of the DCT coefficient. For instance, if only the LSB of the coefficient is being manipulated, XOR is applied to the remaining seven bits of the DCT coefficient. Then, the same XOR operation is applied to the result of the operation and the bit from the iris descriptor. The results of the latest XOR operations are stored in a separate matrix, which is encrypted by using Arnold Transform, and are used to replace the bits in the coefficients extracted from the coefficients of the cover image, as shown in Figure 3.1. The encryption of the watermark information ensures that an attacker cannot reproduce the watermark information because the position of the DCT coefficient that is used in the XOR operation is unknown. Moreover, collecting the bits of the watermark image does not allow the reproduction of the watermark, if the DCT values are manipulated by an attacker.

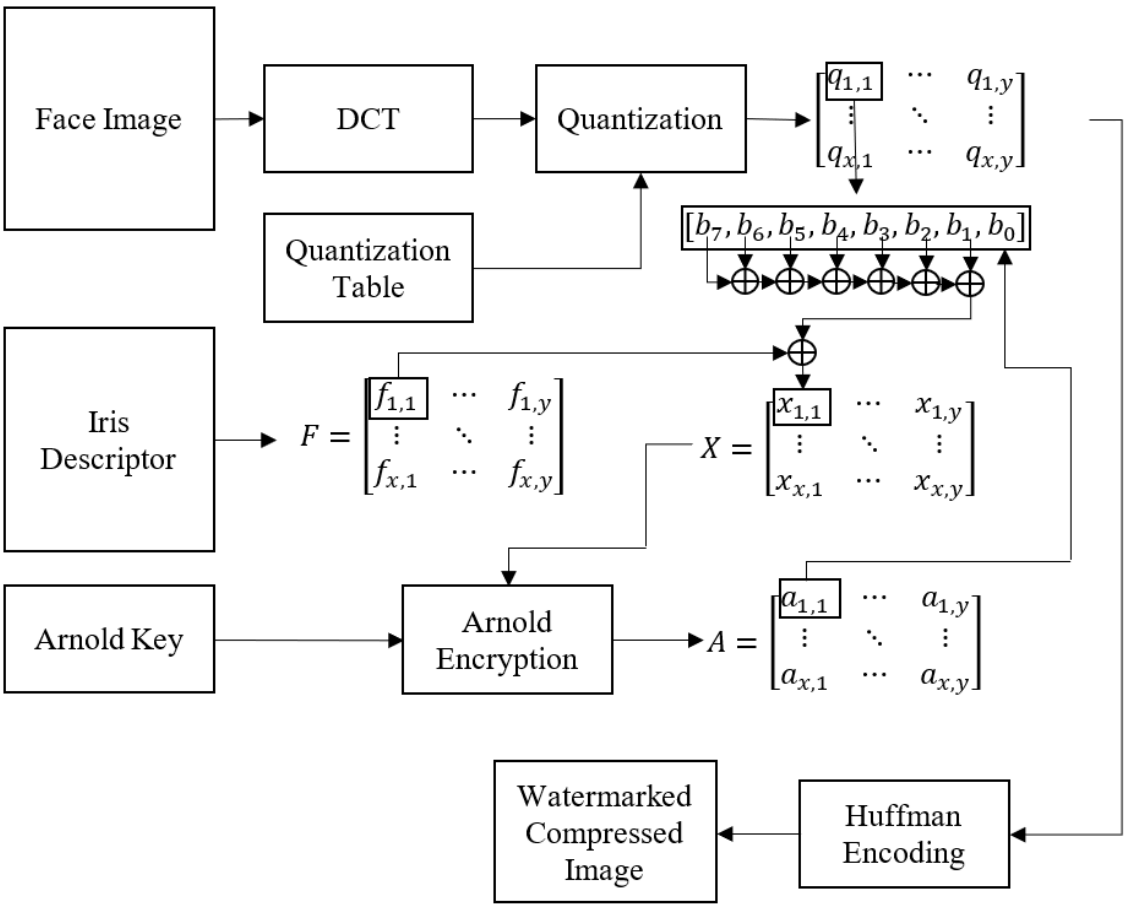


Figure 3.1: Block diagram of the proposed watermarking method.

3.3. TAMPER DETECTION

Upon the arrival of the watermarked template, the bits corresponding to the iris are extracted from the image in a separate matrix, as well as a separate matrix for the result of applying XOR to the remaining bits in the DCT coefficients. Then, Arnold Transform is used to rearrange the bits corresponding to the iris descriptor before applying XOR for each of the rearranged bits and the one in the same position in the matrix resulted from applying the XOR to the bits of the DCT coefficient, as shown in Figure 3.2. The result of this operation is then transformed into a descriptor of the iris, which represents the data generated by the neural network that is used to extract these values from the original iris template. Accordingly, these values must contain information about the iris patterns in the original

input, otherwise, the received template is considered to be tampered with and the authentication process is aborted.

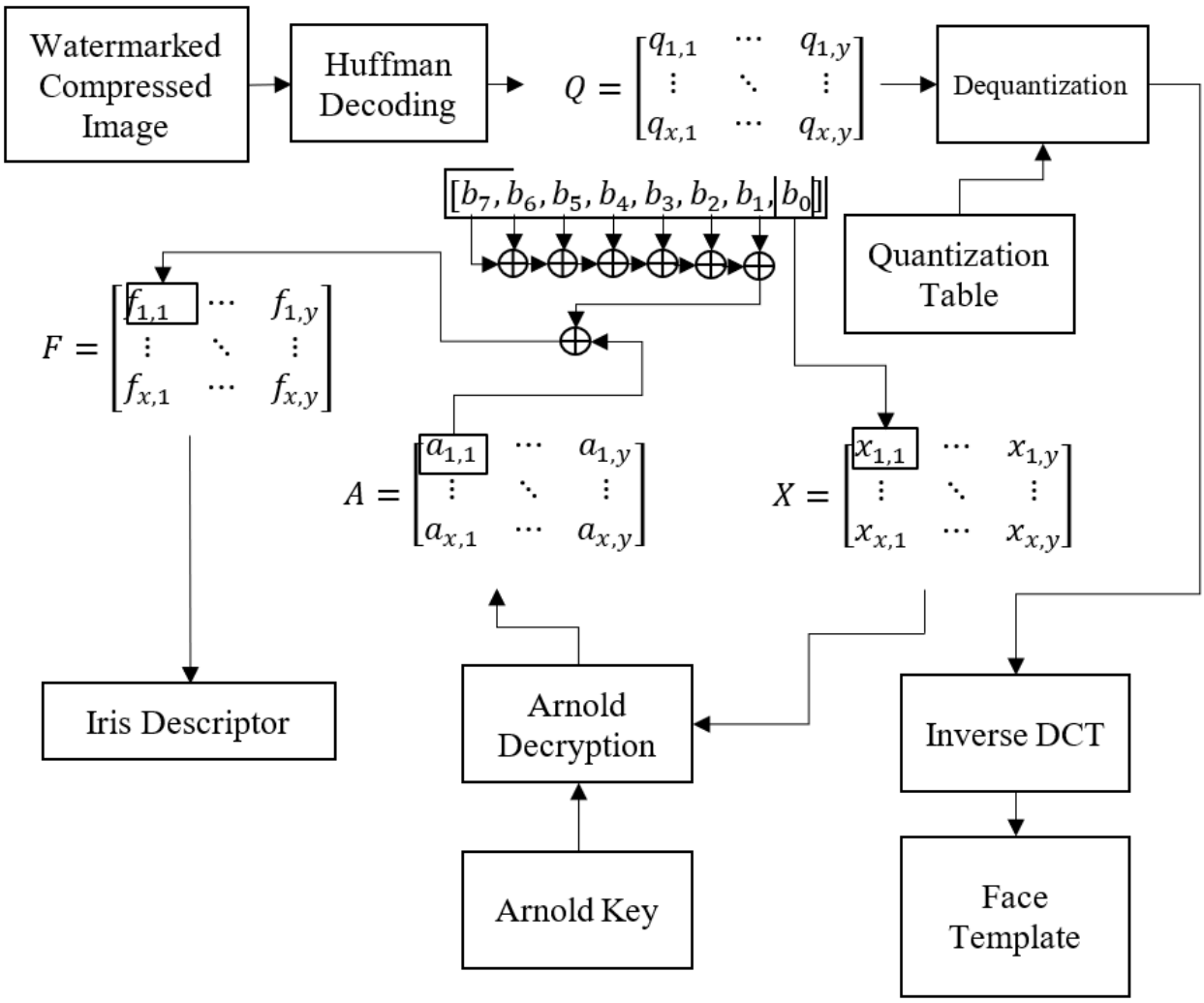


Figure 3.2: Watermark information extraction from the watermarked biometric template.

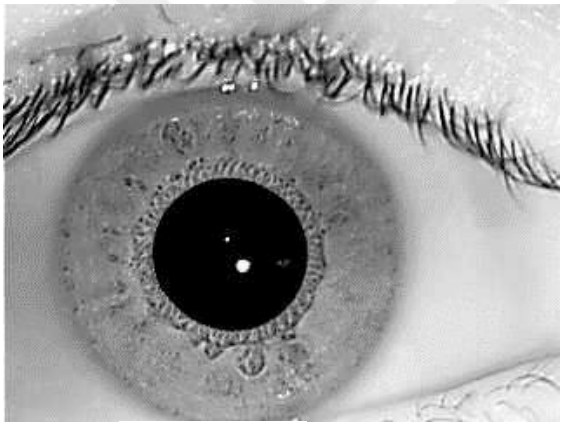
As mentioned earlier, when the watermark information is extracted from a biometric template, this information cannot be consistent even for the same user. Accordingly, tampering cannot be detected by simply comparing the extracted watermark information to a predefined model. Instead, a neural network is trained to recognize the validity of the extracted descriptor, whether to be generated by the proposed feature extraction neural network for a legit iris template or not. Accordingly, the neural network implemented for this task is trained as a binary classifier, where the classifiers to be intact or tampered with. By

comparing to the output of this neural network to a threshold value, a decision about whether to proceed with the matching procedure to authenticate the user or to abort it can be made.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. EXPERIMENTAL SETUP AND DATASETS

The proposed method is implemented using Python programming language with the aid of the Tensorflow library for neural networks implementation, training and evaluation. Two datasets are used during the training and evaluation, which are the Indian Faces Dataset and the Indian Institute of Technology (IIT) Delhi iris dataset. Figure 4.1 shows samples of the selected datasets. All experiments are conducted using Windows operating system, running on a computer with Intel Core i7 processing, 16GB of random access memory and GeForce GTX-1080Ti Graphical Processing Unit (GPU), which is equipped with 8GB of memory.



(a)

(b)

Figure 4.1: Samples of the images in the selected datasets. (a) image from the IIT Delhi iris dataset; (b) image from the Indian Faces Dataset.

4.2. INFLUENCE OF THE AUTOENCODER ON IRIS MATCHING

Despite the ability of ANNs to provide efficient compression, i.e. reduce the size of data required to communicate the image, it is important to evaluate the distortion imposed by this compression and its influence on the matching of the iris templates. For this purpose, two experiments are conducted to evaluate this influence. In the first experiment, the Structural Similarity Index Measure (SSIM) and Peak Signal to Noise Ratio (PSNR), shown in Equations (4.1) and (4.2) consequently, are measured between the iris templates provided to the autoencoder neural network and those outputted from it. For this evaluation, the

autoencoder hierarchy proposed in [64] is used with the iris dataset. The results show that this autoencoder has been able to maintain an average PSNR of 48.87dB and SSIM of 99.76%. The high similarity between the input and output images is according to the relatively highly repetitive patterns in the iris templates, e.g. sclera and pupil, which can be easily compressed and restored with high similarity.

$$SSIM = \frac{4\sigma_{IR}\overline{I}\ \overline{R}}{(\sigma_I^2 + \sigma_R^2)\left[(\overline{I})^2 + (\overline{R})^2\right]} \tag{4.1}$$

where,

$$\overline{I} = \frac{1}{n}\sum_{i=1}^{n} I_i, \qquad \overline{R} = \frac{1}{n}\sum_{i=1}^{n} R_i,$$

$$\sigma_I^2 = \frac{1}{n-1}\sum_{i=1}^{n}(I_i - \overline{I})^2, \quad \sigma_R^2 = \frac{1}{n-1}\sum_{i=1}^{n}(R_i - \overline{R})^2,$$

$$\sigma_{IR} = \frac{1}{n-1}\sum_{i=1}^{n}\left(I_i - \overline{I}\right)\left(R_i - \overline{R}\right).$$

$$PSNR = 20log_{10}Max - 10log_{10}MSE, \tag{4.2}$$

where,

$$MSE = \frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - R(i,j)]^2$$

For the second evaluation, the influence of the compression of the image over the authentication accuracy is evaluated. First, the iris is normalized using rubber sheet model, after being extracted from the image using Hough transform, and then Alex-Net is used to extract distinctive features that are used in the recognition, i.e. matching, phase, based on the system designed in [65]. This system achieved 98.62% accuracy using the IIT Delhi iris dataset directly, i.e. without compressing the images. However, this accuracy has dropped to 98.58% when the features are extracted from the compressed versions of the image. This indicates that despite the significant reduction in the size of the data, represented by binary

values, the recognition accuracy has only been reduced by 0.04%. This slight reduction also indicates that the compressed templates have been able to maintain the patterns in the original templates, which indicates the ability to use these templates for accurate tamper detection.

4.3. INFLUENCE OF THE PROPOSED WATERMARKING METHOD ON COVER TEMPLATE

When watermark information is used to manipulate the cover image, this manipulation changes the values assigned to the pixels in that image. Hence, similar to the experiments conducted in the previous section, two experiments are conducted to evaluate the influence of the watermark embedding over the face image. Thus, the iris images in the IIT Delhi dataset are compressed and used as a watermark over the face image in the Indian Faces Dataset. The average SSIM between the face images before being watermarked and the watermarked ones is 99.83%, whereas the average PSNR is 48.98dB. These high similarity measures illustrate the high efficiency of the proposed watermarking method, as this method only manipulates the LSB and there are equal chances between changing or maintaining the bit, as it has only two possible values. Additionally, the influence of this change in the face images over the face recognition neural network (FaceNet) proposed in [32] has shown a similar performance of 98.17% accuracy in both cases. The similar accuracy indicates that the proposed watermarking method has minimal influence over the face image and the ability of FaceNet to handle such minor changes in the templates without affecting their performance.

Moreover, the proposed method has shown better performance than the methods proposed by [66-68] by measuring the influence of the watermark on the similarity between the original face image and the watermarked one. This performance, summarized in Figure 4.2, shows that the proposed watermarking technique has the least manipulation of the cover image. Such low influence is a result of only manipulating the LSB value and replace it with a value from the watermark information, which can be the same bit value in 50% of cases. Hence, in general, the influence is limited to the LSB of only 50% of the values, which has provided the proposed method with the ability to maintain high fragility while maintaining low influence over the cover image.

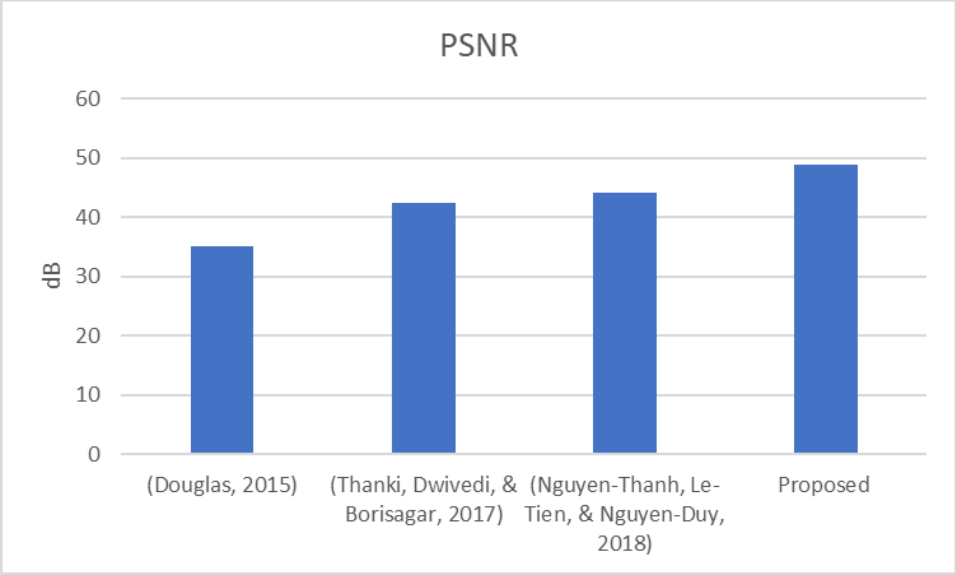


Figure 4.2: Comparison of cover image similarity before and after watermarking.

4.4. TAMPER DETECTION

Based on the results of the previous experiments, which show that the proposed watermarking method has minimal influence on the recognition accuracy of biometric authentication systems, it is important to evaluate its ability to detect tampering with the biometric templates. First, the fragility of the proposed method is evaluated by executing a set of attacks on the watermarked image. The results shown in Table 4.1 show that the proposed method has maintained a very high fragility, illustrated by the low similarity, i.e. SSIM and PSNR, between the original and extracted watermark information.

Table 4.1: Similarity measures of the original and extracted watermarks after attacking the watermarked image.

Attack	PSNR (dB)	SSIM (%)
Median Filter (15x15)	3.56	0.04
Median Filter (9x9)	3.99	0.09
Median Filter (7x7)	3.86	0.04
Median Filter (5x5)	3.69	0.1
Median Filter (3x3)	3.86	0.08
Gaussian Noise (0.05,0)	3.83	0.05

Gaussian Noise (0.1,0)	4.43	0.08
Gaussian Noise (0.02,0)	4.52	0.11
Gaussian Noise (0.01,0)	3.88	0.05
Gaussian Noise (0,0.1)	3.96	0.06
Gaussian Noise (0,0.05)	4.19	0.05
Gaussian Noise (0,0.01)	3.6	0.1
Salt & Pepper (0.1)	4.42	0.06
Salt & Pepper (0.05)	3.87	0.06
Salt & Pepper (0.01)	3.84	0.05
Salt & Pepper (0.002)	4.13	0.09
Salt & Pepper (0.001)	4.03	0.05
JPEG compression (90%)	4.26	0.08
JPEG compression (70%)	4.25	0.07
JPEG compression (50%)	4.25	0.1
JPEG compression (30%)	3.83	0.04
JPEG compression (10%)	3.71	0.09

Moreover, in addition to the slightly better PSNR and SSIM achieved by the proposed method, compared to [68], as shown in Figure 4.3, a significantly better fragility is achieved by the proposed method. This fragility is illustrated by the dramatic reduction in the SSIM and PSNR between the original and retrieved watermarks, when different attacks are applied. Accordingly, despite the capability of the method proposed by [68] to detect these attacks, the significantly lower similarities achieved by the proposed method ensures more confident detection.

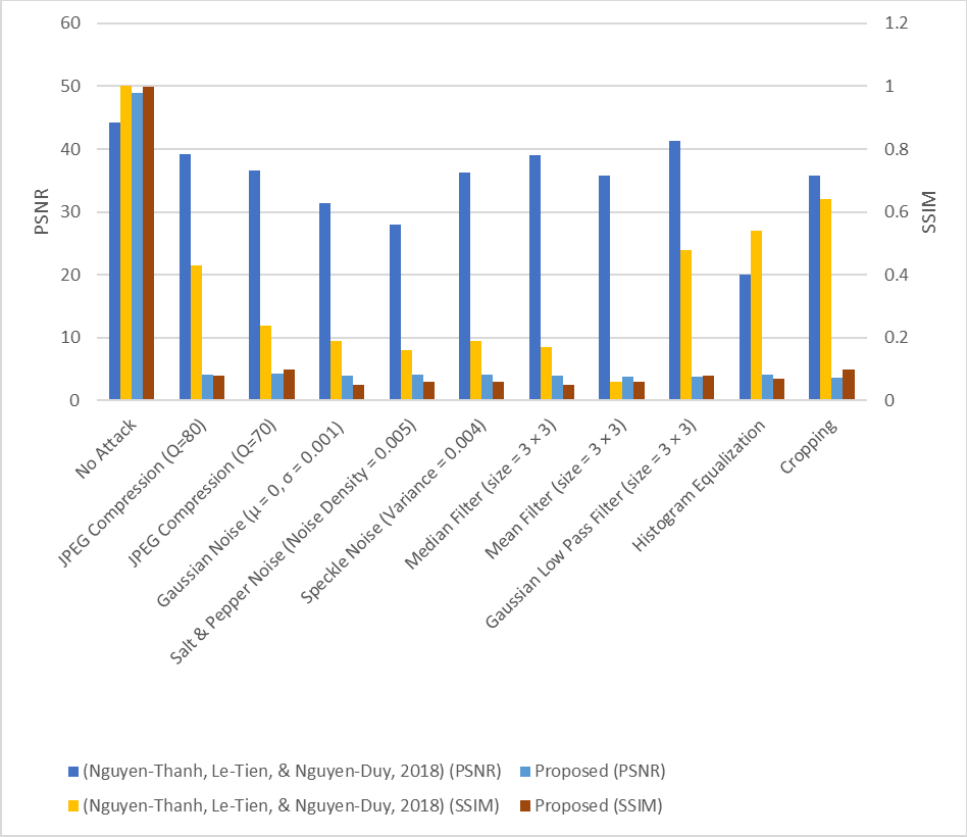


Figure 4.3: Comparison of PSNR and SSIM of the watermark with and without attacks.

Despite the significantly low similarity measures, it is still important to ensure that the watermarks in the attacked images do not contain patterns of iris templates. Hence, a, artificial neural network is implemented and trained to detect tampering in the extracted watermarks in a binary classifier approach, i.e. a single neuron in the output layer with the sigmoid activation function. However, as the values extracted from the watermark already represent the deep features and pattern recognized in the iris template, a single layer with 128 neurons and ReLU activation is placed before the output layer. The values extracted from the watermark are flattened and fed to this layer, which forwards its outputs to the neuron in the output layer. Instead of retrieving the entire iris template by using the decoding part of the autoencoder, this neural network can provide significantly faster decisions of whether to continue with the authentication process or not, by investigating tampering in the received templates. This tamper detection approach has achieved 100% accuracy when evaluated using the selected dataset. This perfect accuracy shows that the use of the autoencoder approach has been able to maintain the significant features in the iris image, whereas the

proposed watermarking method has maintained minimal influence on the face image, which is the cover image. Moreover, these results also illustrate the ability of neural networks to detect complex patterns and deep features in the inputs, similar to the results provided in [69, 70].

5. CONCLUSION

With the growing importance of digital services, sensitive information is being stored and provided to users. Protecting this information has become a major concern in recent years, as compromising this information or service can produce catastrophic results. Thus, biometric authentication systems have been introduced to govern access to these services, according to the high robustness and availability of these credentials and the difficulty of replicating them. Additional accuracy has been provided to these systems by using templates collected from different parts of the human body but has addressed vulnerabilities of biometric authentication systems. The main vulnerability of such a system is the manipulation of the templates during transportation between the different stages of the authentication system. Hence, a method is proposed in this study to detect any tampering with the templates when they arrive at the matching stage.

The proposed method relies on fragile watermarking the iris features over the face template, so that, the same bandwidth that is consumed to deliver the face template is used to deliver both templates. Moreover, the proposed method embeds the watermark information in the compressed face image, instead of the full-size template to further reduce bandwidth consumption. The evaluation results of the proposed method have shown its ability to achieve 100% tamper detection accuracy, which shows that the method has been able to detect any tampering with the communicated watermarked templates. This detection has been achieved while imposing minimal influence on the visual features in both the face and iris templates. However, although the proposed method has shown no change in the recognition accuracy of face templates with and without tampering, it has reduced the recognition accuracy based on the iris template by 0.05%. This slight reduction in the iris recognition accuracy can be rectified by the high face recognition rate, which has been the same while maintaining perfect, i.e. 100%, tamper detection rate.

In future work, the ability to apply the proposed method to other types of templates is going to be investigated. However, this application can be limited by the huge amount of information that is required to represent other types of biometric templates, compared to the iris. Additionally, the ability of generating descriptors of the templates in the watermark, instead of compressing the entire template and embed it to the cover image is also going to

be investigated. Such an approach allows the proposed method to impose less distortion to the cover image but the efficiency of the descriptor generator must be validated against the results of using the entire template in this study.

REFERENCES

[1] H.-M. Sun, S.-T. Chen, J.-H. Yeh, and C.-Y. Cheng, "A shoulder surfing resistant graphical authentication system," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, pp. 180-193, 2018.

[2] M. Nagatomo, Y. Kita, K. Aburada, N. Okazaki, and M. Park, "Implementation and user testing of personal authentication having shoulder surfing resistance with mouse operations," *IEICE Communications Express*, vol. 7, pp. 77-82, 2018.

[3] D. Dasgupta, A. Roy, and A. Nag, *Advances in User Authentication*: Springer, 2017.

[4] P. Gupta and P. Gupta, "Multi-biometric Authentication System using Slap Fingerprints, Palm Dorsal Vein and Hand Geometry," *IEEE Transactions on Industrial Electronics*, 2018.

[5] S. Ghouzali, "Watermarking based multi-biometric fusion approach," in *International Conference on Codes, Cryptology, and Information Security*, 2015, pp. 342-351.

[6] O. Nafea, S. Ghouzali, W. Abdul, and E.-u.-H. Qazi, "Hybrid multi-biometric template protection using watermarking," *The Computer Journal*, vol. 59, pp. 1392-1407, 2016.

[7] R. Thanki and K. Borisagar, "Multibiometric Template Security Using CS Theory–SVD Based Fragile Watermarking Technique," *WSEAS Transactions on Information Science and Applications*, vol. 12, pp. 1-10, 2015.

[8] R. Thanki and K. Borisagar, "Biometric watermarking technique based on cs theory and fast discrete curvelet transform for face and fingerprint protection," in *Advances in Signal Processing and Intelligent Recognition Systems*, ed: Springer, 2016, pp. 133-144.

[9] R. M. Thanki, V. J. Dwivedi, and K. R. Borisagar, "Multibiometric Watermarking Technique Using Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT)," in *Multibiometric Watermarking with Compressive Sensing Theory*, ed: Springer, 2018, pp. 91-113.

[10] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE,* vol. 87, pp. 1062-1078, 1999.

[11] D. L. Mahajan and S. A. Gogate, "Overview of Digital Watermarking and its Techniques," *KHOJ: Journal of Indian Management Research and Practices,* pp. 197-204, 2016.

[12] M. Ulker and B. Arslan, "A novel secure model: Image steganography with logistic map and secret key," in *Digital Forensic and Security (ISDFS), 2018 6th International Symposium on*, 2018, pp. 1-5.

[13] B. Anand and M. P. K. Shah, "Face Recognition using SURF Features and SVM Classifier," *International Journal of Electronics Engineering Research. ISSN,* pp. 0975-6450, 2016.

[14] W. Meng, D. S. Wong, S. Furnell, and J. Zhou, "Surveying the development of biometric user authentication on mobile phones," *IEEE Communications Surveys & Tutorials,* vol. 17, pp. 1268-1293, 2015.

[15] A. Mahfouz, I. Muslukhov, and K. Beznosov, "Android users in the wild: Their authentication and usage behavior," *Pervasive and Mobile Computing,* vol. 32, pp. 50-61, 2016.

[16] L. Sobrado and J.-C. Birget, "Graphical passwords," *The Rutgers Scholar, an electronic Bulletin for undergraduate research*, vol. 4, pp. 12-18, 2002.

[17] R. Spolaor, Q. Li, M. Monaro, M. Conti, L. Gamberini, and G. Sartori, "Biometric Authentication Methods on Smartphones: A Survey," *PsychNology Journal*, vol. 14, 2016.

[18] M. Harbach, A. De Luca, and S. Egelman, "The anatomy of smartphone unlocking: A field study of android lock screens," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 4806-4817.

[19] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation computer systems*, vol. 16, pp. 351-359, 2000.

[20] H. Saevanee and P. Bhatarakosol, "User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile device," in *Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on*, 2008, pp. 82-86.

[21] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM computing surveys (CSUR)*, vol. 35, pp. 399-458, 2003.

[22] E. Jose, M. Greeshma, M. H. TP, and M. Supriya, "Face recognition based surveillance system using facenet and mtcnn on jetson tx2," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, 2019, pp. 608-613.

[23] M. He, J. Zhang, S. Shan, M. Kan, and X. Chen, "Deformable face net: Learning pose invariant feature with pose aware feature alignment for face recognition," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, 2019, pp. 1-8.

[24] T. Kanade, "Picture processing system by computer complex and recognition of human faces," 1974.

[25] L. Wiskott, J.-M. Fellous, N. Krüger, and C. Von Der Malsburg, "Face recognition by elastic bunch graph matching," in *International Conference on Computer Analysis of Images and Patterns*, 1997, pp. 456-463.

[26] P. J. Phillips, S. Z. Der, P. J. Rauss, and O. Z. Der, *FERET (face recognition technology) recognition algorithm development and test results*: Army Research Laboratory Adelphi, MD, 1996.

[27] K. Shrivastava, S. Manda, P. Chavan, T. Patil, and S. Sawant-Patil, "Conceptual Model for Proficient Automated Attendance System based on Face Recognition and Gender Classification using Haar-Cascade, LBPH Algorithm along with LDA Model," *International Journal of Applied Engineering Research*, vol. 13, pp. 8075-8080, 2018.

[28] G. Du, F. Su, and A. Cai, "Face recognition using SURF features," in *MIPPR 2009: Pattern Recognition and Computer Vision*, 2009, p. 749628.

[29] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, pp. 346-359, 2008.

[30] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, pp. 143-152, 2009.

[31] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC*, 2015, p. 6.

[32] F. Schroff, D. Kalenichenko, and J. 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.

[33] F. Zhang, S. Xin, and J. Feng, "Combining global and minutia deep features for partial high-resolution fingerprint matching," *Pattern Recognition Letters*, vol. 119, pp. 139-147, 2019.

[34] R. Ramachandra, M. Stokkenes, A. Mohammadi, S. Venkatesh, K. Raja, P. Wasnik, *et al.*, "Smartphone Multi-modal Biometric Authentication: Database and Evaluation," *arXiv preprint arXiv:1912.02487*, 2019.

[35] C. Shoemaker, "Hidden bits: A survey of techniques for digital watermarking," *Independent study, EER*, vol. 290, pp. 1673-1687, 2002.

[36] X. Ling, M.-B. Hu, R. Jiang, R. Wang, X.-B. Cao, and Q.-S. Wu, "Pheromone routing protocol on a scale-free network," *Physical Review E*, vol. 80, p. 066110, 2009.

[37] K. Gurney, *An introduction to neural networks*: CRC press, 2014.

[38] O. Niculaescu, "Neural networks and how machines learn meaning," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 25, pp. 64-66, 2019.

[39] O. Eluyode and D. T. Akomolafe, "Comparative study of biological and artificial neural networks," *European Journal of Applied Engineering and Scientific Research*, vol. 2, pp. 36-46, 2013.

[40] S. K. Esser, R. Appuswamy, P. Merolla, J. V. Arthur, and D. S. Modha, "Backpropagation for energy-efficient neuromorphic computing," in *Advances in Neural Information Processing Systems*, 2015, pp. 1117-1125.

[41] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013, pp. 1058-1066.

[42] S. Sharma, "Activation functions in neural networks," *towards data science,* vol. 6, 2017.

[43] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853,* 2015.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, pp. 1137-1149, 2017.

[45] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems,* vol. 1, pp. 111-122, 2011.

[46] M. Sheinfeld, S. Levinson, and I. Orion, "Highly accurate prediction of specific activity using deep learning," *Applied Radiation and Isotopes,* vol. 130, pp. 115-120, 2017.

[47] S.-C. Wang, "Artificial neural network," in *Interdisciplinary computing in java programming*, ed: Springer, 2003, pp. 81-100.

[48] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *International Conference on Machine Learning*, 2019, pp. 1675-1685.

[49] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, *et al.*, "Learning to learn by gradient descent by gradient descent," *arXiv preprint arXiv:1606.04474*, 2016.

[50] J. He, J. Rexford, and M. Chiang, "Design for optimizability: Traffic management of a future internet," in *Algorithms for Next Generation Networks*, ed: Springer, 2010, pp. 3-18.

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

[52] X. Glorot and Y. 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.

[53] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, ed: Elsevier, 1992, pp. 65-93.

[54] I. K. Sethi and A. K. Jain, *Artificial neural networks and statistical pattern recognition: old and new connections* vol. 11: Elsevier, 2014.

[55] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85-117, 2015.

[56] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep features learning for medical image analysis with convolutional autoencoder neural network," *IEEE Transactions on Big Data*, 2017.

[57] F. Huang, J. Zhang, C. Zhou, Y. Wang, J. Huang, and L. Zhu, "A deep learning algorithm using a fully connected sparse autoencoder neural network for landslide susceptibility prediction," *Landslides*, vol. 17, pp. 217-229, 2020.

[58] S. Nag, "Lookahead optimizer improves the performance of Convolutional Autoencoders for reconstruction of natural images," *arXiv preprint arXiv:2012.05694*, 2020.

[59] G. K. Wallace, "The JPEG still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, pp. xviii-xxxiv, 1992.

[60] S. Shang, Y. Zhao, and R. Ni, "Double JPEG detection using high order statistic features," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*, 2016, pp. 550-554.

[61] A. Gore and S. Gupta, "Full reference image quality metrics for JPEG compressed images," *AEU-International Journal of Electronics and Communications*, vol. 69, pp. 604-608, 2015.

[62] S. Naresh, B. V. Kumar, and G. Karpagam, "A literature review on quantization table design for the JPEG baseline algorithm," *International Journal of Engineering And Computer Science*, vol. 4, pp. 14686-14691, 2015.

[63] A. Raid, W. Khedr, M. A. El-Dosuky, and W. Ahmed, "Jpeg image compression using discrete cosine transform-A survey," *arXiv preprint arXiv:1405.6147*, 2014.

[64] Y. Choi, M. El-Khamy, and J. Lee, "Variable rate deep image compression with a conditional autoencoder," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3146-3154.

[65] M. G. Alaslani and L. A. Elrefaei, "Convolutional neural network based feature extraction for iris recognition," *International Journal of Computer Science & Information Technology*, vol. 10, pp. 65-78, 2018.

[66] R. M. Thanki, V. V. Dwivedi, and K. R. Borisagar, "A watermarking technique using discrete curvelet transform for security of multiple biometric features," *arXiv preprint arXiv:1701.04185*, 2017.

[67] M. Douglas, "Fingerprint watermarking using svd and dwt based steganography to enhance security," 2015.

[68] H. Nguyen-Thanh, T. Le-Tien, and T. Nguyen-Duy, "Multiple Watermarking with Biometric Data Using Discrete Curvelets and Contourlets," *Journal of Image and Graphics,* vol. 6, 2018.

[69] O. N. A. AL-Allaf, S. A. AbdAlKader, and A. A. Tamimi, "Pattern recognition neural network for improving the performance of iris recognition system," *Journal of Scientific and Engineering Research,* vol. 4, pp. 661-667, 2013.

[70] F. N. Sibai, H. I. Hosani, R. M. Naqbi, S. Dhanhani, and S. Shehhi, "Iris recognition using artificial neural networks," *Expert Systems with Applications,* vol. 38, pp. 5940-5946, 2011.