Ministry of Higher Education and Scientific Research
University of Babylon
College of Sciences for Women
Department of Computer Science

A Digital Image Encryption System Based on Multi-dimensions Hyper-Chaotic and Convolutional Neural Networks

A Thesis

Submitted to the Council of College of Sciences for Women, University of Babylon in a Partial Fulfillment of the Requirements for the Degree of Master in Science\ Computer Sciences

By

Noor Haider Abd-Ali Witwit

Supervisor

Ali Yakoob Al-Sultan

2023 A. D. 1445 A. H.

بسم الله الرحمن الرحيم

صدق الله العلي العظيم

المجادلة: ١١

Supervisor Certification

I certify that this thesis entitled "**A Digital Image Encryption System based on Multi-Dimensions Hyper-Chaotic and Convolutional Neural Network**" was done by (Noor Hayder Abd-Ali) under our supervision.

Signature:

Name: Asst. Prof. Dr. Ali Yakoob: / / 2023

Address: University of Babylon/College of Science for Women

The Head of the Department Certification

In view of the available recommendations, I forward the dissertation entitled "**Digital Image Encryption System based on Multi-dimension Hyper-chaotic and Convolutional Neural Network (CNN)**" for examining committee.

Signature:

Name: Dr. Saif M. Kh. Al-Alak

Date: / / 2023

Address: University of Babylon/College of Science for Women

Dedications

I dedicate the fruit of my humble effort to those who gave me life, hope, and upbringing with a passion for learning, learning, and knowledge, and those who taught me to ascend the ladder of life with patience, wisdom, benevolence, righteousness, and loyalty to them, my dear father, my dear mother.

To whom God gave me the blessing of his presence in my life to the strong knot and who was a helper for me in my thesies journey and supported me as we pave the way together towards success in our scientific career to my companion on my path, my beloved husband

To the seed of benefits and hope for tomorrow, my beloved daughters, Rihanna, and Joanna

To my brothers and sisters is the source of my pride in life

To my dear brother Ameer K. Jawad

Finally, to everyone who helped me from near or far and had a role in completing the study, I ask the Lord Almighty to reward everyone with the best reward in this world and the Hereafter.

Then, to every seeker of knowledge who strives with his knowledge to benefit humanity in general and Islam and Muslims in particular with all that God has given him of knowledge and knowledge.

Noor 2023

Acknowledgments

All thanks and praise to Allah, the Lord of the world, who gave me courage and enabled me to achieve this work.

My thanks and gratitude go to my supervisor Asst. Prof. Dr. Ali Yakoob for the support and guidance they have given me and the effort and time to complete this thesies.

Thanks, and Gratitude to all my professors and all the staff of the Department of Computer Sciences \ College of Sciences for Women \ University of Babylon for their help.

Noor 2023

Abstract

This thesis presented a digital image encryption system based on multi-dimensions Hyper-chaotic and “convolutional neural network (CNN)” techniques to maintain data security and privacy concerns in digital image data. The system leverages the strengths of multi-dimensional Hyper-chaotic encryption algorithms and CNNs to achieve high security while preserving the integrity and diagnostic value of images. The thesis investigates existing encryption techniques and identifies their limitations in terms of security, efficiency, and image quality preservation. The proposed system integrates the multi-dimension Hyper-chaotic encryption algorithm and CNN-based techniques into a comprehensive framework, evaluating its performance through extensive experiments and evaluation metrics.

The proposed system presented a component that tests the Hyper-chaotic signal based on a convolutional neural network to ensure it is chaotic before inserting it into encryption methods. Also presented were two methods of Digital image encryption: The first method using a chaotic hashing technique, this approach encrypts the pixel bits of a digital image. The second encryption method for images is encrypted using three Hyper-chaotic generators using this method.

The thesis used the measurement testing of Lyapunov exponents to check the Hyper-chaotic signal and NIST testing to check the randomness of chaos. Also, to evaluate the encryption and decryption on images, mean sequence error (MSE), peak signal-to-noise ratio (PSNR), correlation coefficients (Corr), structural similarity index measure (SSIM), and entropy were used.

The simulation results shown by the CNN-Tester achieve nearly 100% accuracy in testing and classifying the Rabinovich system signals. The delay

to test by Lyapunov exponents is about 1.5077 seconds for one set of parameters, while CNN-Tester has a delay of about 0.0112 seconds. Testing chaotic signals using CNN-Tester yields better classification results and is faster than Lyapunov exponents.

For the 1^{st} method, row-column and one-vector scrambling methods had the best encryption results, but the key space in one-vector is less than the key space of row-column. The 2^{nd} method demonstrated higher encryption performance than the 1^{st} method. The 2^{nd} method offers better security and efficiency, making it a more promising choice for image encryption. The key space of 1^{st} method is about 2^{548} and about 2^{1096} in the row-column case, while the key space in the 2^{nd} method is about 2^{1644}.

List of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviations	The Mean of Abbreviations
3DES	Triple Data Encryption Algorithm
Acc	Accuracy
AES	Advanced Encryption Standard
ANN	Artificial Neural Network
CNN	Convolution Neural Network
Corr	Correlation Coefficient
CRIN	Chaotic Random Integer Numbers
CRP	Generation Chaotic Random Position
D	Diagonal
Dec.	Decryption
DES	Data Encryption Standard
DNN	Deep Neural Networks
DSA	Digital Signature Algorithm
E	Entropy
EI	Encrypted Image
Enc.	Encryption
FN	False Negative
FP	False positives
GPUs	general-purpose graphic processing units
H	Horizontal
HCRS	Hyper Chaotic Rabinovitch System
HCS	Hyper-Chaotic Sequence
IDEA	Individuals with Disabilities Education Act
IE3HS	Image Encryption Based on 3 Hyper-chaotic Signals (2^{nd} encryption method)

To be continued

IEHSB	Image Encryption Based on Hyper-chaotic Scrambling Bits (1st encryption method)
LUT	Scrambler Look-Up Table
DI	Digital Color Image
MLP	multiple hiddenlayers
MSE	Mean Square Error
NIST	Ntional Institute of Standards and Technology
Pre	Precision
PRNG	Pseudo-Random Number Generator
PSNR	Peak Signal-to-Noise Ratio
RC4	Rivest Cipher 4
ReLU	Rectified Linear Unit
RSA	Rivest-Shamir-Adleman
Sec	Second
Sen	Sensitivity
Spe	Specificity
SSIM	Structural similarity index measure
SVM	Support Vector Machines
TN	True Negative
TP	True Positives
TRNG	Genuine Random Number Generator
V	Vertical

List of Symbols

Symbol	Meaning
$x_n, y_n, z_n, and\ w_n$	state vectors of Ribanovich system
$f(\cdot)$	iterative function
$\dot{x}(t)$	state vector
K, n	number of classes
Zj	production corresponding to class j
y, $\hat{y}$	represents the true value and predicted value
λ	Lyapunov number
h	pixel filter (or sub-region)
Rj	Rectified Activation Maps
$(x\ ,\ y)$ or (A , B)	pixel values of the original and the encrypted images
p($\boldsymbol{i}$)	represents probability in bits
$\mu_x\ , \mu_y$	Average of of x and y pixels images respectivly
σ_x^2, σ_y^2	variance of x and y pixels images respectivly
σ_{xy}	Covariance between x and y pixels images.
Y(i), Y(i+1)	The present and the next states, ,
h	The step size
F(X)	The chaotic differantional equation

Chapter One

General Introduction

1.1 Introduction

With the emergence of the big data era, images have become crucial carriers of information. Consequently, the security of digital images during transmission and storage has garnered significant attention, particularly in aerospace, the military, healthcare, and Internet of Things. Breaches of security related to digital images can have severe consequences for individuals, organizations, and nations[1]. As a result, protecting the security of digital images has become an urgent matte [2]. An increasing number of people are using digital cameras, smartphones, and other mobile devices to access and transmit digital image information, which may encompass crucial data such as facial photos, images, satellite maps, and architectural drawings of important national institutions. However, during transmission, security issues including information leakage and malicious use can arise due to technical failures or illegal user attacks [3]. Various measures are required to safeguard the security of digital images. First, encryption and decryption need to be strengthened to ensure protection against attacks and theft during transmission and storage [4]. Second, access control mechanisms should be reinforced to ensure that only authorized users can access and transmit the images [5] . Chaotic systems are increasingly employed in encryption applications due to their unpredictability and sensitivity to initial conditions, which enable the generation of long-term, unpredictable pseudo-random sequences. These properties make chaotic systems an effective means of encryption. certain encryption algorithms employ simple heteroskedastic operations to encrypt plaintexts, which are vulnerable to selective attacks. To enhance security, more complex encryption algorithms, such as permutation- and diffusion-based structures should be employed to strengthen their resistance. In recent years, encryption has emerged as a crucial method for safeguarding information security. However, traditional image encryption algorithms have limitations in randomness,

unpredictability, and security [6]. To address these limitations, an image encryption algorithm that combines multi-dimensional Hyper-Chaotic Systems and Convolution Neural Networks (CNN) is proposed.

1.2 Problem Statements

In symmetric encryption systems, the same key is used to encrypt and decrypt data. This means that if an attacker can obtain the encryption key, they can decrypt the encrypted data. To prevent this, it is important to change the encryption key regularly. Changing the encryption key can be a complex and time-consuming process. It requires all parties involved in the communication to agree on a new key and to update their encryption and decryption systems. This can be especially challenging in large organizations with complex communication systems.

However, one of the challenges with using chaotic systems in encryption is that it is important to ensure the system is chaotic. This can be done using various methods, such as Lyapunov exponents. However, these methods can be time-consuming and computationally expensive.

In this thesis, we propose a new method for testing the chaoticity of chaotic systems using a convolutional neural network (CNN). CNNs are a type of deep learning algorithm well-suited for image processing tasks. The authors show that their CNN-based method is more efficient and accurate than traditional methods, such as Lyapunov exponents.

1.3 Thesis Aims

The primary objective of this thesis is to create a robust and effective digital image encryption system based on multi-dimensional Hyper-chaotic and Convolutional Neural Network (CNN) algorithms. The suggested system would offer high security while maintaining the diagnostic usefulness and

integrity of images. By combining the strengths of Hyper-chaotic systems and CNN, this thesis aims to overcome the limitations of existing encryption methods and offer a practical solution for securing digital image data. The main objectives of this thesis as the following:

1. Design and integrate a multidimensional hyper-chaotic encryption algorithm and CNN-based techniques into a comprehensive framework for digital image encryption.
2. To develop a fast and accurate method for testing the chaoticity of hyperchaotic signals using CNNs.
3. Developing a new digital image encryption algorithm that is more secure, efficient, and preserves image quality better than the existing techniques.
4. The performance of the proposed encryption system was evaluated through extensive experiments and evaluation metrics including Lyapunov exponents, NIST testing, MSE, PSNR, Correlation, SSIM, and entropy.
5. To investigate the impact of different parameters on the security and performance of the proposed encryption system.
6. To improve the speed of the encryption and decryption processes without compromising security.

1.4 Thesis Contributions

The contributions of the proposed digital image encryption methods based on hyper chaos and CNN and be performed as the following points:

1. To ensures the parameters and initial values giving a Hyper-chaotic signal by Rabinovitch system proposing a fasting testing method (compare with testing by Lyapunov Exponents) based on CNN tester.

2. **The first encryption method** based on digital Hyper-chaotic scrambling to change the location of bits of pixels to change the value for pixels.

3. **The second encryption method** is based on three Hyper-chaotic generators, the first chaotic system using to generate a random integer number between 1 to 255, the second and third Hyper- chaotic systems using a random selector.

1.5 Literature Review

The related works on digital image encryption use Hyper-chaos and Convolutional Neural Networks. These works cover various techniques and results and include reference citations for further exploration:

- In 2023, Chang et al. [7]: Proposed a new fractional-order seed chaotic generator is designed to solve the problem of the complex operations of single low-dimensional systems and simple high-dimensional systems. The fractional-order chaotic system generated is proven to have better chaotic performance using Lyapunov exponential differential calculus, approximate entropy, 0–1 test and other indicators. On this basis, the "multiple squares nested body scrambling (MSNBS)" model is extended from fractal theory to complete the image scrambling process, and a new algorithm is proposed to be applied to the encryption field in combination with the fractional-order coupled chaotic system.
- In 2023, Shakir et al. [8]: Proposed a new four-dimensional hyper-chaotic system has been suggested that is used in the keys generation, which are utilized in the image encryption process to achieve permutation and substitution operations. Firstly, color bands are permuted using the index of the chaotic sequences to remove the high correlation among

neighboring pixels. Secondly, dynamic S-boxes achieve the principle of substitution, which are utilized to diffuse the pixel values of the color image. The efficiency of the proposed method is tested by the key space, histogram, and so on.

- In 2023, Xiaowu Li et al. [9] Proposed an encryption and decryption framework uses the ResNet concept. ResNet's residual structure and jump connections extract deep medical image information and accelerate model convergence. A logistic chaotic system encrypts ResNet model output for security, adding unpredictability and complexity. An attention technique improves the model's responsiveness to the medical image's region of interest, boosting encrypted network security.
- In 2023, Mahalingam et al. [10] Proposed an encryption with a two-layer image encryption scheme involving bit-level encryption in the time-frequency domain. The top layer consists of a bit of plane slicing the image, and each plane is then scrambled using a chaotic map and encrypted with a key generated from the same chaotic map. Next, image segmentation, followed by a Lifting Wavelet Transform, is used to scramble, and encrypt each segment’s low-frequency components. Then, a chaotic hybrid map is used to scramble and encrypt the final layer.
- In 2022, Hayder Mazin et al.[11] proposed design and Field Programmable Gate Array implementation an algorithm for image encryption systems based on stream cipher using multi-dimensional hyperchaotic generators. In this algorithm three different hyperchaotic generators are combined together to generate new Pseudo-Random bit generator to be used for image pixels masking (encryption). Four dimensional, six dimensional, and seven dimensional hyperchaotic generators are combined together using XOR operation to generate the new random bit stream. The hyperchaotic dynamical equations are solved

by using Forward Euler integration method. The X-dynamics of each hyperchaotic system has been converted to binary stream. The binary streams are combined to generate the random bits generator to be used for encryption.

- In 2022, Alarood et al. [12] Proposed an improved shuffled confusion-diffusion based colour Image Encryption Scheme (IES) using hyper-chaotic plain images. First, five different sequences of random numbers were generated. Two of these sequences were used to mix image pixels and bits, while the remaining three were used to XOR the values of image pixels.
- In 2022, Uğur Erkan et al.[13] proposed encryption technique proposes a unique strategy that employs a new chaotic log-map and a deep convolutional neural network (CNN) model for key generation. In addition, the process of manipulation includes a bit reversal operation. By creating sensitive keys, beginning values, and control parameters for the Hyper-chaotic log-map, the system generates a chaotic sequence that is essential for safe encryption.
- In 2021, Renxiu Zhang et al. [14] Proposed combines the 6-dimensional cellular neural network (CNN) and Chen’s chaotic system. This encryption scheme belongs to symmetric cryptography. In the proposed scheme, the initial key and switching function generated by the plaintext image are first utilized to control the CNN to complete the scrambling process. Then, Chen’s chaotic system is used to diffuse the scrambled image for realizing higher security.
- In 2020, Xing Yuan Wang et al. [15] recommended a phased composite chaotic map. The map is employed as the Fisher-Yates scrambling controller because its cryptographic qualities are superior to the logistic map's. As a diffusion controller, the encrypting process employs a

fractional-order five-dimensional cellular neural network system due to its increased complexity. The message may be deciphered using the mapping, the secret key, and the plaintext.

- In 2020, Jiming Zheng et al. [16], Proposed an improved two-dimensional logistic-sine coupling map (N2D-LSCM) and an improved Henon map (NHenon) are proposed. Furthermore, by combining N2D-LSCM and NHenon map, an image encryption algorithm is proposed based on these two chaotic systems and DNA coding. The chaotic sequences generated by N2D-LSCM are used as the parameters of NHenon. In the scrambling stage, DNA encoding is carried out for pixels after scrambling by two chaotic sequences generated by N2D-LSCM; in the stage of diffusion, DNA random coding acts on random matrix obtained by two chaotic sequences generated by NHenon, and DNA XOR operation is carried out with the image obtained in the scrambling stage to diffuse.
- In 2020, Song et al. [17]: Proposed an efficient and secure quantum video encryption algorithm for quantum videos based on qubit-planes controlled-XOR operations and improved logistic map in multi-layer encryption steps. Three simple cryptosystem steps are presented in the proposed approach to accomplish the whole encryption process: inter-frame permutation, intra-frame pixel position geometric transformation, and high 4-intra-frame-qubit-planes scrambling. Firstly, the inter-frame positions of quantum video are permutated via inter-frame permutation that is controlled by the keys which are generated by improved logistic map. Secondly, intra-frame pixel positions are encrypted by intra-frame pixel position geometric transformation and improved logistic map. Finally, the high 4-intra-frame-qubit-planes are scrambled via quantum controlled-XOR operations and improved logistic map.

- In 2019, Gong, Lihua, et al. [18]: proposed The original image is first permuted by the Arnold transform to reduce the block effect in the compression process, and then the resulting image is compressed and re-encrypted by compressive sensing, simultaneously. Moreover, the bitwise XOR operation based on chaotic system is performed on the measurements to change the pixel values and a pixel scrambling method is employed to disturb the positions of pixels. Besides, the keys used in chaotic systems are related to the plaintext image.

Table 1-1: The Summarize of Related Works.

Authors	**The Main Contributions**	**Image Types**	**System Type**	**Main Encryption Results**
Chang et al. [7] in 2023	Proposed a fractional-order seed chaotic generator to address the complex operations of single low- and simple high-dimensional systems.	Normal Images	Fractional-order chaotic system	Key Space= 2^{256}, Corr. (H=−0.0003, V=0.0021, D=−0.0030), Entropy= 7.9972, NPCR= 99.6079%, UACI= 33.5046%
Shakir et al. [8] in 2023	Proposed a novel four-dimensional hyper-chaotic system for image encryption that uses permutation and substitution operations.	Normal Images	New four-dimensional hyper-chaotic system	Key Space= 2^{627}, Entropy=7.99857, Corr(H= -0.00020, V= -0.00045,D= -0.00474), NPSR= 99.6367**%**, UCAI= 33.0305**%**, MSE= 8914.693, PSNR= 3.8508
Xiaowu L et al.[9] in 2023	Proposed a Utilizes ResNet model for encryption and decryption. ResNet's residual structure and	Medical image	ResNet model	Corr(H=0.0021), Entropy=7.9887,

Authors	The Main Contributions	Image Types	System Type	Main Encryption Results
	Jump connections extract profound medical image information, speeding up convergence.			
Mahalingam et al. [10] in 2023	Proposed a two-layer image encryption scheme that uses bit-level encryption in the time-frequency domain.	Normal Images	Chaotic map	PSNR= 9.2465, Corr (H= −0.002153, V= −0.0000901, D= −0.0006059), NPCR= 99.6230**%**.
Hayder Mazin et al.[11] in 2022	Proposed an image encryption algorithm using multi-dimensional hyperchaotic generators, combining three generators (4D, 6D, 7D) with XOR operations to create a new Pseudo-Random bit generator for pixel masking. The hyperchaotic equations are solved with Forward Euler integration, converting their X-dynamics to binary streams.	Normal Images	Four Dimensional System Six Dimensional System Seven Dimensional System	PSNR= 6.9019, MSE= 1.3271e+04, Corr= 0.0263, Entropy= 7.945 7, NPCR= 99. 34, UACI= 33. 4.
Alarood et al. [12] in	Proposed an enhanced shuffled confusion-diffusion-based color	Normal Images	ive dimensional (5D) hyper-	Corr= 0.000732, Entropy= 7.9997, PSNR= 7.61, MSE= 11,258,

Authors	The Main Contributions	Image Types	System Type	Main Encryption Results
2022	image encryption scheme that uses five different sequences of random numbers.		chaotic system	
Uğur Erkan et al. [13] in 2022	Proposed encryption scheme uses chaotic log-map, deep CNN, and bit reversion for key generation and manipulation.	Normal Images	chaotic log-map	Entropy=7.9994, Corr(H=-29 × 10^{-5}, V=21× 10^{-5}, D=33 × 10^{-6}),
Renxiu Zhang et al. [14] in 2021,	The proposed method combines a 6D CNN with Chen's chaotic system for symmetric encryption. It uses the initial key and plaintext image's switching function to control CNN scrambling, followed by Chen's chaotic system for enhanced security.	Normal Images	Chen's chaotic system	Corr=(H= 0.0073, V= 0.0134, D= 0.0034 ,Entropy= 7.9997, NPCR=99.6059, UCAI= 33.4151)
Xing yuan Wang et al. [15] in 2020	Proposed a phased composite chaotic map for image encryption that uses the Fisher-Yates scrambling controller and a fractional-order five-dimensional cellular neural network system.	Normal Images	Chaotic map	Corr (H= -0.0002, V= 0.0011, D= 0.0965), Entropy=7.9989, NPCR= 99.63**%**, UACI= 2 33.45**%**

Authors	The Main Contributions	Image Types	System Type	Main Encryption Results
Jiming Zheng et al.[16] in 2020	Proposed an image encryption technique that is based on chaotic systems and DNA coding.	Normal Images	Two-dimensional logistic-sine coupling map (N2D-LSCM) and an improved Henon map (NHenon)	Key Space=10^{70} Entropy=7.99923, Corr (V= -0.0064, H= -0.0022, D= -0.023)
Song et al. [17] in 2020	Proposed a multi-layer quantum video encryption scheme that uses controlled XOR operations on a qubit plane and an improved logistic map.	Normal Images	Logistic map	Entropy= 7.9878, Corr (V= 0.0112, H−0.0084, D= 0.0044)
Gong, Lihua, et al. [18] in 2019	Proposed an image encryption and compression scheme that uses compressive sensing and chaotic systems.	Normal Images	Chaotic Map	Corr (H=0.0016, V= 0.0081, D= -0.0016), NPCR= 99.6201%, UACI= 33.5247%,
Renxiu Zhang et al. [14] in 2019	Proposed 6D Cellular Neural Network (CNN) and Chen's chaotic system combined for secure symmetric cryptography. Initial key and switching function control CNN to	Normal Images	Chen's chaotic system	Corr(H= 0.0084,V= 0.0133,D= 0.0055), NPCR= 99.6124%, UACI= 33.4492%, Key space= 6×2^{192}

Authors	The Main Contributions	Image Types	System Type	Main Encryption Results
	scramble the plaintext image, followed by Chen's chaotic system for increased security.			

1.6 Structure of the Thesis

The remaining sections of this thesis are structured as follows:

Chapter Two examines the existing encryption algorithms for images, their limitations, and the state of the art in digital image encryption.

Chapter Three describes the proposed multi-dimensional Hyper-chaotic encryption method, including its design, implementation, and integration with CNN-based approaches.

Chapter Four discusses the experimental setup and methodology used to evaluate the performance of the proposed encryption system, evaluation metrics, comparative analysis, the results, and an analysis of the experiments conducted to demonstrate the effectiveness and efficiency of the proposed digital image encryption system.

The conclusion of the proposed system and ideas for further development are discussed in Chapter Five.

Chapter Two

Theoretical Background

2.1 Introduction

In this chapter, the basic concepts of chaos, types of chaos, and Lyapunov exponents, the general image encryption methods (changing the position of pixels, changing the value of pixels, and changing the values and positions of pixels) will be presented. Also, this chapter introduces the basic concepts of machine learning and deep learning and more details about the 1-dimensional Convolutional Neural Network (CNN).

2.2 Chaos-based Cryptography

To encrypt and decrypt data, chaos-based cryptography practitioners use the characteristics of chaotic systems. In chaos theory, thesiesers examine complex and unpredictable systems with chaotic behavior highly sensitive to initial conditions. Several natural phenomena, including weather patterns, fluid dynamics, and population dynamics, are examples of chaotic systems [19], [20], [21].

To create cryptographic keys or encrypt and decode data, chaos-based cryptography uses the chaotic behavior of specific systems. Since it is difficult for an adversary to anticipate or evaluate the encryption process without knowing the initial conditions or key, chaotic systems are well-suited for cryptography applications. Chaotic maps and iterations, in which a chaotic function is repeatedly applied to a plaintext message or a cryptographic key, are prominent techniques in chaos-based cryptography. The output is then encrypted using the same chaotic system and related settings, yielding ciphertext. Complexity in the chaotic system's behavior gives this encryption method its security, making it hard to crack for an opponent [22].

The benefits of chaos-based encryption include increased security and less vulnerability to statistical assaults. It facilitates a speedier encrypting and decrypting procedure compared to standard cryptographic techniques. The fascinating field of chaos-based cryptography investigates the potential of chaotic systems for secure communication and data protection. However, it faces challenges such as the need for precise control over chaotic systems, vulnerability to attacks exploiting system parameters, and susceptibility to synchronization attacks. Ongoing studies attempt to overcome chaos-based encryption systems' limits and security problems while developing more robust and efficient encryption algorithms [23].

2.3 A Comparison of Symmetric and Asymmetric Key Encryption

Symmetric and asymmetric encryption are two fundamental cryptographic techniques for modern secure digital communication. The key differences between these encryption algorithms are explained in Table 2.1.

Table 2.1: A Comparison of Keys that are Symmetric and Asymmetric.

Difference Points	Symmetric Encryption	Asymmetric Encryption
Key usage	Uses a single secret key for both encryption and decryption.	Encrypts using a public key and decrypts with a private key.
Key distribution	The secret key must be exchanged securely between the sender and the recipient.	The receiver must maintain the confidentiality of the private key, while the public key may be freely transferred.
Security	Less secure comber the asymmetric encryption.	More secure because the private key is never transmitted or shared,

Difference Points	Symmetric Encryption	Asymmetric Encryption
		reducing the risk of interception or attack.
Speed	Is faster and more efficient than asymmetric encryption because it requires less processing power to encrypt and decrypt data.	Low speed.
Applications	Large amounts of data, including files and messages, are frequently encrypted using this method.	Used for key exchange, digital signature verification, and secure comm. Protocols.
Key size	Uses shorter key sizes.	Uses longer key sizes.

2.4 Image Encryption Algorithms

Changing the pixel's value and changing the pixel's position are two kinds of image encryption utilizing a chaotic encryption system. The first employs a chaotic system as a random number generator, with the resulting sequence performing a specific operation on the plain text to produce ciphertext, which alters the pixel's value. The other is used to adjust the coordinates of pixels that are affected by random chaos. From the created sequence, the chaotic systems construct a scrambling matrix [24].

2.4.1 Pixel Value Transform

Pixel value transforming (changing pixel value) is the first type of image encryption used to transfer the plaintext to a ciphertext by changing the value of the pixels. As shown in **Error! Reference source ot found.**, A represents the original image with a dimension size of M*N

with S as gradation layers (for RGB, S=3). When using a chaotic system for encryption, initial conditions and system parameters must be set as an encryption key used for generating the ciphertext. Then, the chaotic sequence will be generated. Because of the randomness characteristic of the chaotic system, the encryption process will be achieved by performing such as XOR bitwise or hashing bits of pixels of the image, and the encrypted image will appear as pseudo-random data. For image decryption (in other words, obtaining plaintext from ciphertext), the XOR operation must be applied to the pixel value of the ciphertext by using the same initial conditions and parameters of the same chaotic systems that have been used for the encryption process [25].

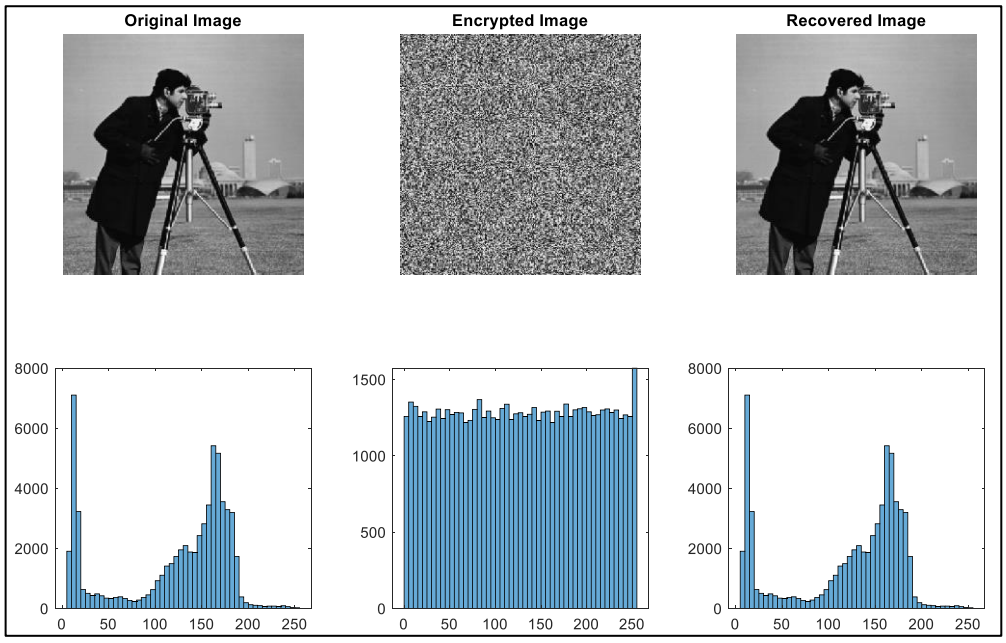


Figure 2.1:Image Encryption Based on Pixel value changing[25].

2.4.2 Transform of the Pixel Position (Scrambling)

The ciphertext is produced by altering the locations of the pixels rather than their values, which is the second kind of digital image encryption procedure. To create the cipher image, a two-dimensional chaotic map is used to move each pixel of the plaintext to a new location.

This type of encryption is also called image scrambling, as shown in Figure 2.2. Presently, many encryption algorithms using chaos are either a simple transforming pixel position or a simple pixel's value changing [26].

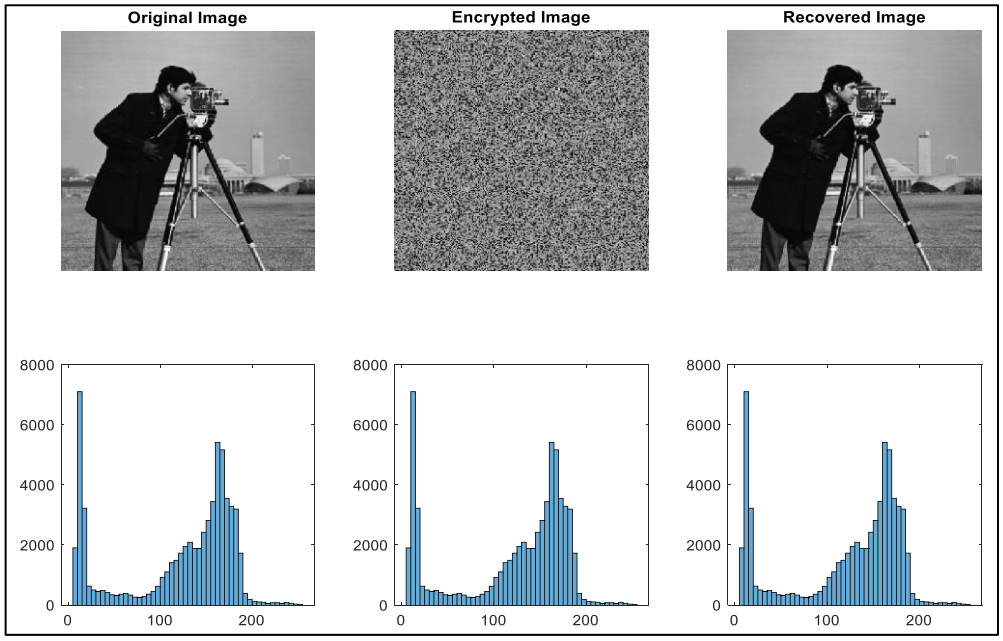


Figure 2.2: Image Encryption Based on Pixel Position Changing [26].

Image pixel scrambling has useful intruders' opposition when attacking the ciphertext, but it is considered weak for known plain text attacks because it only changes the position without modifying the pixel's values [27].

Most chaotic encryption technology is buildup with low dimensions systems such as one or two dimensions which are not good enough to create a confidential image encryption system. To enhance the security of these systems, many methods can be considered, such as:

1) Designing chaotic systems with long periods means developing chaotic encryption systems with high dimensions to design a vigorous encryption system.

2) Merge transforming pixel positions with the changing pixel value methods for a step-by-step image encryption process, increasing the difficulty of attacking the ciphertext. This method is beneficial for high-security matters. However, the disadvantage of this method is increasing the processing time [28].

2.5 Chaotic Signals

Chaotic signals are dynamical, non-periodic signals that resemble random noise and come from non-linear processes. The state variables in a dynamical system typically have a fixed number of independent states, and the movements or trajectories of each of these states are controlled by a set of equations that include all the state variables. Chaotic state variables in dynamical systems fluctuate in a limited, non-periodic, random-like manner [29].

They also possess a characteristic known as sensitive dependence on initial conditions, which denotes the ease with which any two close-by initial conditionsmay result in two entirely uncorrelated state variable movements or trajectories. With the use of various initial conditions from the same system, this feature enables the creation of an unlimited number of chaotic signals that are not associated [30].

2.6 Lyapunov Exponents

Lyapunov characteristic exponent of this strategy is taken as a measuring element for all components of chaos and sensitivity reliance on initial conditions. Lyapunov normal exponents can be utilized for measuring the detachment of two close directions as far as initial conditions. The division $\delta(t)$ of such two directions is quicker with the development of time [31].

It may retain two points, $x_n(t)$ and $x_n(t) + \Delta x_n(t)$; they are on the attractor at time t, so that firstly:

$$x_n(t) = |x_n(t) + \Delta x_n(t)| << 1.$$

$$\Delta\,(t) \approx \Delta\,(0) e^{\lambda t}.$$

Figure 2.3 it is finding a clear definition of this principle. where λ is the Lyapunov number and $\{x_1, x_2, \ldots\ldots, x_n\}$ is the path of the *f* on the real line.

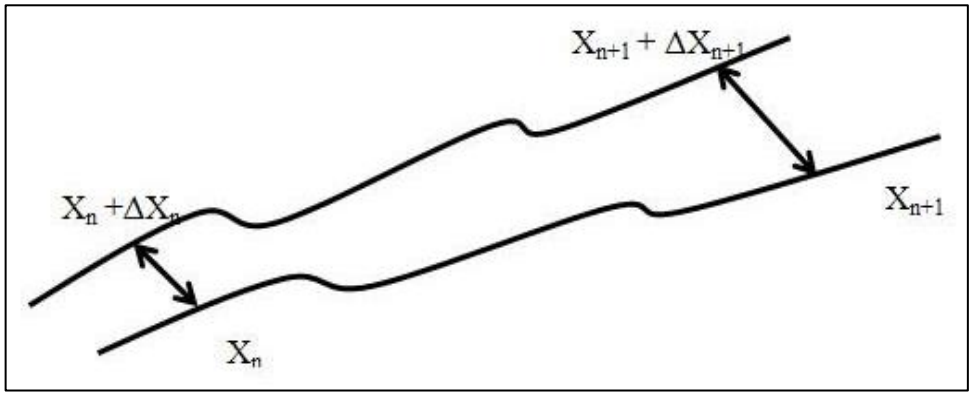


Figure 2.3: Lyapunov Exponent [32].

For 1-D maps the characteristic of the Lyapunov exponent is obtained by the section$< \log \left|\frac{df}{dx}\right| >$. This way, the quantity of exponents is equivalent to the dimensionality of the stage space ($\lambda_1, \lambda_2, \lambda_3 \ldots .\lambda_n$). The exponents are sorted out on the premise of the diminishing worth, and it is generally realized that the length of the line between directions increments as $e^{\lambda_1 t}$, the areas increment as$e^{(\lambda_1+\lambda_2)t}$, and the volume increments as $e^{(\lambda_1+\lambda_2+\lambda_3)t}$, (where *t* denotes the continuity of time in flows while for maps denotes to the iteration index).

- There are many Lyapunov exponents for a chaotic system equal in number to the dimensionality of map. There are, for instance, a single positive Lyapunov exponent for one dimensional map and two

Lyapunov exponents for the two-dimensional map as Henon map, one is negative and the other is positive. Regarding Lorenz chaotic flow, there are three exponents, positive, negative, and equal to zero.

- When Lyapunov exponent is greater than zero ($\lambda > 0$), it is used as signature of chaos.
- When Lyapunov exponent is smaller than zero ($\lambda < 0$), it means that the orbit is associated to a stable fixed point or stable periodic orbit.
- While Lyapunov exponent is equal to zero ($\lambda = 0$), the orbit is a limit orbit.
- When Lyapunov exponent has multiple positive λ, it means Hyper-chaos.

2.7 Chaotic Maps

Typically, chaotic maps are characterized by iterative or difference equations in which the system's present state is updated depending on its past state. Typically, these equations include non-linear functions, contributing to chaotic systems' complicated and unpredictable behavior. Examples of well-known chaotic maps are the logistic and Hénon maps [33]. Equation (2.1) illustrates the Chaotic Map:

$$\mathbf{x_{n+1}} = \mathbf{f}(\mathbf{x_n}) \tag{2.1}$$

Where x_n It is the state vector, and $f(\cdot)$ denotes the iterative function.

Another common type of chaotic map is the logistic map [34]. The following equation (2.2) illustrates a time series map produced as:

$$\mathbf{x_{n+1}} = \mathbf{1} - \mathbf{2x_n^2} \tag{2.2}$$

Figure 2.4 depict the trajectory and dynamics of the logistic map, respectively.

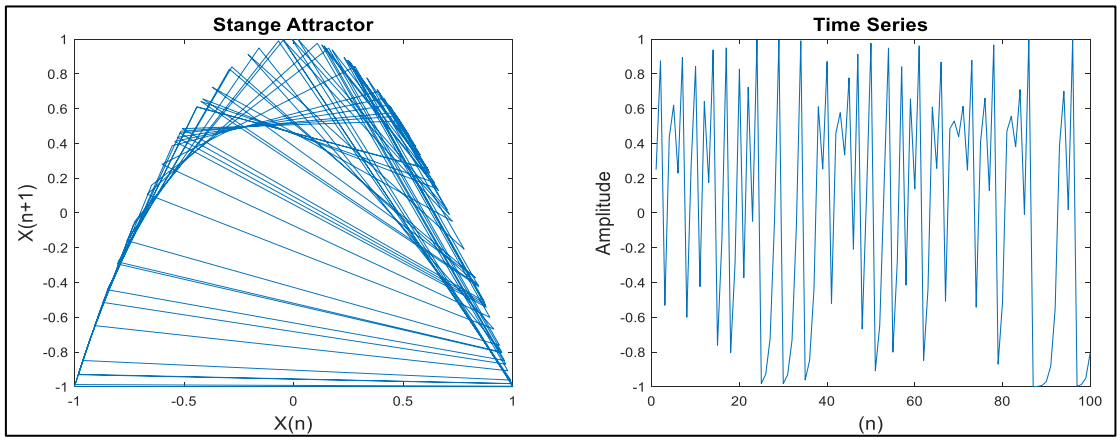


Figure 2.4: The Logistic Chaotic map (a) Strange Attracter (b)Time Series Xn Selecting of the Chaotic System [34].

2.8 Chaotic Flows

The chaotic flow signal is generated from a set of differential equations, and chaotic flow itself is a continuous-time system [35].

The following equation (2.3) illustrates Chaotic Flow:

$$\dot{\mathbf{x}} = \frac{\mathbf{d_x}}{\mathbf{d_t}} = \dot{\mathbf{x}}(\mathbf{t}) \tag{2.3}$$

Where, $\frac{d_x}{d_t}$ *and* $\dot{x}(t)$ is the state vector of the system at the given time (t) in the dynamical system is a function of any chaotic flows [36]. Famous flows include the Rössler System, the Lorenz System, Chen's System, the Chua System, and the Lü System [36]. Rössler chaotic flow strange attracter and time series are illustrated in Figure (2.5.a) and (2.5.b).

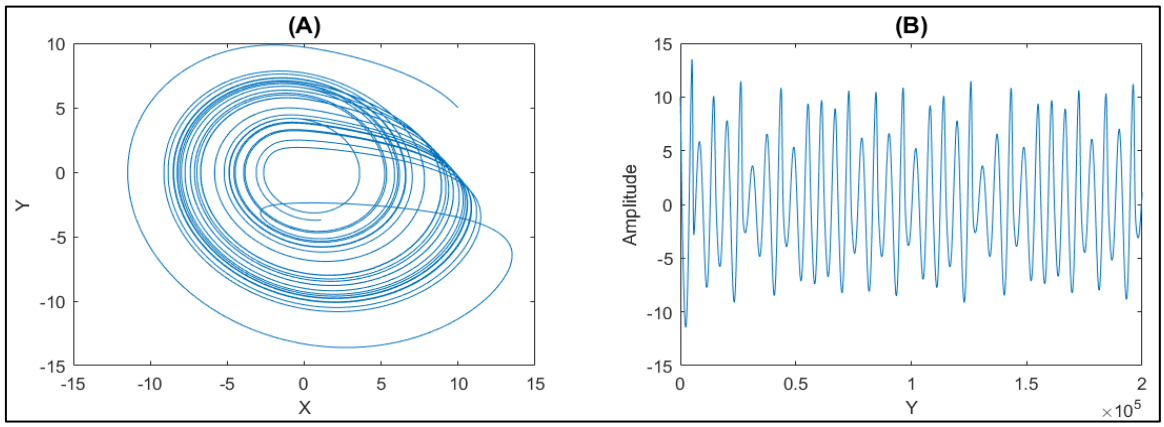


Figure 2.5: The flow of the Ressler System (a) The Weird Attractor, and (b) the Time Series x (t) [36].

2.8.1 Hyper Chaotic System

One of its defining characteristics is that Hyper-chaos is very sensitive to both the initial conditions and the parameters of the system. Even though the two connected chaotic systems' initial conditions were identical, the sensitivity characteristics led to the courses of the two systems diverging rapidly [37].

A dynamical model is Hyper-chaotic if it contains two or more positive Lyapunov exponents; this means that the dynamics of such systems will grow in several directions simultaneously. Table 2.2 is illustrate the differences between the normal and Hyper-chaotic.

Table 2.2: Chaos vs. Hyper-chaotic

Features	Chaos	Hyper-Chaos
Dimensions	At least 1	At least 4
Number of positive Lyapunov exponents	At least 1	At least 2
Examples	Logistic Map Tent Map Lorenz system	Chens Hyper Chaotic system Hyper Chaotic Lorenz type system Hyper Chaotic Rabinovich System
Complexity	Less Complex	More Complex

2.8.2 Hyper Chaotic Rabinovitch System (HCRS)

A continuous-time hyper chaotic autonomous system has at least four dimensions, with positive Lyapunov exponents of at least two. As a result, it benefits from increasing the system's randomness and unpredictability, which is significant in the field of secure communication or image encryption. The HCRS is made up of the following equations [38].

$$\begin{aligned} \dot{x} &= hy - ax + yz \\ \dot{y} &= hx - by - xz \\ \dot{z} &= -dz + xy + w^2 \\ \dot{w} &= xy + cw \end{aligned} \quad (2.4)$$

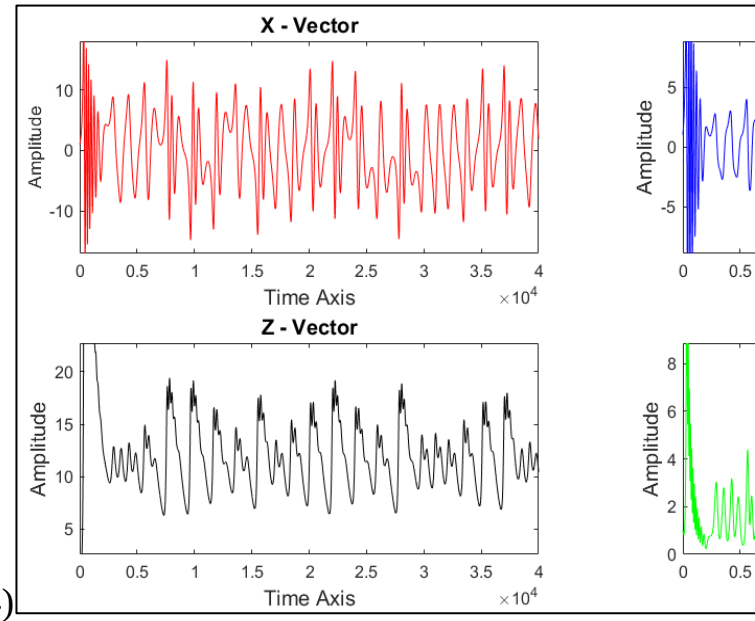


Figure 2.6 shows the time series for each vector, the system parameters constrain the form of the chaotic attractor. We can observe the studied chaotic system's irregular hyper attractors simply by looking at it in Figure 2.7.

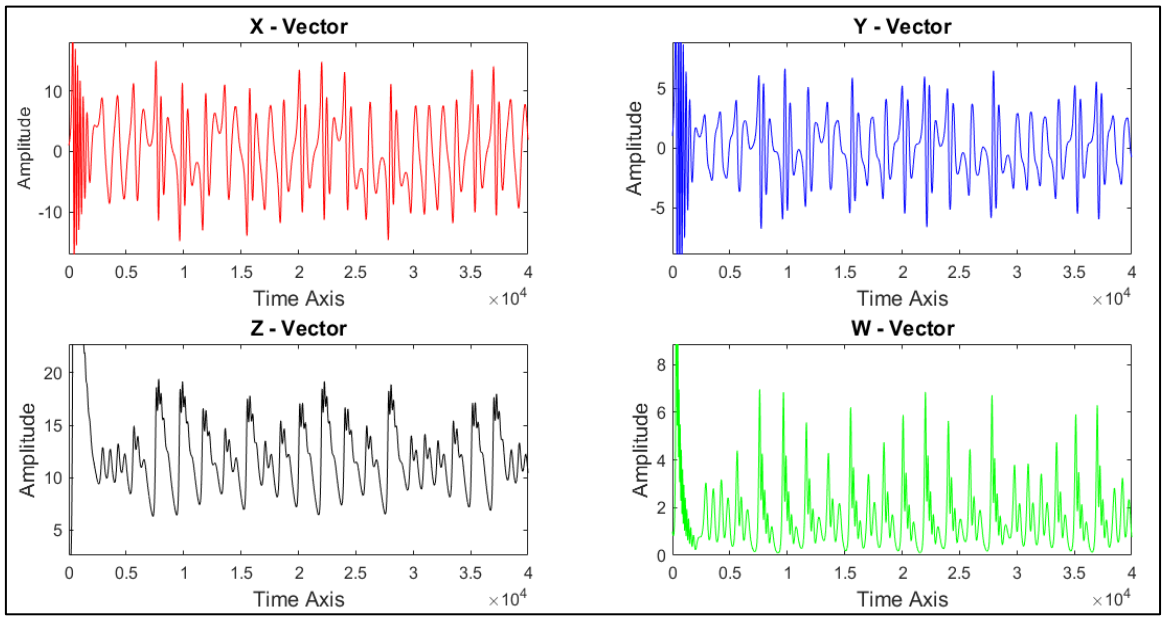


Figure 2.6: Time Series for all Vectors to Rabinovich System [38].

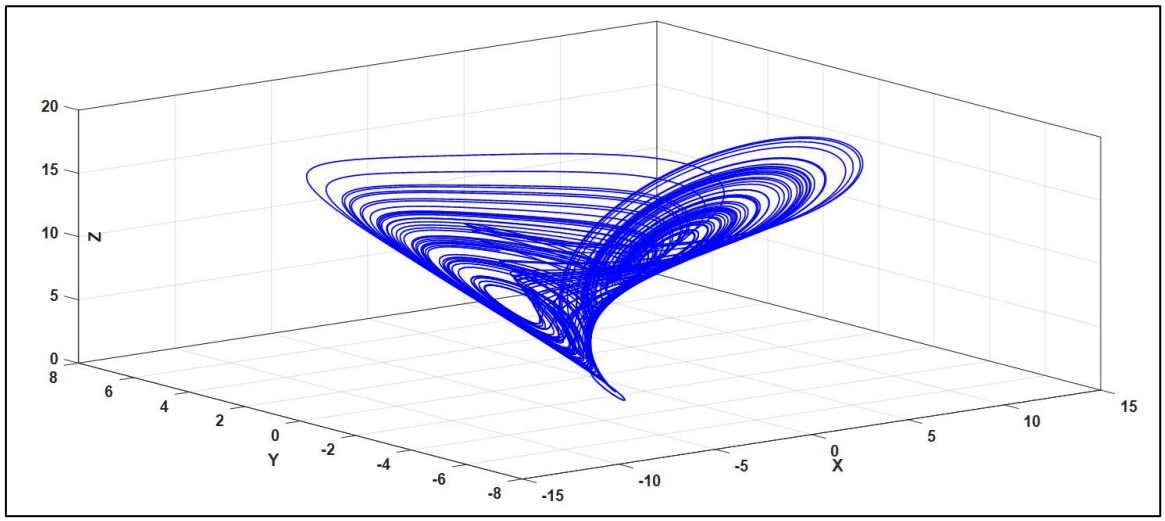


Figure 2.7: Rabinovitch System Portraits in three Dimensions [39].

The Rabinovitch model (eq. (2.4)) shows that one of the trajectories is closely connected to another yet has quite distinct properties. It is sensitive to the system's initial conditions and parameters. In Figure 2.8, the state's time-domain waveform x(t) is shown, with the blue line representing the starting conditions (x_0, y_0, z_0, *and* w_0) that are equal to (1,1,1and1) and the red line representing the initial conditions (x_0, y_0, z_0, *and* w_0) that are equal to (1,1,1.000001 and 1) respectively. The system behaves in a manner that is different depending on the initial

conditions. Similar to this, little adjustments to the parameters produce a completely different time series [40].

To generate Hyper-Chaotic signal by Rabinovitch system as in algorithm (2.1)

Algorithm (3.1) Generate Hyper-chaotic Signals from the Rabinovitch System and Euler Method.

Inputs: chaos signl initial conditions X0, Y0, Z0, W0,
chaos signl parameters a, b, c, d, r.
h: The step size of the Euler Method, (h = 0.001)
m: the length of chaotic signal.

Output: Pseudo Random Sequences (X, Y, Z, & W).

Begin

1: Set n=1

2: While n <= m do

3: $X_{n+1} = x_n + h(ry_n - ax_n + y_n z_n)$
4: $Y_{n+1} = y_n + h(rx_n - by_n - x_n z_n)$
5: $Z_{n+1} = z_n + h(-dz_n + x_n y_n + w_n^2)$
6: $W_{n+1} = w_n + h(x_n y_n + cw_n)$

7: Increment n by 1

8: end while

9: Getting a Hyper-chaotic signals

End

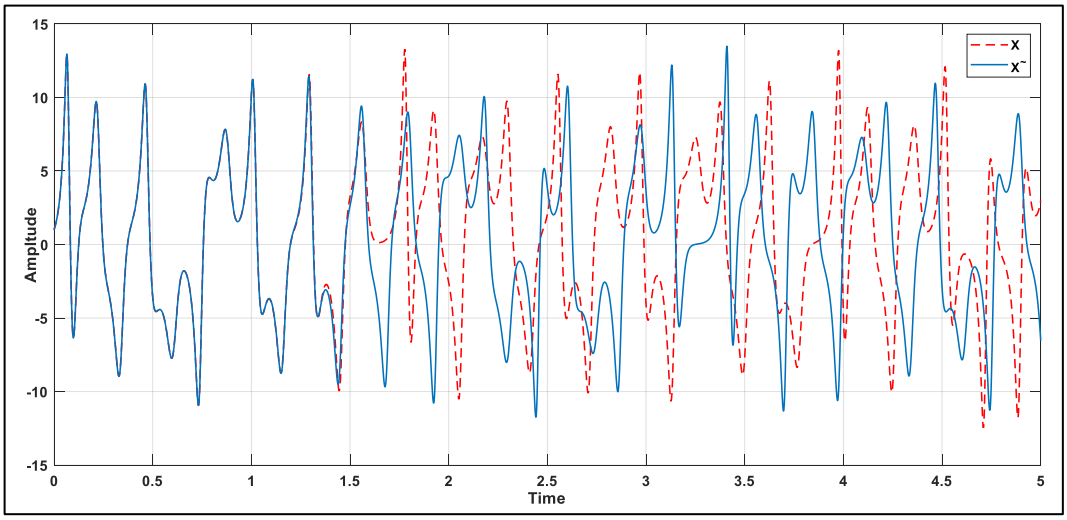


Figure 2.8: Sensitive Rabinovich System to the Initial Condition [40].

2.9 Machine Learning Models

One area of artificial intelligence that relies on computer science, statistics, and mathematics is machine learning. The basic goal of machine learning and statistical modeling is to enable computer programs to learn from data and then make appropriate decisions based on the information that a prior experience or prior skills have learned. The machine learns directly from the fundamental input data structure and becomes more intelligent [34].

There are two basic subcategories of machine learning approaches: supervised learning and unsupervised learning.

2.9.1 Supervised Learning

Models are trained in supervised learning based on input data (X) and output data (y), to forecast the future outputs of the unseen input, where it is supervised or monitored learning based on output data or label the ground truth. Depending on this label, the algorithm frequently operates forecasts until it reaches the level of acceptable performance. General algorithms in supervised learning are divided into two main categories:

A. Classification: is a group of data related to each other where the intention is to divide the points of this data into a collection of predefined categories based on some characteristics of the data such as Neural Network, Support Vector Machines, k-nearest-neighbors, Random Forest, Naive Bayes.

B. Regression: is used to forecast real values that are referred to as continuous values such as Decision Trees, Linear Regression, Assembly methods [37].

Notice that it is called supervised because the algorithm can be guided towards the correct answer during the training phase by using the target values (y) in the data set [41]**.**

2.9.2 Unsupervised Learning

Common algorithms within unsupervised learning are the clustering problem algorithm such as k-mean, the algorithm of A priority for association rule learning, and the dimensional reduction algorithm. Models are trained in unsupervised learning based on input data only, without these inputs being labeled i.e., the model is not given the ground truth label during training. It is the exact opposite of supervised learning, where input data is divided into a group of elements that share the same attribute [38].

2.10 The Activation Functions

The activation function must be present for a neural network to learn and accomplish difficult tasks. The output value of activation functions often ranges between [-1,1] and [0,1], with some being linear and others being non-linear. Additionally, some activation functions are linear, while others are non-linear. The sigmoid function equation (2.5) illustrates and the Rectified Linear Unit (ReLU) function in equation (2.6), are the activation functions that are used in the hidden layers most of the time; this can be seen in Figure (2.9) [42].

$$A = \frac{1}{1+e^{-x}} \tag{2.5}$$

where e ≈ 2.71 is the base of the natural logarithm

$$g(z)=\max(z,0) \tag{2.6}$$

ReLU is a non-linear function that substitutes all image pixels whose value is negative in the activation map with zero value. This leads to the advantage of calculation speed and reduces the occurrence of overfitting [40].

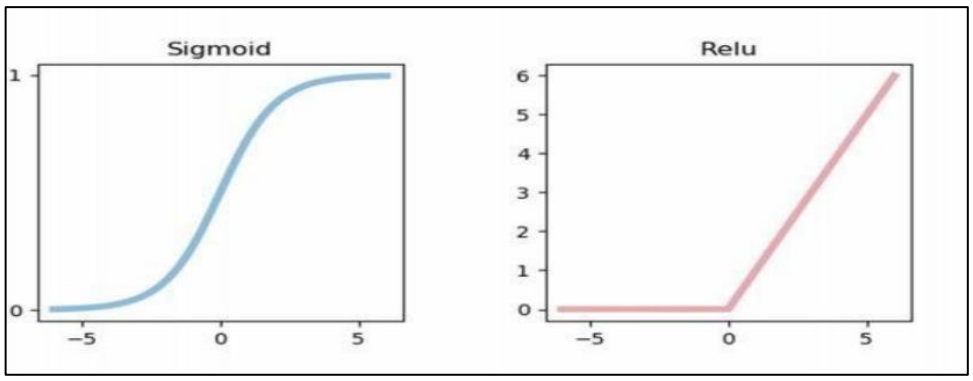


Figure (2.9): The Activation Functions [40].

Whereas in the output layer, the softmax function can be used as an activation function that is frequently employed to compute probabilities

and perform multi-class classifications because soft max ranges between (0 and 1) and has the value of 1 if all the features are applied using the following formula: (2.7) [43].

$$\boldsymbol{softmax}(\boldsymbol{x_i}) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \boldsymbol{for} \;\; \boldsymbol{j = 1 \ldots K} \tag{2.7}$$

Where K is the number of classes, **Zj** is the production corresponding to class j

2.11 Loss Function

The cost function (also referred to as the loss function) measures a neural network's ability to predict the presented date's right output. Several used loss functions in machine learning, such as the mean squared error (MSE) and cross-entropy [44].

The cross-entropy is the most specific loss function of classification since it determines the classification algorithms perform depending on the probability of the class falls within the range (0, 1) and is described as following equation (2.8):

$$\boldsymbol{loss} = -\sum_{i=1}^{n} \boldsymbol{y_i} \log(\hat{\boldsymbol{y}}_i) \tag{2.8}$$

Where n represents the number of classes, y represents the true value, and $\hat{y}$ represents the predicted value[45].

The cross-entropy cost and the sigmoid function are called the binary cross-entropy loss and are used when there are only two class classifications (0,1).

In addition to the Soft-max function, the cross-entropy cost is called categorical cross-entropy loss and is used where two or more label classes exist (labels are given in the one-hot expression). The Sparse

Categorical Cross-entropy loss function is used if the labels are presented as integers [46].

2.12 Deep Learning

The deep learning technique is a subset of machine learning that simulates complex abstract concepts in data using a multi-layer architectural design, most commonly from neural networks, and non-linear transforms in its algorithms. The goal of using such techniques is to achieve "real" artificial intelligence, whichmeans that a machine can learn how to perform extremely complex tasks comparable to how the human brain works through layers of neurons. Computationally, construction and training and deep learning are intensive. Still, recent developments in applications based on general-purpose graphic processing units (GPUs), the rapid advancement of machine learning algorithms, the processing of signals and information, and the growing amount of data that is used in training are all reasons that have increased the popularity and success of deep learning [47]. Figure (2.10) demonstrates a deep nervous system structure consisting of several layers that make it deep[48].

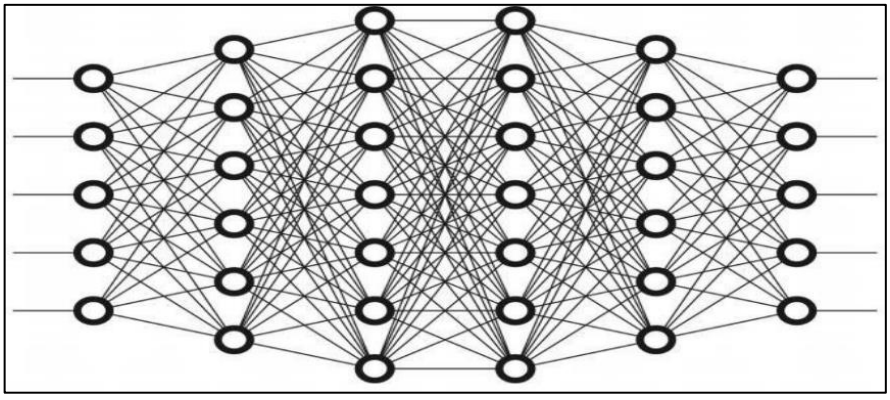


Figure 2.10: Shows Deep Network Architecture [48].

2.12.1 Deep Neural Networks (DNN)

The DNN is an artificial neural network that is made up of multiple hiddenlayers. MLP is the most general ANN architecture that is used for DNN. Neural networks are made up of interconnected neurons with several levels of connections. DNN needs long calculation periods to implement, and many data feeds at training samples, so the number of weights in these networks will reach thousands or equal millions. There are different architectures for deep learning networks, the most important of which are: convolution Neural Network (CNN), which is used to classify images, and Recurrent Neural Network (RNN), used with texts and continuous data [49].

2.12.2 Convolution Neural Networks (CNN)

Like the multi-layer Perceptron, CNN is one of the deep neural network types most frequently used in machine vision fields. The distinction resides in the fact that it can mix many layers that are locally linked for feature extraction with several levels that are fully connected for classification [50]. It is a useful tool for object identification, image visualization, and signal detection. CNN are the most widely utilized neural networks in AI for Deep.

The neural network architecture known as one-dimensional convolutional neural networks, or 1D CNN, is designed especially for processing sequential input. While traditional CNN are well-known for their success in image recognition tasks, 1D CNN excel in analyzing data with a sequential nature, such as time series, audio signals, and text [51]. 1D CNN have demonstrated great success in a wide range of applications. In time series analysis, they can be used for tasks such as forecasting, anomaly detection, and signal processing. In audio analysis, 1D CNN are

effective in speech recognition, speaker identification, and music classification. In natural language processing, they can be employed for sentiment analysis, text classification, and named entity recognition [52].

The advantages of using 1D CNN include their ability to automatically learn relevant features from raw data, and capturing local and global dependencies within the sequence. They can handle variable-length inputs and effectively model temporal or sequential relationships. Additionally, 1D CNN benefit from the vast knowledge and techniques developed for traditional CNN, making them a powerful tool for analyzing sequential data[53]. Three distinct layers make up the 1D CNN standard architecture, as shown in Figure (2.11), and Any Convolution Neural Network model is built from these different layers: as explained:

1. Convolution layer.

2. Max pooling layer (or Sub Sampling layer).

3. Fully Connected Layer (Classification layer).

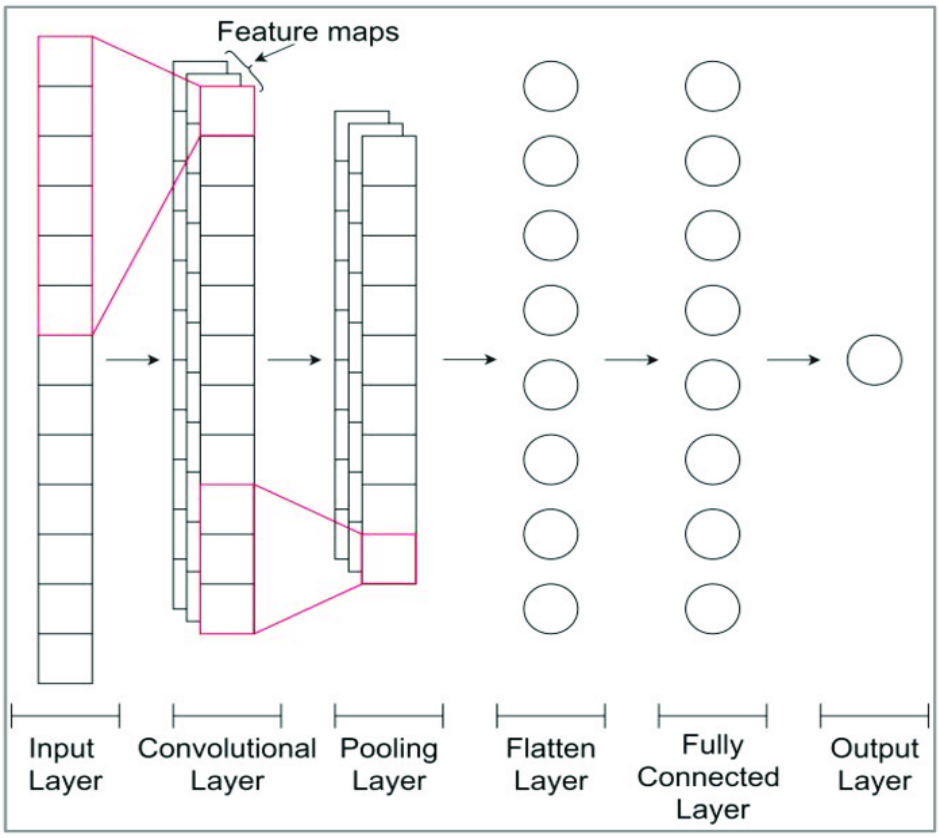


Figure 2.11: Demonstrates the Structure of a CNN [54].

2.12.2.1 Convolutional layers:

The term "convolution" refers to a mathematical process involving merging two functions to get a third function. When it takes place, two different collections of data are joined.

In order to generate a feature map from the input data, 1D CNN first apply a convolutional layer, sometimes referred to as a filter or kernel [53], as shown in Figure 2.12.

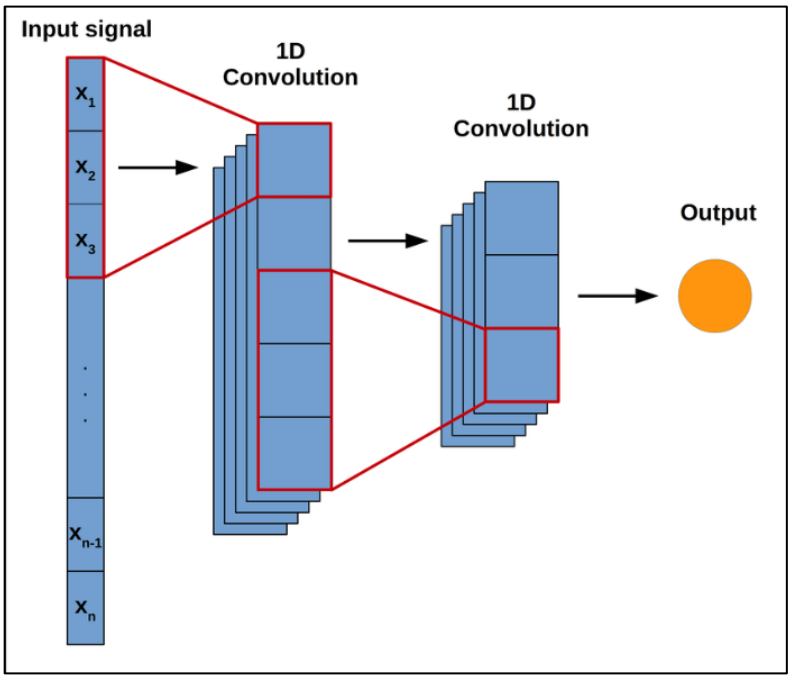


Figure 2.12: The Filter Executes its Output on the New Layer while Sliding over the Input [53].

A filter matrix and a portion of the input signals matrix are subjected to a multiplication that takes the form of a dot product. On the feature map, the output value, also known as the "target pixel," equals the sum of the components found in the resulting matrix. The feature map is finished when the filter moves over the input matrix and repeats the dot product multiplication with each remaining combination of areas. Many filters are applied to one input, and then the feature maps generated by those filters are merged to produce the final output of one convolutional layer [53]. Equation (2.9) describes a general convolution layers process frimages.

$$\boldsymbol{Z_{ij}} = (\boldsymbol{X} * \boldsymbol{K})_{\boldsymbol{i,j}} \sum\sum\sum \boldsymbol{X_{i+1,j+a,b}} \boldsymbol{K_{i,a,b}} \tag{2.9}$$

In the above equation, the convolution process between the input X and the Filter weights K in convolution layer l to produce the feature map Z [55].

Following every convolution layer, the results pass through the activationfunction (ReLU) non-linear [56].

Three parameters determine the size of the feature map: **Depth, strides,** and **padding**.

- Depth:

The depth represents the number of filters implemented in the convolution process. If the original signal was convoluted using three filters, the activation maps 'depth' will equal three.

- Strides:

Represents the number of pixels in the filter matrix that leaves during the convolution process. If a stride equals 1, the kernel will move by 1. If it equals 2, the kernel moves by 2. As the number of steps increases, the function maps diminish.

- Padding:

Convolutional neural networks (1D CNN) often employ padding to preserve the spatial dimensions of input data while performing convolutional operations. Padding is applied throughout the input sequence length in the case of a 1D CNN. Padding is necessary when you want the output sequence from the convolutional layer to have the same length as the input sequence. It helps to avoid information loss at the edges of the input when applying convolutional operations. Padding also assists in maintaining spatial information and enables the network to learn from the entire input sequence.[57]. Two types of padding are commonly used in 1D CNNs: 'valid' and 'same.'

1. **Valid Padding**: With valid padding, no padding is added to the input sequence. As a result, the output sequence will have a smaller length than the input sequence.
2. **Same Padding**: With the same padding, the input sequence is padded symmetrically on both ends to ensure that the output sequence has the same length as the input sequence. [58].

2.12.2.2 Pooling layer (also called subsampling or down sampling).

Pooling makes a given mapping less dimensional while emphasizing its key qualities. In order to lower the dimension of the convolution output, pooling is often used after the convolution layer. Additionally, it lessens overfitting. Max pooling is the most well-known pooling method [59]. Following each pooling operation, a filter of size f is dragged across the input with a stride of length s to pick up the maximum value. This technique is known as max pooling, as shown in Figure 2.13.

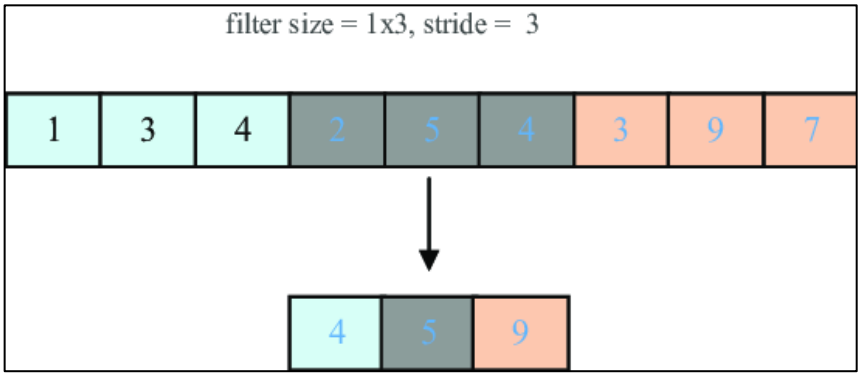


Figure 2.13: Types of Pooling [59].

The max-pooling down-sampling process is described in equation (2.10).

$$S_i = \max_{i \in R_i} h(i) \tag{2.10}$$

Where h represents some pixel filter (or sub-region) R_j from therectified Activation Maps [58].

2.12.2.3 Fully Connected Layer (Classification layer).

Classification occurs at entirely interconnected levels. A collection of fully connected layers similar to the fully connected ANN architecture are used to handle the input matrix after flattening it into a column vector [60].

The output Dense layer is processed using Softmax, while each completely connected layer (also known as a Dense layer) is delivered via an activation function. "Dense layer" refers to the processed layer (such as tanh or ReLU). Within the Softmax multi-class classification context, cross-entropy serves as the loss function [60].

The output of the SoftMax function is a vector with a size of N, where N is the number of possible classes that the 1D CNN must choose one from. The integers that make up this N-dimensional vector each represent the possibility that an signal is a member of a certain category. A signal has a chance of 10 percent of belonging to class 2, a chance of 10 percent of belonging to class 3, a chance of 75 percent of belonging to class 4, and a chance of 5 percent of belonging to class 10. For example, if the output vector reads [0.1.1.75 0 0 0 0 0.05], then there is a 10% possibility that the signal in question is part of class A [61].

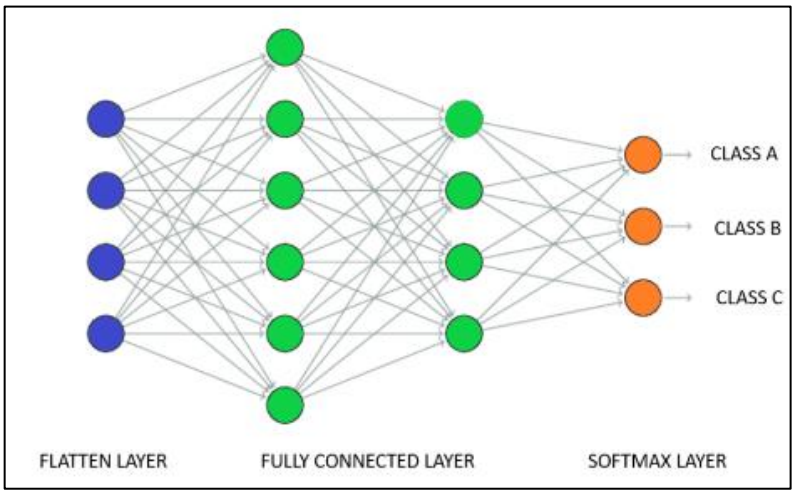


Figure 2.14: Fully Connected Layer [60].

2.12.3 Training the Convolutional Neural Networks.

The fundamental building block of ANN, backpropagation, is more challenging in CNN due to the variety of layers included. The first step of this method, forward propagation, involves training the network from the inputs of the first layer to the last layer and calculating the error between the outputs and the desired value using the loss function. This method calculates gradient descent for all network weights in two steps. The second method is referred to as backward propagation, and it starts from the network's bottom layer and moves up to the top layer [62].

2.12.3.1 Forward Propagation

Training samples are spread across all network layers from input to output using various equations that depend on the layer type to output a prediction value.

- The convolution layer in the forward propagation implements a convolution process between all its inputs and filters using equation (2.9). Then the output is passed to the ReLU function.
- Max pooling layer in the forward propagation is explained in the equation (2.10).
- A fully Connected Layer in the forward propagation works in a similar way to the MLP network that is illustrated in equation (2.11).

$$s = \bar{X}\bar{W} = \sum^{n} x_i \, w_i + b \quad (2.11)$$

Where a neuron computes the dot multiplication between the input vector X=[x1, x2,..., xn] with their identical weights W=[w1,w2,...,wn] and then adds the bias value (*b*) as in the formula (2.11). Finally, the value (s) is transferred to the activation function. The Soft Max function is shown in the equation (2.6) was used to assess the probability of the object in this layer. After that, the error between the desired value and the output is calculated using the loss function that has been defined in the equation (2.8) [63].

2.12.3.2 Backward Propagation

Backpropagation, or backward propagation of errors, is an important algorithm for training convolutional neural networks (CNN). It updates the network's weights based on the calculated error during the forward pass [64] [65].

This process is repeated iteratively, with each iteration consisting of a forward pass to calculate the output and error and a backward pass to update the weights and biases. The network learns to approximate the

desired output through repeated iterations better and improve its performance [66].

2.12.4 Regularization Techniques

Several techniques have been proposed to prevent the problem of overfitting, called regularization.

2.12.4.1 Dropout

The dropout randomly drops a set of neurons with a predetermined probability value at each training iteration. This technique greatly improved neural networks' performance to solve the overfitting problem and was introduced by Nitish Srivastava and Geoffrey Hinton in 2014 [67], as shown in Figure (2.15).

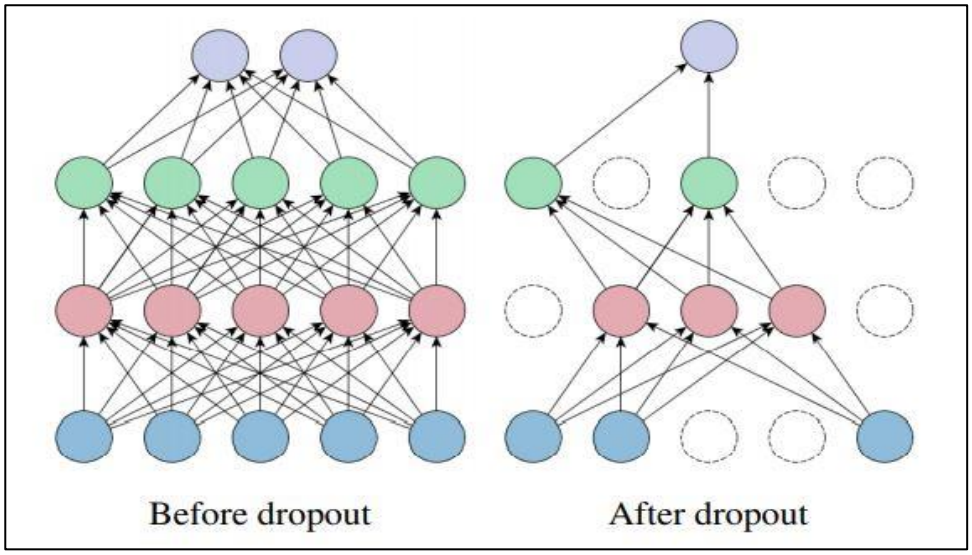


Figure 2.15: shows the dropout influence in a network [67].

2.12.4.2 Early Stopping

When training a deep learning network, one of the most effective regularization techniques is called early stopping, and it is used to determine the optimal amount of training epochs [68]. The training process must be stopped after a few subsequent periods using early stopping, as shown in Figure (2.16). A forward pass determines the

output and error, while a backward pass updates the weights and biases. This procedure is performed recursively. The network learns to approximate the intended output more accurately and enhance its performance via repeated rounds [69].

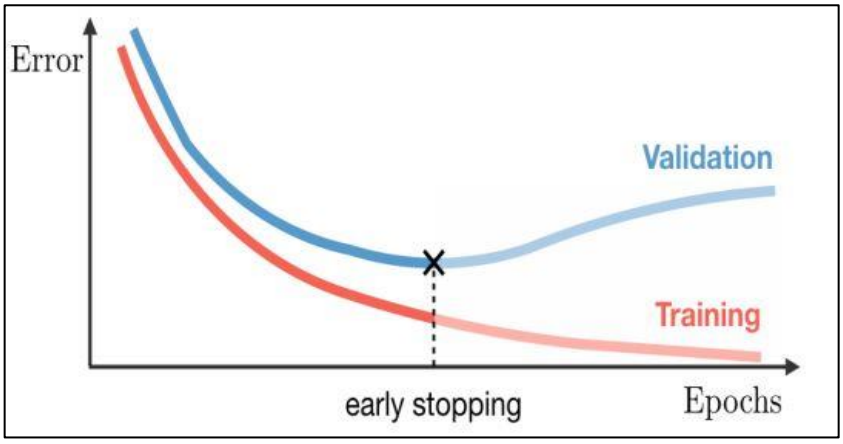


Figure 2.16: The Early Stopping Technique [69].

2.12.5 Optimization Algorithm (Adaptive Moment Estimation (Adam))

The output error of the loss function may be reduced thanks to the optimizer, which does this by adjusting the weights and bias values inside the model. The Adam Optimizer is the most important of the many performance optimization approaches used in deep neural networks.

Adam refers to "Adaptive Moment Estimation," a tool that determines the adaptive learning rates for each parameter and keeps track of the average exponential decay of earlier square gradients. Adam beats every other optimization technique, making it ideal for neural networks that need rapid convergence and high levels of complexity [70].

2.13 Performance Measures

2.13.1 Testing the Classification Algorithms

Several criteria have been employed to assess the performance of the classification algorithms:

1. The accuracy is measured by how many positive or negative cases were properly categorized.

$$\text{Accuracy (Acc)} = \text{TP}+\text{TN}/\text{TP}+\text{TN}+\text{FP}+\text{FN} \quad (2.11)$$

2. Sensitivity is the accuracy with which positive samples are correctly identified.

$$\text{Sensitivity (Sen)} = (\text{TP}/\text{TP}+\text{FN}) \quad (2.12)$$

3. Precision Testing the genuine positive from the Expected positives gives information on the accuracy of the model's performance.

$$\text{Precision (Pre)} = (\text{TP}/\text{TP}+\text{FP}) \quad (2.13)$$

4. Specificity is the percentage of identification of negative examples rightly.

$$\textit{Specificity}\ (\text{Spe}) = (\text{TN}/\text{TN}+\text{FP}) \quad (2.14)$$

Table 2.3 shows the matrix of the classifier system

Table 2.3: Confusion Matrix of Classifier System.

		Predict	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Four parameters record the prediction error:

- The positive states accurately classified as positive are called True Positives (TP).
- The Negative states mistakenly classified as positive conditions are false positives (FP).
- The Negative states accurately classified as Negative are called True Negative (TN).
- The positive states mistakenly classified as Negative conditions are called False Negative (FN).

2.13.2 Encryption's Quality Measurement Tests

Image encryption relies on visual monitoring for feature skulking, but measuring algorithm strength is limited. Scientists propose algorithms to compare images after encryption with before encryption, comparing pixel values and positions. Techniques include Correlation Coefficient Measuring Factor (Corr), Entropy, PSNR, Structural similarity index measure (SSIM), and mean square error (MSE).

2.13.2.1 The Correlation Coefficient (Corr)

The correlation coefficient is a statistical analysis used to determine the relationship between random variables or data sets, such as image processing and encryption. It measures the similarity between images and determines the strength of encryption algorithms. A zero value indicates a complete difference, while a value of one indicates the encryption process failed to hide the original image's details.

$$\boldsymbol{corr} = \frac{\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{N}((x_i - E(x)))^2}\sqrt{\sum_{i=1}^{N}((y_i - E(y)))^2}} \qquad (2.15)$$

where $E(x) = \frac{1}{N}\sum_{i=1}^{N} x(i)$, x and y are the pixel values of the original and the encrypted image, respectively [71].

2.13.2.2 Entropy

Information Entropy Analysis measures randomness and encryption quality by calculating the entropy of plain and cipher images and comparing them using the equation.

$$\boldsymbol{E = \sum_{i=0}^{2^n-1} \left[p(i) * log_2 \left(\frac{1}{p(i)} \right) \right]} \quad (2.16)$$

The symbol p($\boldsymbol{i}$) represents probability in bits, and the maximum entropy for images with 256 gray levels is 8; this is an ideal case of randomness. In the encryption process, the encrypted image's entropy should ideally equal 8, confirming the degree of predictability. To resist entropy attacks, the entropy of the encrypted image should be close to the maximum value.[72]

2.13.2.3 Structural Similarity Index Measure (SSIM)

SSIM is a method for predicting the perceived quality of digital images and videos. It measures image similarity using a full reference metric based on an initial uncompressed or distortion-free image. The SSIM index is calculated on various windows of an image, measuring between two windows of common size N*N.

$$\boldsymbol{SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)}} \quad (2.17)$$

μ_x , μ_y are the average of x and y respectivly.

σ_x^2, σ_y^2 the variance of x and y , σ_{xy} The covariance of x and y.

$c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$ two variables to stabilize the division with a weak denominator.

L the dynamic range of the pixel values (typically is $2^{bits\ per\ pixel} - 1$)

$k_1 = 0.01$, $k_2 = 0.03$ by default [73].

2.13.2.4 Mean Square Error (MSE)

The most popular metric estimate used to measure image quality is MSE. It is a comprehensive reference measure, and as the following equation demonstrates, the closer values are to zero, the better.

$$MSE = \frac{1}{MN}\sum_{n=0}^{N}\sum_{m=1}^{M}\big(x(n,m) - y(n,m)\big)^2 \qquad (2.18)$$

x and y represent the original, encrypted, or decrypted images [74].

2.13.2.5 Peak Signal-to-Noise Ratio (PSNR)

PSNR, which compares the pixel values between the original and encrypted images, can be used to assess the encryption scheme's strength. This equation may be used to compute it.

$$\boldsymbol{PSNR = 10 * log_{10}\left[\frac{M*N*255^2}{MSE}\right]} \qquad (2.19)$$

By comparing the pixel values between the original and encrypted images, PSNR may be used to assess the encryption scheme's strength [75].

2.13.3 The Randomness Testing

The NIST test suite, comprising more than ten statistical tests, evaluates the suggested hyperchaotic random behavior (used seven tests). These tests assess how random the generated sequence is. The test is dependent on the probability value (p-value). The areas of rejection and non-rejection are separated by a significance threshold of = 0.01, which is compared to the p-value. The sequence is rejected if the p-value is less

than 0.01 and indicates that it is not random. The sequence is recognized as random if the p-value is greater than or equal to 0.01 [1], [76], [77],

The main testing of NIST is the following:

2.13.3.1 Frequency (Monobit)

The Frequency (Monobit) test is a statistical test that evaluates the randomness of a binary data sequence, such as a stream of 0s and 1s. It checks whether the sequence has roughly equal amounts of ones and zeros, as one would anticipate from a random sequence. A deviation from an equal distribution of ones and zeros could indicate bias or non-randomness in the data

2.13.3.2 Frequency Test within a Block

The Monobit test is further upon in the Frequency Test inside a Block. Instead of analyzing the entire sequence as one unit, it divides the sequence into fixed-size blocks and then applies the Monobit test to each block; this allows the examination of localized patterns or variations in the data within smaller sequence segments.

2.13.3.3 Longest Run of Ones in a Block

Another randomness test used on binary data is the Longest Run of Ones in a Block test. The procedure entails breaking the sequence into blocks and then measuring the length of the longest run of consecutive ones inside each block. This test aids in determining whether any segment of the data has successive ones that considerably differ from what would be predicted from a random sequence.

2.13.3.4 Random Execution Variant

The Random Execution Variant is a concept used in some cryptographic techniques or algorithms, particularly in hardware-based security. It involves introducing random or unpredictable execution paths in a program or algorithm to deter certain attacks, such as timing or side-channel attacks. This randomness adds a layer of complexity, making it harder for an attacker to exploit vulnerabilities.

2.13.3.5 Longest Runs Test:

The Longest Runs Test is a statistical test used to assess the randomness of a data sequence. It focuses on the sequence's consecutive runs of the same value (e.g., runs of ones or zeros). The test checks whether the observed length of the longest runs falls within an expected range for a random sequence.

2.13.3.6 Frequency Block

In randomness testing, a Frequency Block refers to a fixed-size segment of a binary sequence. Randomness tests often divide the data into blocks to examine the distribution of ones and zeros within each block, allowing for more localized analysis.

2.13.3.7 Auto Correlation

Auto Correlation, or autocorrelation, is a statistical measure used to determine the degree of similarity between a sequence and a delayed version of itself. In other words, it assesses the correlation between a sequence and its lagged versions. In randomness testing, autocorrelation can help detect certain patterns or regularities in the data that might indicate non-randomness.

Chapter Three

The Proposed System

3.1 Introduction

Digital images are often sensitive and confidential, so it is essential to encrypt them to protect them from unauthorized access. Chaotic systems are unpredictable and sensitive to initial conditions, making them a good choice for generating random sequences that can be used to encrypt images.

This thesis discusses two encryption methods that use Hyper-chaos to encrypt images. The first method uses a chaotic hashing algorithm to encrypt the pixel bits of the digital image. The second method uses three Hyper-chaotic generators to encrypt the digital image. A CNN-Tester tests the signals to ensure it is chaotic.

The two encryption methods are secure but have different trade-offs in terms of security and efficiency.

3.2 The Proposed System

The images are encrypted using the Hyper-chaotic Rabinovitch system by two methods. The two methods change the locations and values of pixels. The first method is based on digital Hyper-chaotic scrambling to change the location of bits of pixels to change the value for pixels. The second method is based on three Hyper-chaotic generators. The first chaotic system generates a random integer number between 0 to 255, and the second and third Hyper-chaotic systems use a random selector.

To ensure the signal give a Hyper-chaotic by the Rabinovitch system, we propose a fasting testing method (compared with testing by Lyapunov

Exponents) based on a 1D Convolutional Neural Network. The proposed approach is shown in Figure 3.1.

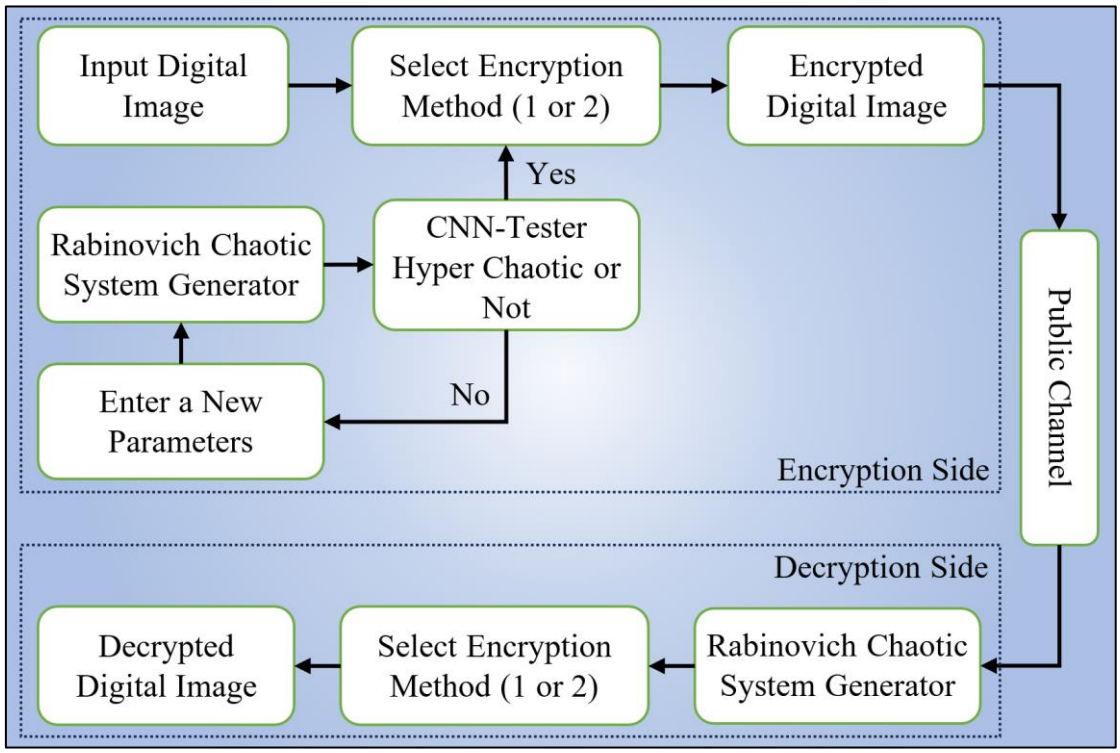


Figure 3.1: Block Diagram of the Proposed Designed System.

The proposed system in the block diagram of a digital image encryption system uses two methods based on Hyper-chaos.

3.3 Generating Hyper-chaotic Signals Based on Rabinovitch System

generate a chaotic sequence (X, Y, Z, and W) using the differential equations of the Rabinovitch chaotic system to produce a Hyper-chaotic signal based on the Rabinovitch system. Use the Euler method to solve the Ordinary Differential Equations for the Rabinovitch System.

$$\boldsymbol{Y(i) = Y(i-1) + h * F(X(i))} \quad (3\text{-}1)$$

Where Y(i), Y(i-1) are the present and the previous states, h is the step size, and F(X) is the chaotic differential equation.

3.4 Proposed Method for Testing Hyper-chaotic Rabinovitch System (HCRS) Signals Based on CNN

The proposed method follows steps to generate the dataset and classify the signal as Hyper-chaos or Not-chaos. These steps include generating Hyper-chaos and Not-chaos signals and performing the down-sampling and normalization on these signals. Then the CNN extracted relevant features, conducted the classification process, and evaluated the model's performance based on multiple criteria. See Figure 3.2.

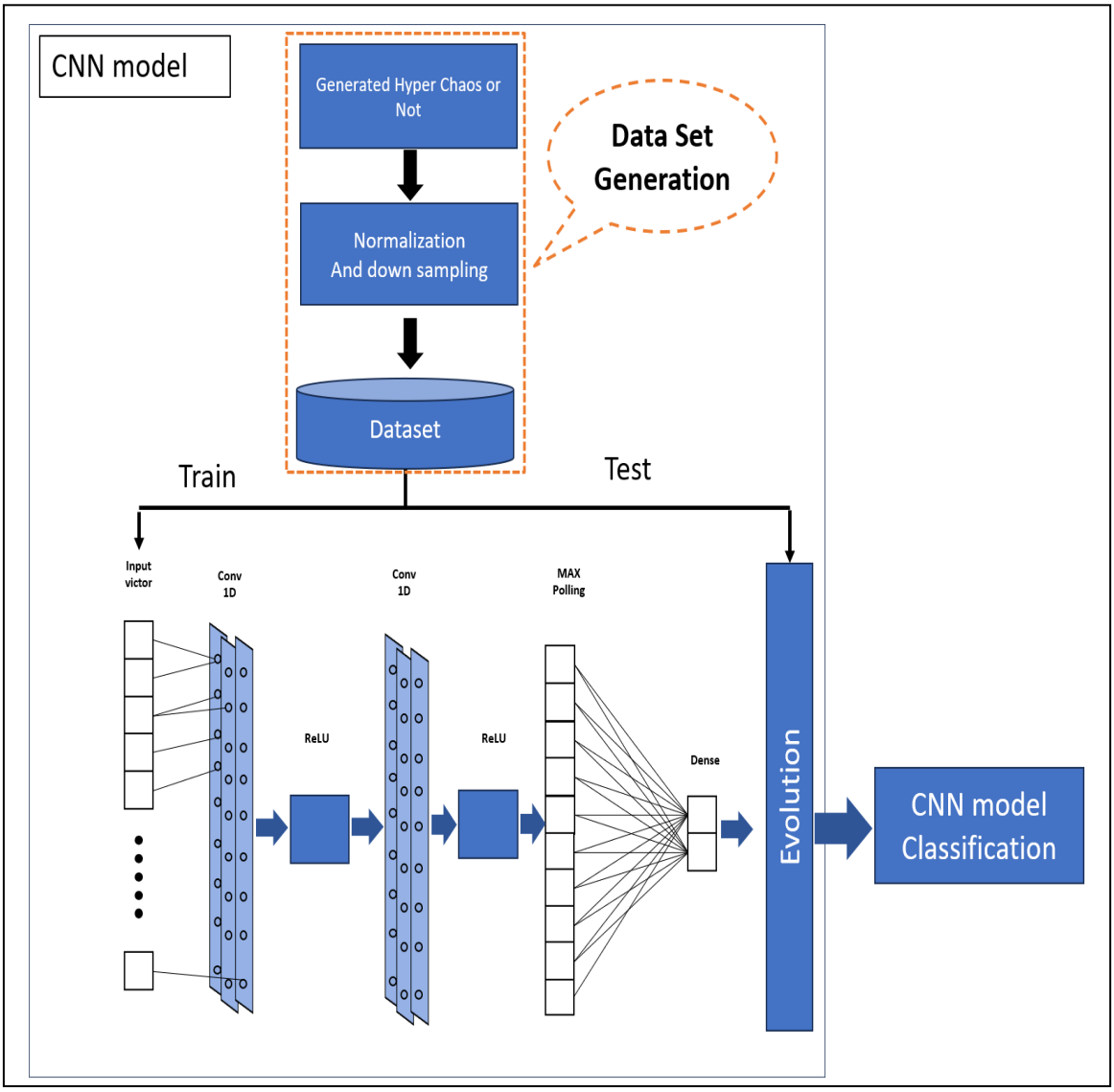


Figure 3.2: The Proposed Method For Testing Signal Based on CNN.

3.4.1 Dataset Generation

Two types of signals (Hyper-chaos and Not-chaos) are generated in algorithm (3.1). This algorithm generates a dataset of signals from the Rabinovitch system. It takes groups of parameters (a, b, c, d, and r) as input and produces two signal groups: 2000 signals of Hyper-chaotic signals and 2000 signals of Not-chaotic signals (fixed point or periodic).

First, the algorithm of dataset generation generates a random initial condition in the same range as the Hyper-chaotic system. And using these initial conditions to generate chaotic sequences size of 30,000 samples. Because the change in chaotic signal is smoothly, make a down-sampling by taking one sample from every 50 samples (the output signal length equals 30,000/50 = 600 samples). Normalize the output sequence between (-1 to 1). Finally, save the results in the dataset matrix.

Algorithm (3.1): Dataset Generation from the Rabinovitch System

Input: groups of parameters (a, b, c, d, r)

Output: Dataset

Begin

1: Set i=1
2: While i<=number groups of parameters, do
3: Set k=1
4: While k <= number groups of initial conditions, do
5: Generate a random initial condition in the same range of the Hyper-chaotic system.
6: Generate Sequence As in Algorithm (2.1).
7: Down-sampling the generated sequence.
8: Normalized the output sequence.
9: Save the results in the Dataset matrix.
10: end while
11: end while

End

3.4.2 Feature Extraction

The feature extraction phase of the proposed system is based on a Convolutional Neural Network (CNN). The CNN utilizes self-learnable filter values to extract features from signals. These extracted features serve as the initial input for the subsequent classification step.

Two sequential Convolutional (Conv) blocks make up the feature extraction step of the proposed convolutional neural network approach used in this thesies.

The convolutional neural network (CNN) has two convolution layers, each with a set of feature maps. The first layer applies convolutions to the input data, using an activation function called Rectified Linear Unit (ReLU), and normalization are applied after each convolution layer to enhancing the network's performance and stability by normalizing the data. The last layer of second conv block is flowed by

max-pooling layer to reduces the feature map's dimensions to a quarter of its original size while preserving essential details. The output is a one-dimensional matrix, which serves as input for a fully connected layer. Algorithm(2.3) of the convolutional neural network construction process includes steps 1 to 5, which cover the feature extraction phase, and the remaining steps correspond to the classification phase. The proposed network architecture, comprises 11 consecutive layers, segregating into two stages based on their respective functions: the feature extraction and classification phases.

Algorithm (3.2) demonstrates the chaotic signals feature extraction and classification process using the proposed convolution Neural Network structure.

Algorithm (3.2): Convolution Neural Network

Input: (X train, y train) is the training dataset, whereas (X test, y test) is the test dataset.

Output: Test loss and accuracy

Begin

1: Initialization: Initialize a CNN model.

2: Convolutional Layers:

3: - Add the first convolutional layer with 32 filters and ReLU activation.

4: - Add the second convolutional layer with 64 filters and ReLU activation.

5: Average Max Pooling: Include a layer of average maximum pooling with a pool size of two.

6: Fully Connected Layer: Include a thick layer that is fully connected, with 64 units and ReLU activation.

7: Output Layer: Include the number of class units and softmax activation in the output layer.

8: Model Construction: Build the model with the Adam optimizer and the categorical cross-entropy loss function.

9: Model Training:

10: - Train the model using the training dataset (X_train, y_train).

11: - Use a batch size of 20 and train for 10 epochs.

12: - Perform validation on the validation dataset (X_val, y_val) during training.

13: Model Evaluation: Evaluate the trained model on the test dataset (X_test, y_test).

14: Performance Metrics Calculation: Calculate the test loss and test accuracy.

15: Output: Return the test loss and accuracy as the results.

End

3.4.3 Classification Phase

Signal features are employed to determine the signal type to categorize it as either Hyper-chaos or Not-chaos precisely. Within the eleventh layer of the Convolutional Neural Network (CNN), the output is represented by two classes: Hyper-chaos and Not-chaos. The SoftMax classifier performs this classification task.

- Softmax Classifier

After the max pooling layer of the Convolutional Neural Network building algorithm (3.2), there is a Fully Connected layer of the Dense architecture with an activation function (ReLU). The output is sent to the Softmax activation function, which determines the probability for each category depending on the input signals. The probabilities are then passed to the loss function Equation (2.8) to determine the error value, which is subsequently used to alter the weights throughout the backpropagation training phase of the proposed architecture.

CNN's training phase employs the Adam optimization method, loss function, learning rate reduction, and early stopping to determine the optimal number of epochs for signal categorization. The training procedure begins with feature extraction, with the signal being fed in both forward and backward directions through the convolutional neural network structure over the course of several epochs. The outcome of the training phase is a trained set of weights and filters for all network architectural layers, which are saved for use in the testing phase.

In the convolutional neural network CNN test phase, the test is conducted on invisible test data, beginning with the signal passing on the convolutional neural network structure in the forward direction to extract features and classify them as Hyper-chaos or Not-chaos utilizing the trained weights and filters stored during the training phase.

3.5 Generation Chaotic Random Position (CRP) and Chaotic Random Integer Numbers (CRIN)

Before entering for the two proposed encryption methods, Chaotic Random Position (CRP) generation and the Chaotic Random Integer Numbers (CRIN) generation and algorithms are explained in the following sub-section.

3.5.1 Generation Chaotic Random Position (CRP) Based on Hyper-chaotic

Using chaotic signals with a random behavior can design a scrambler to achieve security by hashing the original information signal. This operation is called the chaotic scrambling method. This design assumes that the initial conditions and parameters of the Hyper-chaotic are the same in both the transmitter and receiver.

The algorithm takes a digital color image and several parameters and initial conditions as input. It generates a Scrambler Look-Up Table (LUT) based on a Hyper-chaotic Sequence (HCS) of the same length as a specified block 'n' in the image. Generate a Hyper-chaotic Sequence (HCS) using "Algorithm (2.1)" with provided parameters and initial conditions. To neglected the continuity between chaotic samples and restrict the HCS values between 0 to 254 by multiplying with a big number such as (10^8) taking the modulo 255, and sorting the scaled HCS

values in ascending order. Compare the i^{th} element from the sorted vector with the original sequence, the index of the matched element from the original sequence as the i^{th} entry in the Scrambler LUT.

Algorithm (3.3): Generation Random Chaotic Position (CRP) Based on Hyper-chaotic

Input: Digital Color Image, chaos signal initial condition X0, Y0, Z، W0, chaos signal parameters a, b, c, d, r.

n: length of Block.

Output: Scrambler LUT

Begin

1: Generate a Hyper-chaotic Sequence HCS in the same length of n as in Algorithm(2.1).

2: $RandVal = [(HCS * 10^8)\ mod\ 255]$

3: Sorting $RandVal$ ascendingly.

4: Set i=1

5: While i<= n do

6: Compare the i^{th} element from the sorted vector with the original sequence and set the index of the matched element as i^{th} entry to the scrambler LUT.

7: Increment by 1

8: end while

End

Figure 3.3 show the proposed generation chaotic random position (CRP) based on Hyper-chaotic signal.

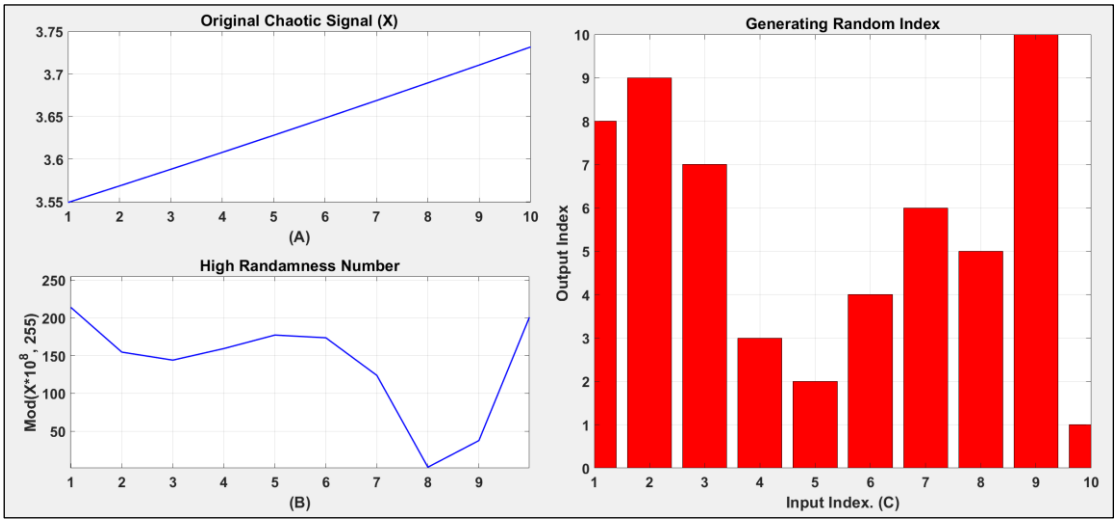


Figure 3.3: (A) Generating Chaotic signal (Output of Algorithm 3.1). (B) RandVal (Algorithm 3.4, Step 2). (C) Scrambler LUT (Output of Algorithm 3.3).

3.5.2 Generation CRIN Method Based on Hyper-chaotic Signals

The algorithm leverages a hyper-chaotic sequence to generate a series of random integers that exhibit chaotic behavior. As shown in the following algorithm, the CRIN sequence can be used for various purposes, such as randomization, encryption, or other applications requiring chaotic randomness.

Algorithm (3.4): Chaotic Random Integer Number (CRIN)

Input: Digital Color Image, chaos signal initial conditions X0, Y0, Z، W0, chaos signal parameters a, b, c, d, r.

n: length of Block.

Output: Chaotic Random Integer Numbers "CRIN"

Begin

1: Generate Hyper-Chaotic Signal of length n as in Algorithm (2.1).

2: $RandVal = \left[(HCS * 10^8)\ mod\ 255\right] + 1$

3: CRIN = $round(RandVal)$

End

3.6 Image Encryption Methods

In the proposed system, several ways exist to encrypt images (change the pixel position, change the pixel value, and change the position and value of the pixel simultaneously). In general, proposing two methods, the first algorithm for digital image encryption is named Image Encryption Based on Hyper-chaotic Scrambling Bits (IEHSB) method, and the second method is named Image Encryption Based on Three Hyper-chaotic Signals (IE3HS) method.

3.7 The Image Encryption Based on Hyper-chaotic Scrambling Bits (IEHSB) Method

In the first proposed system, as shown in the algorithm (3.5), the original digital image is converted to Blocks, as shown in Figure 3.4. The size of blocks has four cases Pixel, Row, Column, and one vector image cases. The fifth case combines the Row and Column and is named the Row-Column case, all cases shown in Figure 3-5.

The Blocks are converted to binary bits and generate a chaotic signal based on an algorithm (2.1) that is the same length as the binary Block. Then, after generation, it tested if this sequence shows Hyper-chaotic or Not-chaotic behavior based on CNN-Tester as in the algorithm (3.2). If the signals are Not-chaotic, changing the parameters and initial conditions is completed until a Hyper-chaotic signal is obtained. Then, using the diffusion (scrambling bits) process using algorithm (3.3) “CRP method” if it has Hyper-chaotic behavior. And then hashing the bits based on a new bit's position generated. Finally, reconstruct the encrypted Blocks and then reshape them to the encrypted image.

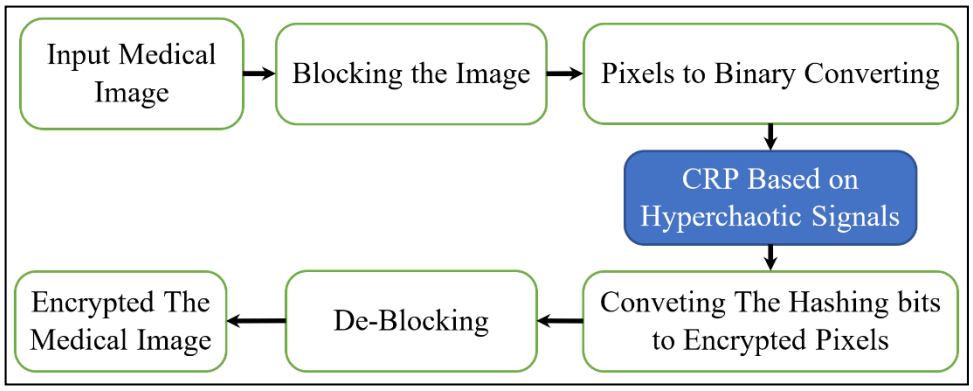


Figure 3.4: The Proposed Image Encryption Based on IEHSB Method.

In Figure (3.5) All cases are available of digital image scrambling.

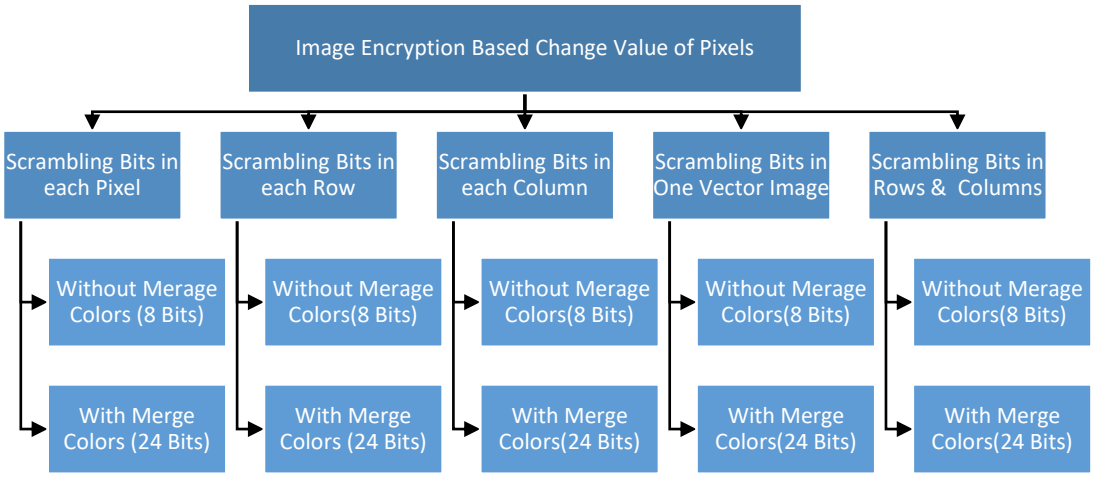


Figure 3-5: Available Cases (Bloking) for Changing Values of Pixels Based on Chaotic Scrambling.

Algorithm (3.5): Image encryption Based on Hyper Chaotic Scrambling Bits (IEHBS) method
Input: Digital Color Image (DI), chaos signal initial conditions X0, Y0, Z0, W0, chaos signal parameters a, b, c, d, r Output: Encrypted Image (EI) **Begin** 1: Read Digital Color Image (DI) 2: Blocking the image into the fixed length blocks. 3: Set d = 1, 4: h = number of blocks 5: while d <= h, do 6: Convert each Block into binary format 7 Generate Signal as the length of the binary Block as in Algorithm (2.1). 8: Test the generated signal from Step 7 using an algorithm (3.2) to determine if it exhibits Hyper-

chaotic or not.

9: Use algorithm (3.3) to sort the integer numbers generated from Step 7.

10: Rearrange the position of bits in each Block according to the sorting result from Step 9.

11: Convert the sequence generated from Step 10 from binary to integer format.

12: EI= Combine the modified blocks to create a 3-D matrix representing the encrypted image.

13: Increment d by 1

14: end while

End

Note: The case of Row-Column Scrambling is done by repeating the previous algorithm two times, first: for hashing bits in rows and the second: for hashing bits in columns.

3.8 Image Encryption Based on Three Hyper-chaotic Signals (IE3HS) method

The proposed system uses three Hyper-chaotic Rabinovitch systems with different initial condition and parameters, as shown in Figure 3.6.

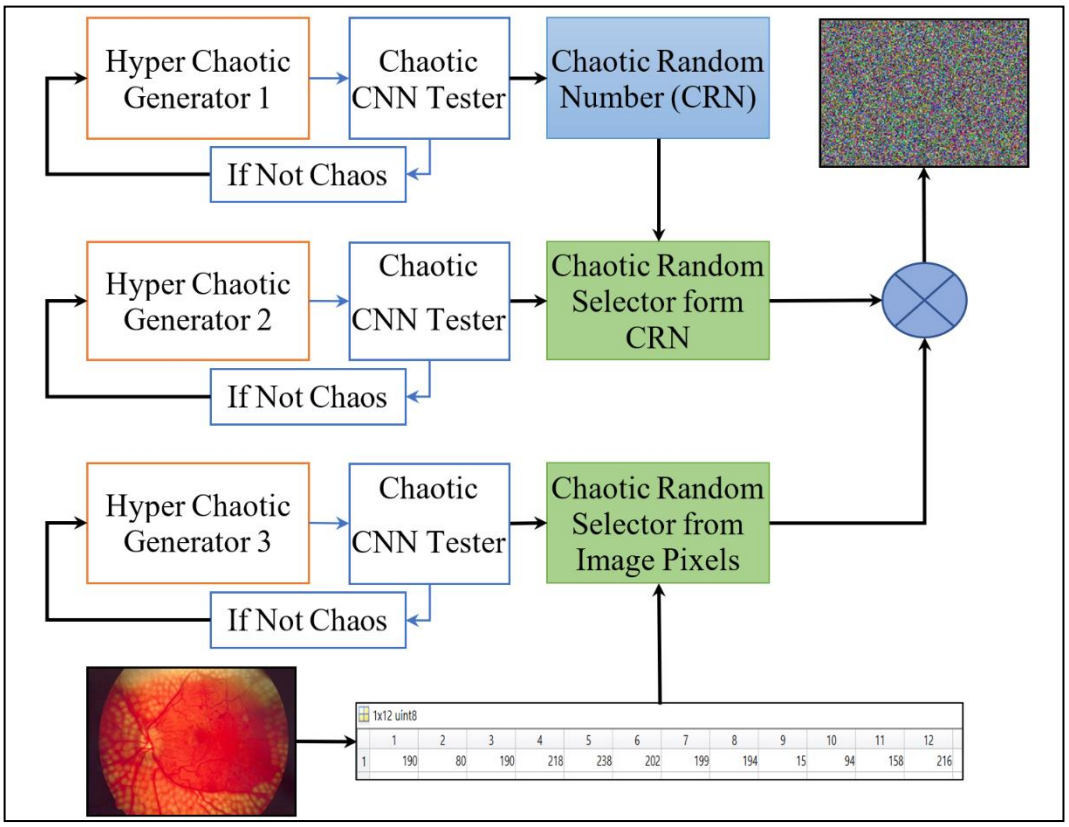


Figure 3.6: General Scheme of The IE3HS Method.

CNN-Tester tests the three chaotic signal generators before being used in image encryption. If the CNN-Tester does not pass the generated signal, in this case, it must change the chaotic parameters and initial conditions and test again until passing by CNN-Tester. The exact initial conditions and parameters are used on the decryption side to recover the original image.

The original image is converted to one vector of pixels, and the first Hyper-chaotic Rabinovitch is used to generate chaotic random integer numbers (CRN) between (1 to 255) and the same length one vector image.

The second Hyper-chaotic Rabinovitch system used a random selector from the CRN vector. At the same time, the third Hyper-chaotic Rabinovitch system used a random selector from one vector image.

Finally, the Encrypted pixel is done by XORed bitwise between the chaotic selected CRN value and the chaotic selected pixel from the one vector image. Repeated this process until encrypted all pixels and reshape one vector of encrypted pixels to an encrypted image the same size as the original image. The proposed EI3HS algorithm is described in the following algorithm (3.6):

Algorithm (3.6): Image Encryption Based on Three Hyper-chaotic Signals (IE3HS) Method

Input: Digital color Image, chaos signal initial condition1 X1, Y1, Z1, W1, chaos signal parameters1 a1, b1, c1, d1, r1, chaos signal initial condition2 X2, Y2, Z2, W2, chaos signal parameters2 a2, b2c, d2, r2, chaos signal initial condition3 X3, Y3, Z3, W3, chaos signal parameters3 a3, b3, c3, d3, r3

Output: Encrypted Image (EI)

Begin

1: Read Digital Color Image

2: MI=Convert original image to one vector

3: Generate Sequence as in Algorithm (2.1), size of sequences =MI.

4: Call algorithm (3.2) to test the generated signal from Step 3 is Hyper-chaotic or Not-chaotic signal.

5: Call Algorithm (3.4) to generate CRIN.

6: Generate Sequence as in Algorithm (2.1),, size of sequences= sequence generated from Step 3

7: Call algorithm (3.2) to test the generated signal from Step 6 is Hyper-chaotic or Not-chaotic signals

8: Call algorithm (3.3) to generate a CRP used as a first random selector from step 5.

9: Rearrange the sequence position generated from Step 5 according to the new index generated in Step 8.

10: Generate Sequence as in Algorithm (2.1), size of sequences= MI

11: Call algorithm (3.2) to test the generated signal from Step 10 is Hyper-chaotic or Not-chaotic signals

12: Call algorithm (3.3) to generate a CRP used as a second random selector from MI in Step 2.

13: Rearrange the position of MI according to the new index generated in Step 12

14: Set n=1

15: While n <= length (MI), do

16: E(n) = XOR between the sequence generated from Step 9 and the sequence generated from Step 13

17: Increment n by 1

18: end while

19: EI = reshape E to the original size of the Digital Image.

End

Chapter Four

The Results of CNN Testing and Image Encryption

4.1 Introduction

This chapter presents the results of two encryption methods and evaluated the performance using several metrics such as PSNR, MSE, Entropy, SSIM, and Correlation. It also gives randomness testing of the Rabinovitch system signals by the classical method based on Lyapunov exponents and shows the result of the proposed CNN Hyper-chaos testing method.

4.2 Environment Distribution

The computer laptop with an Intel Core i7 4500U processor, 1.80 GHz of CPU speed, and 8 GB of RAM was used to develop the simulation findings in this study. The software used was MATLAB (R2022a) and Windows 10 Pro as the operating system.

4.3 Testing Randomness for the Rabinovitch System

4.3.1 Testing Parameters of HCRS based on Lyapunov Exponents

The Lyapunov Exponent test allows the system to be defined in three states, Hyper-chaos, Chaos, and Not-chaos, through lambda values from the test results. The system is Hyper-chaos if it meets the following conditions:

1. The system must contain four state vectors or above, which means four values of λ. Rabinovich's system has four dimensions, 'state vectors' (x, y, z, and w).
2. At least two λ values must be positive.
3. The addition of all Lyapunov exponents ($\lambda_1 to\ \lambda_4$) values must be negative summation.

Table 4.1: Testing Parameters of Chaotic Rabinovich System use Lyapunov Exponent.

Set of Parameters					Lyapunov Exponents				Result	Testing Time
r	a	b	c	d	λ_1	λ_2	λ_3	λ_4		(Second)
-4.551	**5.272**	**-1.767**	**6.389**	**-5.245**	2.835	-0.105	-3.774	-14.095	Chaos	1.161
11.596	**6.645**	**1.435**	**5.063**	**-3.533**	1.223	-1.02	-7.330	-9.550	Chaos	1.048
5.440	**8.437**	**-3.620**	**8.179**	**-2.841**	1.289	-0.618	-6.799	-9.709	Chaos	1.086
3.480	**4.180**	**-1.220**	**1.1**	**-5.8**	0.402	0.025	-4.852	-5.434	Hyperchaos	1.027
10	**5**	**-2**	**1**	**-5**	0.676	0.294	-4.135	-5.835	Hyperchaos	1.077
11.50	**5.270**	**-1.9**	**0.56**	**-6.320**	0.978	0.154	-4.928	-6.454	Hyperchaos	1.089
10.90	**4.320**	**-0.66**	**1**	**-8**	0.532	0.269	-5.649	-7.812	Hyperchaos	1.1050
8.1	**4**	**1.3**	**1**	**-2.2**	0.0725	-0.2199	-3.890	-4.4625	Chaos	1.1021
-6.66	**8.74**	**3.89**	**1.582**	**-8.571**	-0.636	-0.681	-8.44	-13.02	Not-Chaos	1.2012
3.4	**3.47**	**13.92**	**9.221**	**-8.152**	-2.46	-8.159	-9.22	-14.92	Not-Chaos	1.9311
5.77	**13.44**	**6.47**	**4.66**	**-2.22**	-2.232	-3.22	-4.669	-16.671	Not-Chaos	1.7124

Table 4.1 shows that the test was carried out using Lyapunov Exponents for parameters of the Rabinovitch system, and the test results were four values from lambda. The average time to check for parameters (r, a, b, c, and d) is about 1.1 seconds.

4.3.2 Randomness Testing for the Random Bits Generator Based on HCRS

The hyper-chaotic random behavior proposed in the study is assessed using the NIST test suite, which encompasses more than ten statistical tests (seven were

employed). These tests are employed to gauge the randomness of the generated sequence. The assessment hinges on the calculated probability value (p-value). A comparison is drawn between this p-value and a significance level denoted as $\alpha = 0.01$, which acts as a threshold between acceptance and rejection regions. If the computed p-value is below 0.01, the sequence is deemed non-random and is dismissed. Conversely, if the p-value is greater than or equal to 0.01, **Error! eference source not found.** presents the resulting outcomes.

Table 4.2: NIST Testing for The Randomness Results from HCRS.

Test	P-Value	Results
Frequency (Monobit)	0.18999	passed
Frequency Test within a Block	1.0056	passed
Longest Run of Ones in a Block	0.53869	passed
Random Execution Variant	0.84871	passed
Longest Runs Test	1	passed
Frequency Block	0.16296	passed
Auto Correlation	1	passed

4.4 Dataset Signals Generations

The suggested system created the dataset, which consists of around 4000 Signals. (2000 Hyper-chaos) and (2000 Not-chaos) signals make up the dataset. The training data sample comprises 70 percent of the overall dataset (2800 Signals), whereas the testing data sample comprises 30 percent (1200 Signals). Table 4.3 shows the Statistics of the Signals divide of the dataset.

Table 4.3: Statistics of Signals Divide of Dataset.

	Hyper-chaos	Not-chaos	Total
Train	1419	1381	2800
Test	581	619	1200
Total	2000	2000	4000

Figure 4.1 shows six samples of non-chaotic signals, while Figure 4.2 shows six samples of Hyper-chaotic signals.

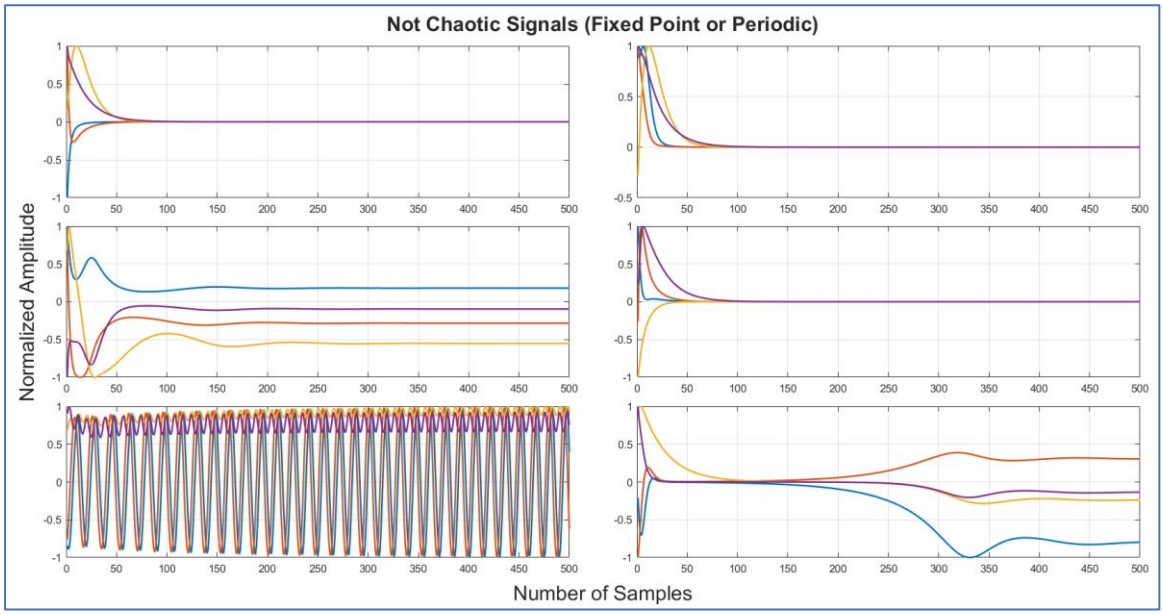


Figure 4.1: Samples for Non-chaotic Signals.

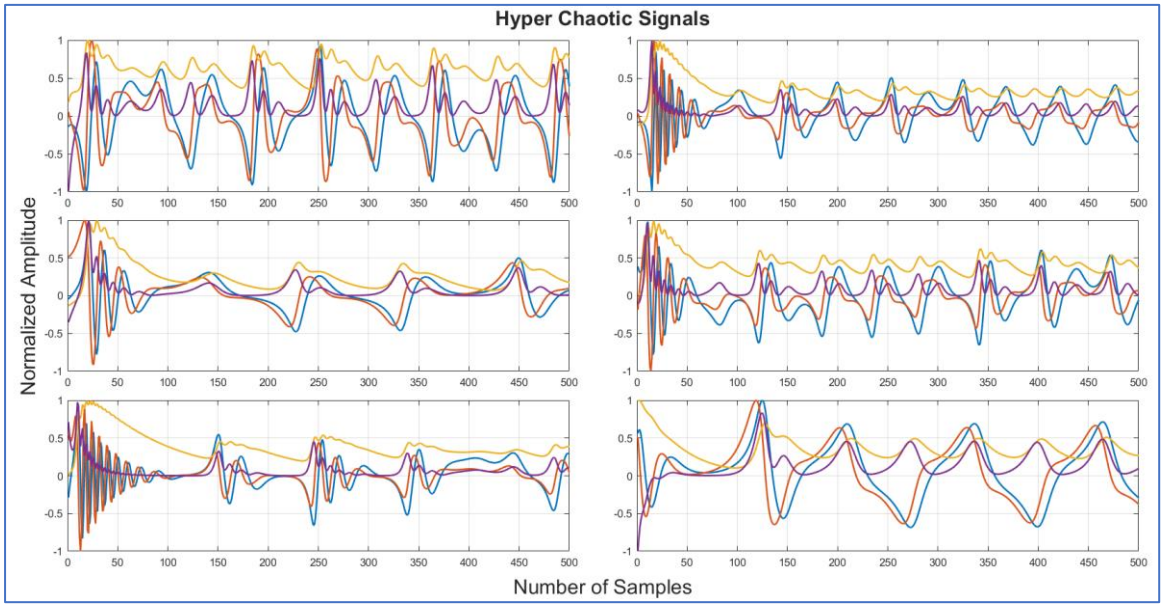


Figure 4.2: Samples for Hyper-chaotic Signals.

4.5 Result of Signal Classification

This section comprises two parts detailing CNN output: the results of the training phase and the test phase.

4.5.1 Training Phase Outputs

In the training phase, all training samples (2800) signals with their labels must pass through the system simultaneously to be trained. The following sections show the results and output signals of feature extraction and classification steps.

4.6 Results of the Features Extraction

Feature extraction procedures are implemented using a convolutional neural network (CNN) architecture comprising two consecutive blocks. Each block has three layers: convolution, Rectified Linear Unit (ReLU), and normalization. The Max-Pooling layer follows it, as shown in Table 4.4.

Table 4.4: CNN Layers Specific Details

Layer (type)	Filter	Parameter
Conv1D	32	2080
ReLU	32	0
Normalization	32	64
Conv1D	64	32832
ReLU	64	0
Normalization	64	128
MAX_Pooling1D	64	0
Total parameter		35104

Table 4.4 provides details on each layer of the CNN-1D that was implemented and displays the total number of parameters, which came to 35104. The information provided is organized in a tabular format.

4.6.1 Results of the Classification

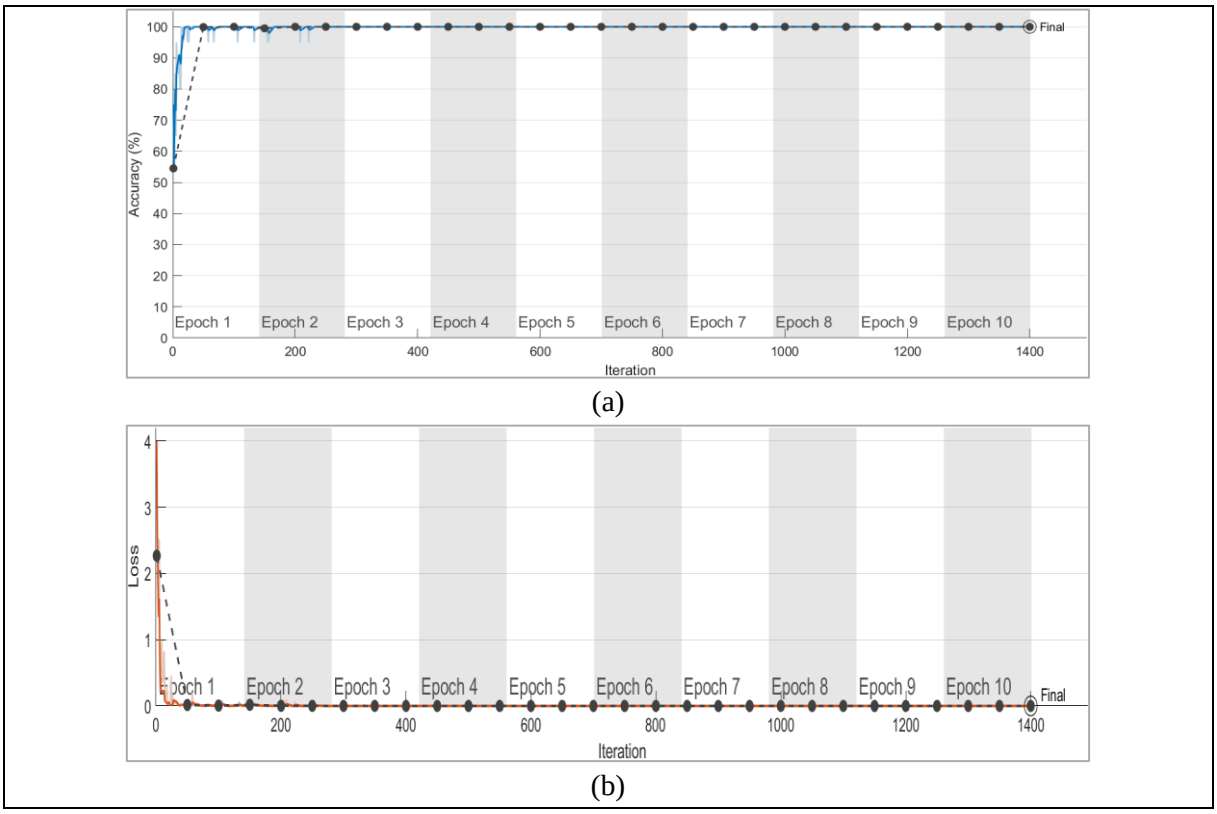
The classifier CNN-Softmax is used in this study to detect signals, and it consists of the remaining two layers of convolution neural network architecture CNN.

4.6.1.1 Result of Training Signals Classification

The classification technique for Signals, and it comprises of the last two layers of CNN convolution neural network design.

The fully Connected activation function layer of the Dense architecture (ReLU), and the output layer comprise the required number units (2). The output of this layer is sent to the Softmax activation function, which estimates the probability for each class given the input Signals. The possibilities are provided to the loss function (sparse categorical cross entropy) to determine the error value, which is then used to update (alter) the weights throughout the back propagation procedure. The Softmax activation function is used to classify input Signals into two categories (Hyper-chaos and Not-chaos).

Signals classification employs Adam optimization, learning rate reduction to enhance the value of the validation loss when model performance stops improving (decrease on the plateau), batch size=20, and early stopping to determine the optimal number of epochs (maximum= 10) during the training phase. The learning curves of the accuracy and loss for (the training sample and the validation sample) as shown in Figure 4.3.



(a)

(b)

Figure 4.3: The Accuracy and The Loss Learning Curves

4.6.1.2 Testing Phase Results

All 1200 test samples—30% of the database—will be sent to the system without labels during testing. The test Signals are divided into (Hyper-chaos and Not-chaos) based on the learned weights in the fully connected layers and the trained Filter in the convolution layers, which were saved during the training phase.

4.6.1.3 Result of Testing Signals

After the feature extraction stage, the extracted features are passed to the layers for the classification to label signals as Hyper-chaos and Not-chaos.

Figure 4.4 represents the confusion matrix using the “Soft Max function” classifier when the model is trained on the Data of the dataset, True Positive (TP) is the right classification of positive signals such as Hyper-chaos categorized as Hyper-chaos signals. False Positive” (FP) is a wrong classification of positive signals such as Hyper-chaos categorized as non-chaos. “True Negative” (TN) is the right classification of negative signals such as non-chaos signals categorized as Non-chaos. “False Negative” (FN) isa wrong classification of negative signals such as Non-chaos categorized as Hyper-chaos. Table 4.5 illustrate the performance measures results when the system is trained on the signal dataset in termsof Sensitivity, Precision, Specificity, and Accuracy.

Table 4.5: Performance Measures results of Signals Datasets.

Sensitivity	Precision	Specificity	Accuracy
1	1	1	100 %
Learning rate	Iteration per Epoch	Maximum iteration	Frequency
0.001	360	3600	50 iterations

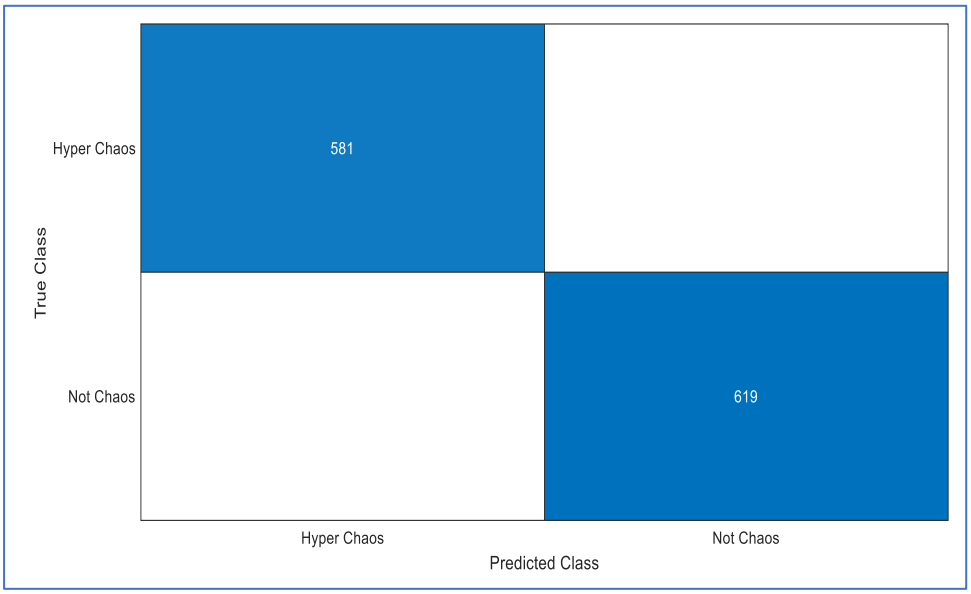
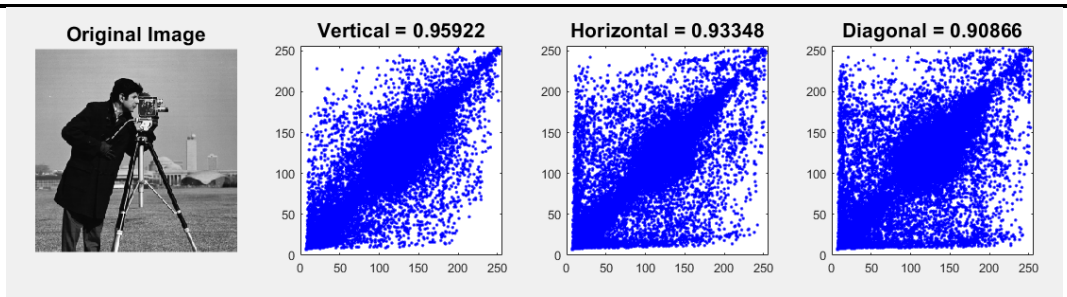
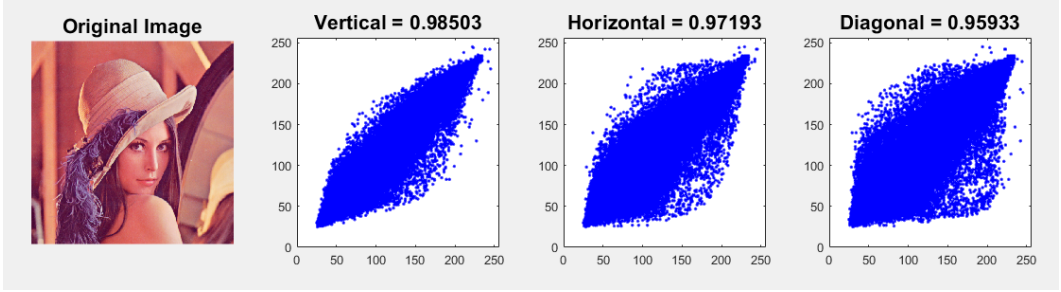


Figure 4.4: Performance Measures Results of Signals Datasets.

4.7 The Images That Using for Encryption

This section shows seven images in various sizes, contrasts, and colors. The suggested encryption algorithms used on these images and their performance rating based on their outputs. Table 4.6 shows the images that used for evaluation the encryption methods.

Table 4.6: The Images That Use for Encryption

 **Image1: 259x194x3**
 **Image2: 621x975x3**

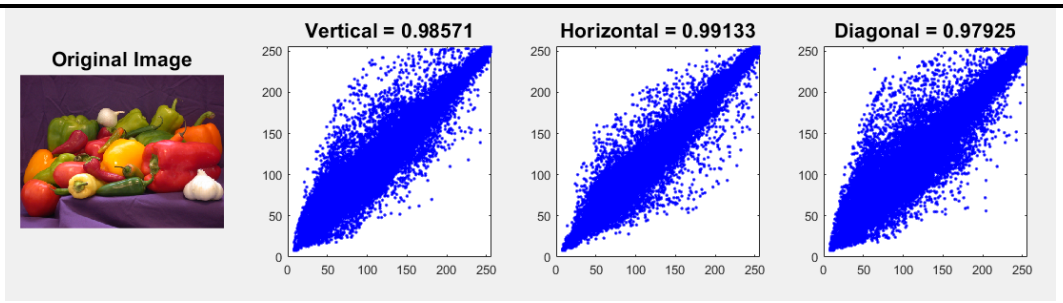


Image3: 350x350

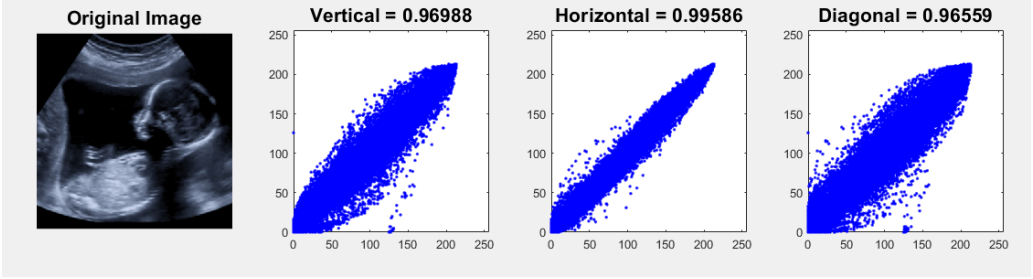


Image4: 320x320x3

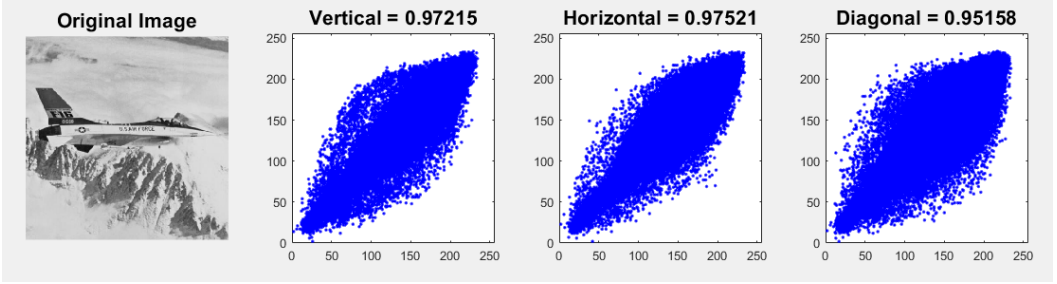


Image5: 320x320x3

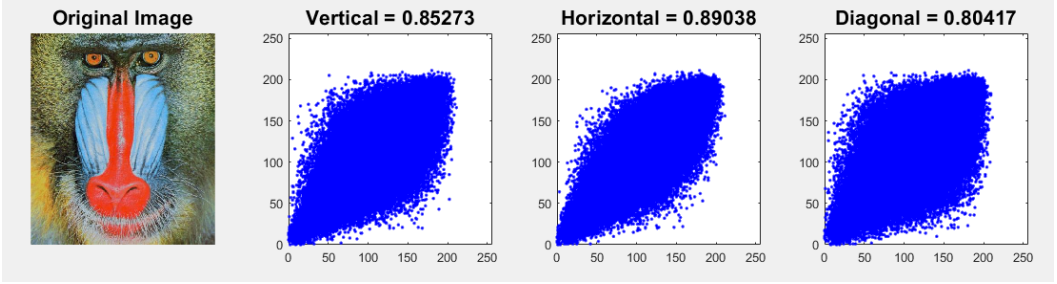


Image6: 532x710x3

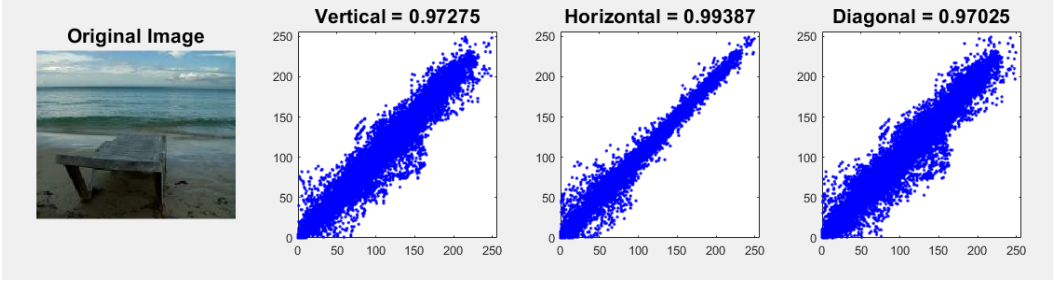


Image7: 183x275x3

Table 4.6 showed the seven original images and the Vertical Correlation Coefficient (V), Horizontal Correlation Coefficient (H), and Diagonal Correlation Coefficient (D). The proposed system method's efficiency is tested regarding security and robustness. Scientists have developed techniques for evaluating the strength of encryption schemes. The SSIM, Entropy, MSE, Correlation, and PSNR are some methods and algorithms used to assess the strength of encryption algorithms.

4.8 Results of The Image Encryption Based on Hyper-chaotic Scrambling Bits (IEHSB) Method

The results varied in the first encryption method due to the different cases of the scramble, which depended on five cases (Pixel, Row, Column, Row-Column, and One-Vector Image), and each type has two states in the color image. The first state: an (merge) between the colors of the image R, G, and B colors. As for the second case: the colors do (without merge), so each color is encrypted separately.

4.8.1 IEHSB Method: Scrambling Bits in Each Pixel

The simulation outcomes for this technique are shown in Table 4.7 for the statistical results of encryption for seven images. Table 4.8 for Vertical (V), Horizontal (H), and Diagonal (D) correlation for the seven images, and Figure 4.6 shows the correlation plots for one digital encrypted image. Figure 4.5 shows the original, encryption, and decryption of two images and the histogram for each image.

Table 4.7: Simulation Results of Scrambling Bits in Each Pixel.

Proposed Method	Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Second)	
						Enc.	Dec.
Pixel Merge Colors	Image1	0.011	7.966	10970	7.728	49.671	50.652
	Image2	0.232	6.519	5633.8	10.623	10.360	10.193
	Image3	0.419	6.297	5216.1	10.358	8.5591	8.419
	Image4	0.022	7.819	7948.1	9.128	36.262	34.437
	Image5	0.012	7.611	6884.6	9.752	17.072	16.828
	Image6	0.016	7.913	6909.8	9.063	4.3162	4.209
	Image7	0.031	7.640	10986	7.722	41.530	42.559
Pixel without Merge Colors	Image1	0.281	7.918	7286.1	9.505	115.362	117.86
	Image2	0.242	6.737	4941.3	11.192	23.456	23.373
	Image3	0.269	6.350	4870.5	10.656	21.104	19.530
	Image4	0.030	7.793	7421.5	9.425	78.952	75.448
	Image5	0.044	7.674	6297.1	10.139	37.779	38.033
	Image6	0.034	7.913	6406.3	9.392	10.565	10.051
	Image7	0.481	6.964	5400	10.807	95.760	96.631

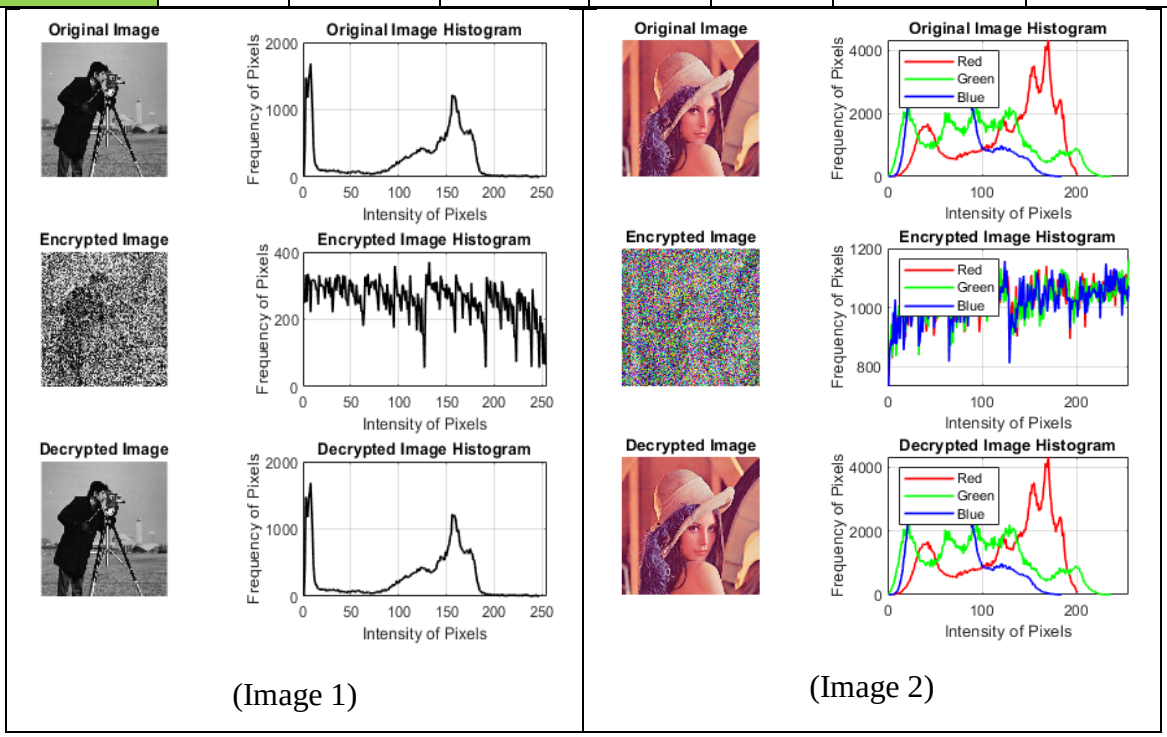


Figure 4.5: Enc. & Dec. Based on Scrambling Bits in Each Pixel

Table 4.8: V, H, & D Correlations for Column Scrambling.

Image	Method	V	H	D	Method	V	H	D
1	Row Merge Colors	0.6069	0.60893	0.58585	Row without Merge Colors	0.6112	0.61898	0.59548
2		0.6449	0.67834	0.64181		0.6422	0.67959	0.64247
3		0.2246	0.23873	0.21814		0.2151	0.25424	0.22297
4		0.5672	0.55224	0.54463		0.5279	0.52527	0.52073
5		0.2207	0.22342	0.21662		0.2255	0.23918	0.21598
6		0.1902	0.19105	0.17451		0.2011	0.22153	0.19304
7		0.6069	0.60893	0.58585		0.6112	0.61898	0.59548

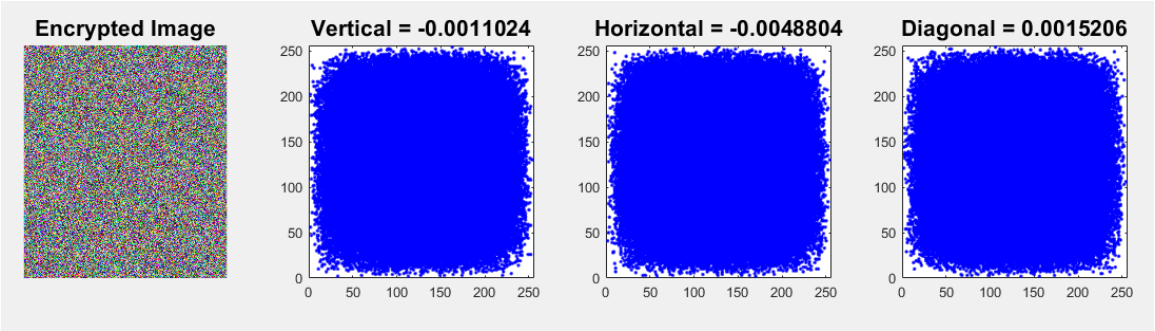


Figure 4.6: V, H, and D Correlation Coefficients for Enc. Image6 Based on IEHSB Method (Scrambling Bits in Each Pixel).

The results in Table 4.7 show that the SSIM in the second image is equal to 0.23, the third image is 0.419, and the time required for encryption and decryption is considerable (approximately 50 seconds for the first image for image 1), with a correlation of about 0.6 for merging all colors and without merging colors in each pixel as shown in Table 4.8 and Figure 4.6.

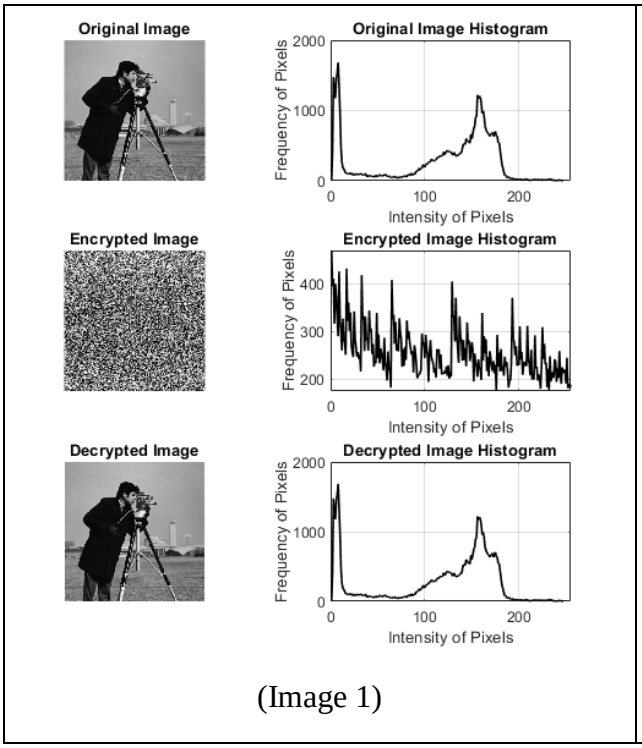
The results proved that the recovered image is identical to the original, as shown in Figure 4.5. The entropy of the "decrypted image" is the same entropy of the original image, SSIM equal to 1, MSE equal to zero, and PSNR equal to infinite. And the histogram of the decrypted images is identical to the histogram of the original images.

4.8.2 IEHSB Method: Results of Scrambling Bits in Each Row

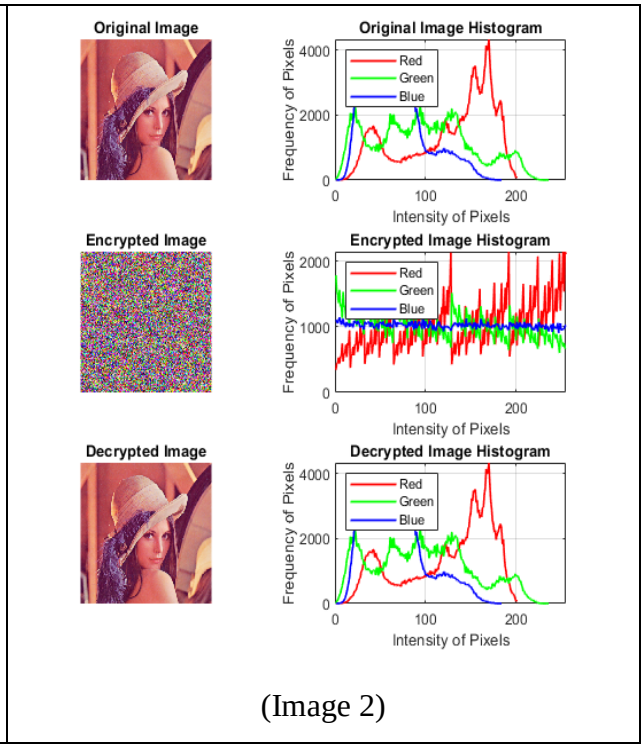
The statistical encryption results for the seven images using this approach are included in Table 4.9. While, Table 4.10 for Vertical (V), Horizontal (H), and Diagonal (D) appearance numeric correlation for the seven images, and Figure 4.8 shows the correlation plots for one digital encrypted image. Figure 4.7 shows the original, encryption, and decryption of two images and the histogram for each image.

Table 4.9: Simulation Results of Scrambling Bits in Each Row.

Proposed Method	Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Sec)	
						Enc.	Dec.
Row Merge Colors	Image1	0.006	7.966	12052	7.320	5.829	5.653
	Image2	0.024	7.249	8465.7	8.854	1.325	1.203
	Image3	0.034	6.971	7141.8	8.993	1.108	1.096
	Image4	0.007	7.818	9226.7	8.480	3.575	3.517
	Image5	0.008	7.771	10198	8.045	2.260	1.826
	Image6	0.010	7.924	8022.5	8.415	0.548	0.512
	Image7	0.004	7.768	14583	6.492	4.523	4.521
Row without Merge Colors	Image1	0.228	7.939	9427.9	8.386	5.984	5.476
	Image2	0.025	7.250	8483.8	8.844	1.330	1.250
	Image3	0.043	6.969	7107	9.014	1.037	1.027
	Image4	0.012	7.816	9226.3	8.480	3.809	3.515
	Image5	0.021	7.767	10134	8.073	1.912	1.790
	Image6	0.027	7.923	7892.8	8.485	0.548	0.584
	Image7	0.285	7.519	11058	7.694	4.565	4.480



(Image 1)



(Image 2)

Figure 4.7: Enc. & Dec. Based on Row Scrambling.

Table 4.10: V, H, & D Correlations for Row Scrambling.

Image	Method	V	H	D	Method	V	H	D
1	Row Merge Colors	0.16997	0.17133	0.04183	Row without Merge Colors	0.18723	0.18789	0.06226
2		0.21433	0.21883	0.21159		0.20323	0.21848	0.21323
3		0.27765	0.32468	0.27949		0.27572	0.32469	0.27126
4		0.07866	0.07809	0.07415		0.07007	0.06716	0.06605
5		0.11998	0.12036	0.11832		0.11042	0.11067	0.10564
6		0.06339	0.06303	0.0613		0.0597	0.06326	0.06415
7		0.02796	0.02598	0.03619		0.03768	0.03494	0.04211

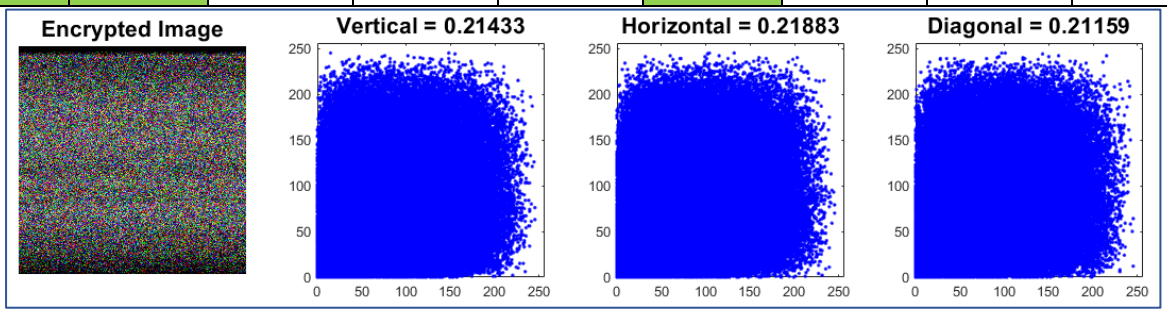


Figure 4.8: V, H, and D Correlation Coefficients for Enc. Image2 Based on IEHSB Method (Scrambling Bits Row merge all colors).

From the results in Table 4.10 and Figure 4.8, the V, H, and D correlation values are about 0.2 in all cases. Table 4.9 and Figure 4.7 show that the results are more acceptable than those in the previous method (Scrambling bits in each pixel). The biggest problem in this method is that when all row pixels have the same color, the row in the encrypted image has the same color.

As illustrated in Figure 4.7, the results proved that the recovered image is identical the original image. The recovered image has the same entropy as the original image, with SSIM = 1, MSE = 0, and PSNR = infinity. And the histograms of the encrypted images and the original images are identical.

4.8.3 IEHSB Method: Results of Scrambling Bits in Each Column

The simulation statically results for this method encryption of seven images are shown in Table 4.11 and Table 4.12 for Vertical (V), Horizontal (H), and Diagonal (D) appearance numeric correlation for the seven images, and Figure 4.10 shows the correlation plots for one digital encrypted image. And Figure 4.9 shows the original, encryption, and decryption of two images and the histogram for each image.

Table 4.11: Simulation Results of Scrambling Bits in Each Column.

Proposed Method	Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Sec)	
						Enc.	Dec.
Column Merge Colors	Image1	0.006	7.962	11471	7.534	6.065	6.220
	Image2	0.033	7.202	8530.8	8.820	1.261	1.148
	Image3	0.007	7.044	8068.6	8.463	1.040	0.956
	Image4	0.007	7.821	9622.8	8.297	3.565	3.485
	Image5	0.006	7.835	12475	7.170	1.848	1.788
	Image6	0.008	7.923	8028.8	8.411	0.548	0.490
	Image7	0.004	7.726	12892	7.027	4.450	4.450
Column	Image1	0.249	7.901	8470.1	8.851	5.583	5.558

Proposed Method	Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Sec)	
						Enc.	Dec.
Without Merge Colors	Image2	0.031	7.202	8537.5	8.817	1.376	1.229
	Image3	0.013	7.044	8016.5	8.491	1.064	1.044
	Image4	0.012	7.820	9611.6	8.302	3.655	3.562
	Image5	0.008	7.835	12483	7.167	1.999	1.918
	Image6	0.027	7.922	7857.2	8.505	0.577	0.556
	Image7	0.305	7.346	9205.3	8.490	4.583	4.409

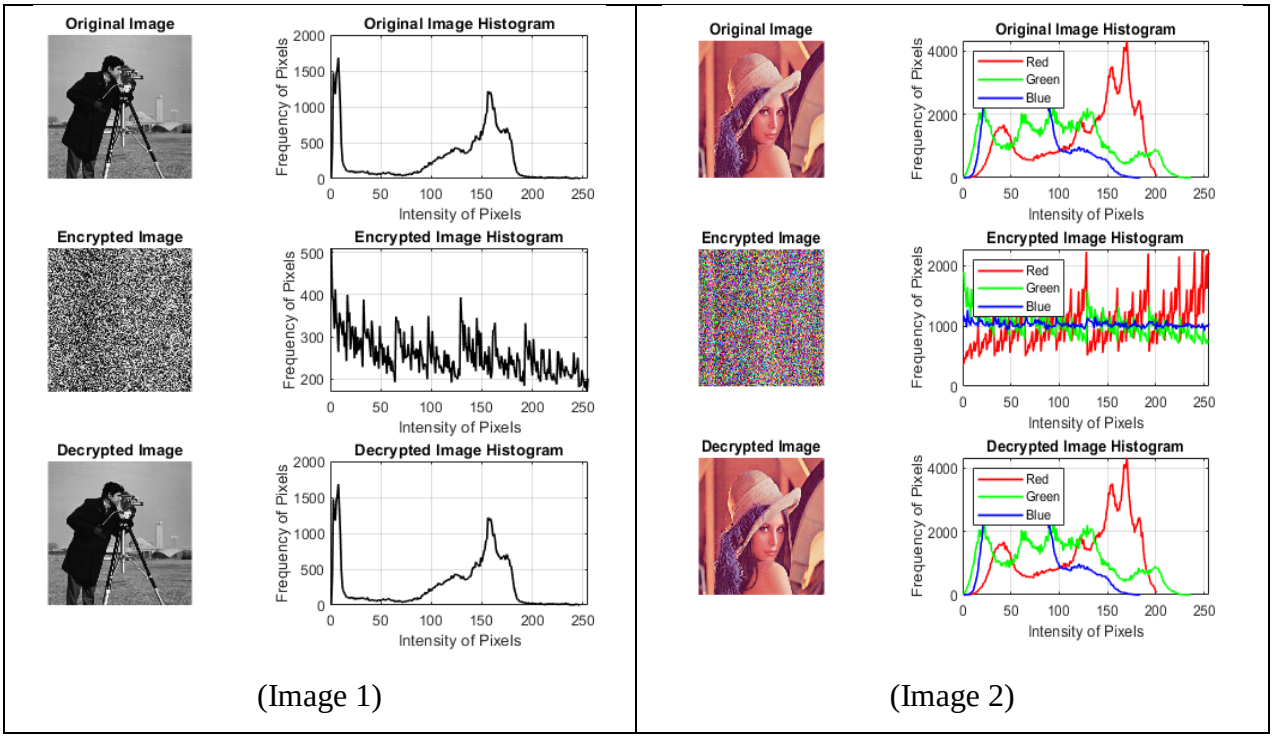


Figure 4.9: Enc. & Dec. Based on Column Scrambling.

Table 4.12: V, H, & D Correlations for Column Scrambling.

Image	Method	V	H	D	Method	V	H	D
1	Row Merge Colors	0.27135	0.27445	0.16077	Row without Merge Colors	0.28074	0.28042	0.17087
2		0.2907	0.28764	0.28278		0.29405	0.28567	0.28305
3		0.09684	0.09603	0.09637		0.08533	0.09308	0.08
4		0.03057	0.0343	0.03025		0.03247	0.03118	0.02954
5		0.34284	0.33863	0.33748		0.31636	0.3171	0.32416
6		0.06915	0.05991	0.07114		0.06524	0.06701	0.06332
7		0.03168	0.02611	0.03418		0.0377	0.03592	0.03065

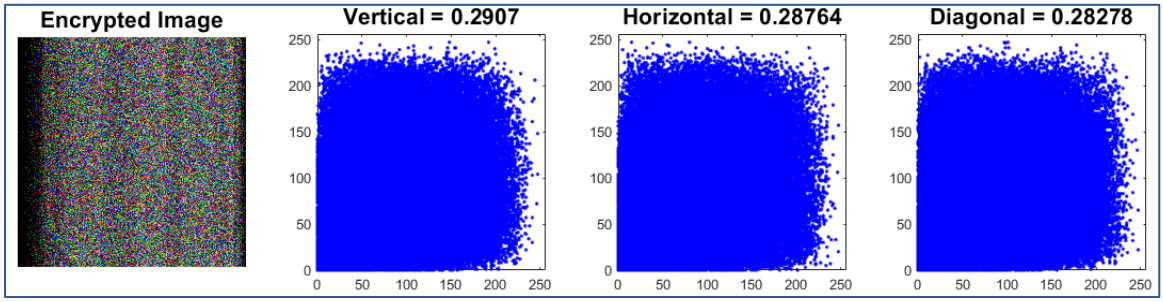


Figure 4.10: V, H, and D Correlation Coefficients for Enc. Image2 Based on IEHSB Method (Scrambling Bits Column Merge All Colors).

Based on the data shown in Figure 4.10, the V, H, and D correlation values are all about 0.2. The statistical encryption results for the seven images are shown in Table 4.11 based on the simulation results of the method described. Table 4.12 for Vertical (V), Horizontal (H), and Diagonal (D) appearance numeric correlation for seven images, and Figure 4.10 for the correlation plots for one encrypted digital image. Figure 4.9 shows the original, encrypted, and decrypted of two images, as well as the histogram for each image.

Figure 4.9 and Figure 4.10, the results are more acceptable than those in the (Scrambling bits in each pixel), and the results are very close to (Scrambling bits in Rows). The biggest problem in this method is that when all column pixels have the same color, the row in the encrypted image has the same color.

As seen in Figure 4.9, the results proved that the recovered image is identical to the original. The recovered image has the same entropy as the original image, with SSIM = 1, MSE = 0, and PSNR = infinity. The histograms of the encrypted images and the original images are identical.

4.8.4 IEHSB Method: Results of Scrambling Bits in One Vector

Table 4.13 displays the simulation results for this method's statistical encryption of the seven images. Table 4.14 shows the Vertical (V), Horizontal (H), and Diagonal (D) appearance numeric correlation for the seven images, and

Figure 4.12 shows the correlation plots for one digital encrypted image. Figure 4.11 shows the original, encryption, and decryption of two images and the histogram for each image.

Table 4.13: Simulation Results of Scrambling Bits in One-Vector Image.

Proposed Method	Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Sec)	
						Enc.	Dec.
One Vector Merge Colors	Image1	0.006	7.966	12105	7.301	6.852	6.589
	Image2	0.006	7.286	9075.6	8.552	1.374	1.323
	Image3	0.007	7.049	8323.5	8.328	1.066	1.057
	Image4	0.007	7.821	9893.4	8.177	4.263	4.278
	Image5	0.006	7.838	13090	6.961	2.229	2.111
	Image6	0.007	7.926	8445.4	8.192	0.519	0.693
	Image7	0.005	7.769	14581	6.492	5.717	5.994
One Vector without Merge Colors	Image1	0.221	7.944	9597.8	8.309	6.374	6.231
	Image2	0.006	7.286	9054	8.562	1.222	1.286
	Image3	0.013	7.048	8322.7	8.328	1.123	0.985
	Image4	0.011	7.821	9869.4	8.187	3.804	3.790
	Image5	0.008	7.838	13095	6.959	1.847	1.837
	Image6	0.028	7.926	8345.7	8.243	0.692	0.571
	Image7	0.266	7.559	11334	7.586	4.883	4.671

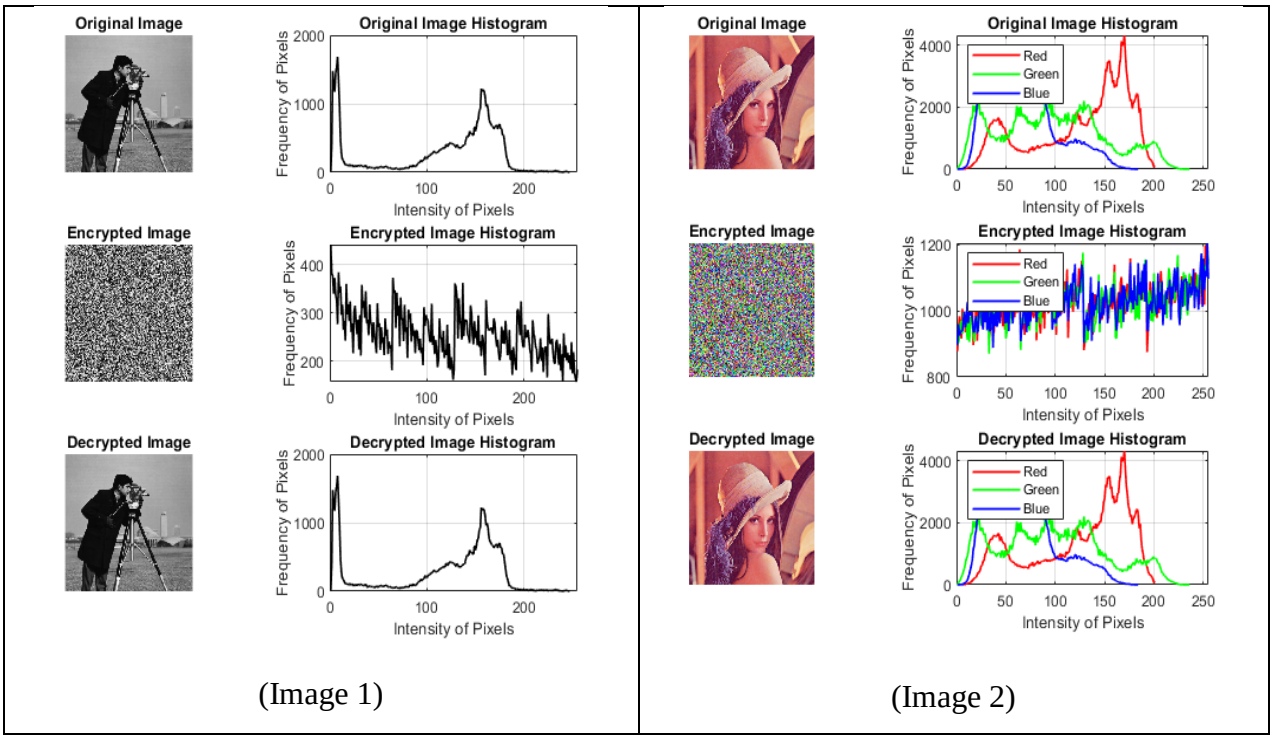


Figure 4.11: Enc. & Dec. Based on One-Vector Scrambling.

Table 4.14: V, H, & D Correlations for One-Vector Scrambling.

Image	Method	V	H	D	Method	V	H	D
1	Row Merge Colors	0.1570	0.1566	0.0248	Row without Merge Colors	0.1566	0.1563	0.0204
2		0.0025	0.0016	0.0023		0.0002	0.0032	0.0030
3		-0.0042	-4e-04	0.0043		2e-05	0.0013	-0.0012
4		0.001	-1e-04	-0.0022		0.0017	0.0025	0.00193
5		0.00309	0.0022	0.00488		-0.0006	-0.003	0.00715
6		-0.0038	-0.009	0.0005		0.00135	-0.004	0.00524
7		0.0045	-0.001	-0.0072		-0.0061	0.0027	-0.0029

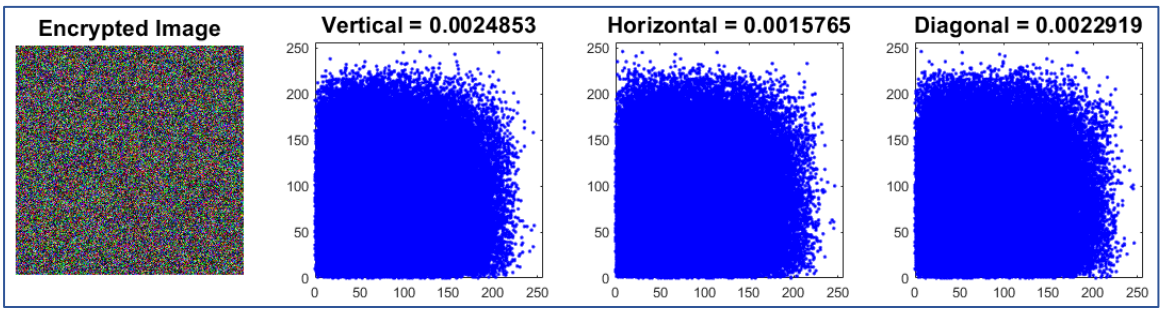


Figure 4.12: V, H, and D Correlation Coefficients for Enc. Image2 Based on IEHSB Method (Scrambling Bits One-Vector Merege All Colors).

The encryption results in Table 4.13 and Figure 4.11 are the best of all the scrambling techniques in all testing measures (the results are very similar to the Row-Column Scrambling results, but the key space in this method is less). In addition, the correlation findings for V, H, and D are close to zero or less than zero, as shown in Table 1.11 and Figure 4.12.

As shown in Figure 4.11, the findings demonstrated that the recovered image is like the original image. The recovered image has the same entropy as the original image, with SSIM = 1, MSE = 0, and PSNR = infinity. The histograms of the encrypted images and the original images are identical.

4.8.5 IEHSB Method: Results of Scrambling Bits in Rows & Columns

The simulation results for this method's statistical encryption of seven images are shown in Table 4.15. Table 4.16 shows the Vertical (V), Horizontal (H), and Diagonal (D) appearance numeric correlation for the seven images, and Figure 4.14 shows the correlation plots for one digital encrypted image. Figure 4.13 shows the original, encryption, and decryption images of two images and the histogram for each image.

Table 4.15: Simulation Results of Scrambling Bits in Each Row & Column.

Proposed Method	Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Sec)	
						Enc.	Dec.
Row-Column Merge All Colors	Image1	0.006	7.966	12103	7.302	6.764	5.790
	Image2	0.006	7.286	9088.3	8.546	1.446	1.416
	Image3	0.006	7.049	8333.6	8.323	1.027	0.973
	Image4	0.007	7.821	9904.6	8.172	3.867	3.724
	Image5	0.006	7.838	13103	6.957	1.815	1.733
	Image6	0.008	7.926	8462.2	8.183	0.506	0.546
	Image7	0.004	7.769	14592	6.489	4.539	4.392
Row-Column Without Merge	Image1	0.0220	7.944	9608.7	8.304	5.659	5.550
	Image2	0.006	7.286	9065.8	8.556	1.274	1.244

Proposed Method	Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Sec)	
						Enc.	Dec.
Colors	Image3	0.013	7.048	8319.3	8.330	1.039	0.981
	Image4	0.011	7.821	9860.6	8.191	3.952	4.024
	Image5	0.008	7.838	13078	6.965	1.833	1.907
	Image6	0.028	7.926	8352.3	8.240	0.556	0.553
	Image7	0.0265	7.559	11367	7.574	4.524	4.355

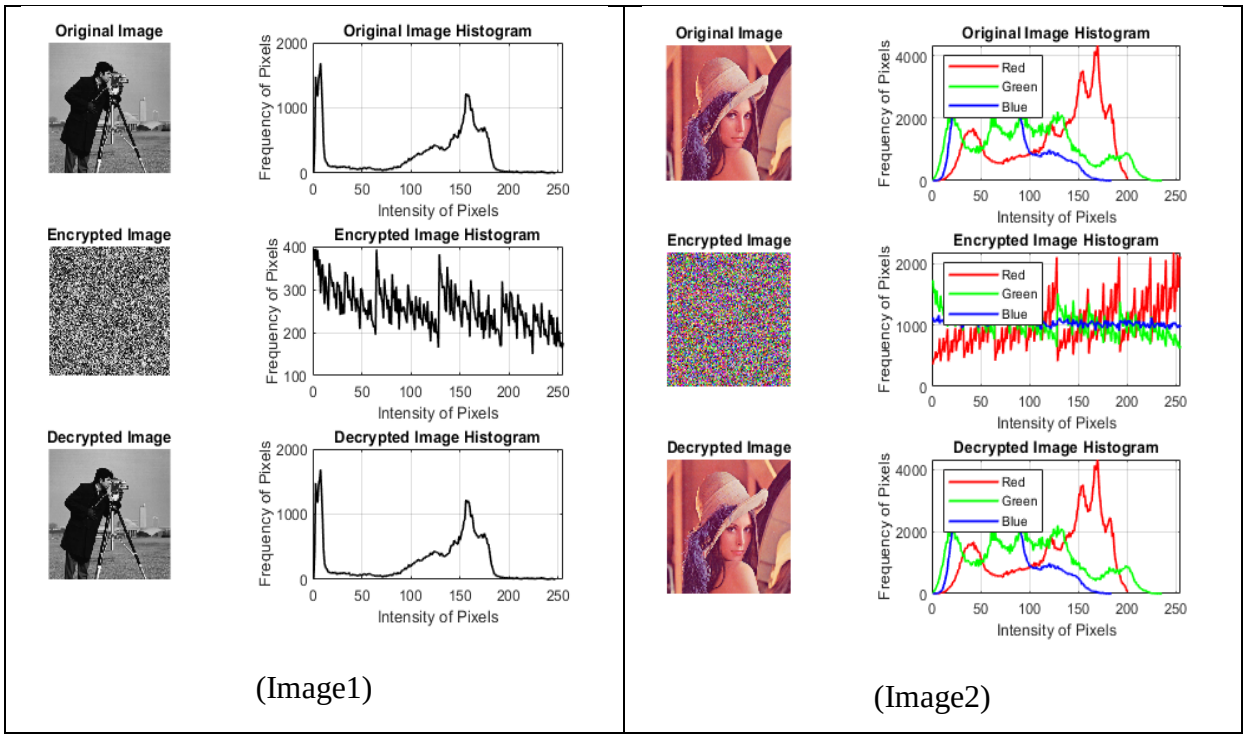


Figure 4.13: Enc. & Dec. Image1 Based on Row-Column Scrambling.

Table 4.16: V, H, & D Correlations for Row-Column Scrambling.

Image	Method	V	H	D	Method	V	H	D
1	Row Merge All Colors	0.15199	0.1561	0.0241	Row without Merge Colors	0.00635	0.00052	0.00114
2		-0.0015	-0.0008	-0.0011		-0.0044	-0.0014	-0.0049
3		0.0002	-2e-4	-2e-4		-0.0023	-0.0043	-0.0023
4		-0.0001	-2e-3	-0.0014		-0.0005	-0.0030	0.0010
5		-0.0014	-0.0026	-0.0021		-0.0063	0.0002	0.0006
6		-0.0022	-0.0019	0.0053		0.0019	-0.0037	-0.0004
7		0.0016	0.0081	-0.0066		0.0002	0.0048	-0.0008

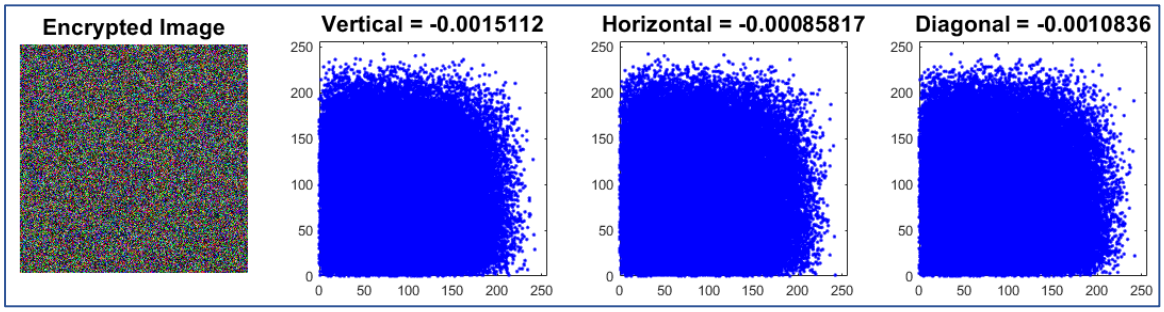


Figure 4.14: V, H, and D Correlation Coefficients for Enc. Image2 Based on IEHSB Method (Scrambling Bits Row-Column Merege All Colors).

From Table 4.15 and Figure 4.13, the encryption results are the best from all the above scrambling methods in all testing measurements. The V, H, and D correlation results are near zero or less than zero, as shown in Table 4.16 and Figure 4.14.

As shown in Figure 4.13, the results proved that the recovered image is identical the original. The recovered image has the same entropy as the original image, with SSIM = 1, MSE = 0, and PSNR = infinity. The histogram of the decrypted images is identical to the histogram of the original images.

4.8.6 Discussions of The IEHSB Method

Any slight change in keys between encryption and decryption keys does not allow the recovery of the original image from the encrypted image. All reliable cryptosystems need high levels of sensitivity. If the Initial conditions of X0 or Y0 or Z0 or W0 is changed by a tiny amount; this simple change, such as $1*10^{-15}$, cannot retrieve information by the Attacker, as displayed in **Error! Reference ource not found.** and **Error! Reference source not found.**. In this method, the key space depends on five parameters [r, a, b, c, and d] and four initial conditions [X0, Y0, Z0, and W0] and the two very big and minor number are used in the mod process. There are 11 coefficients used in the system.

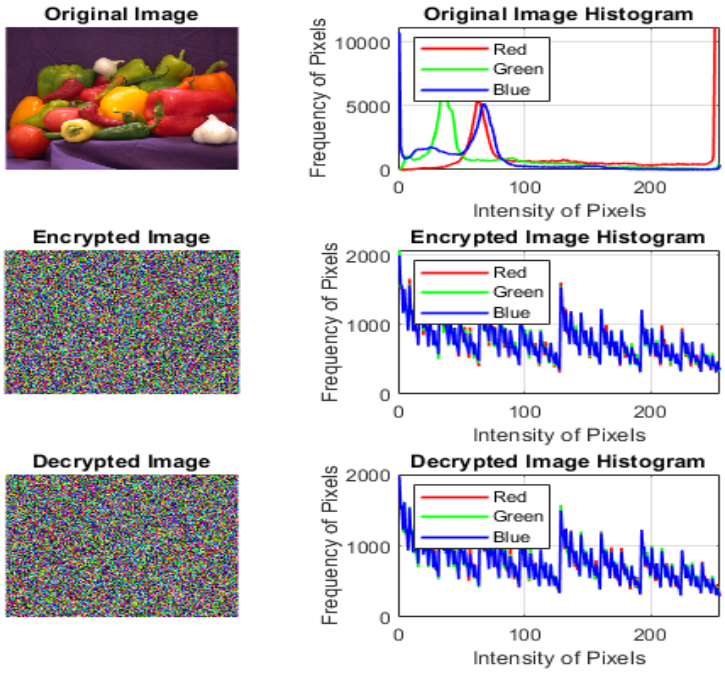


Figure 4.15: Sensitivity for Changing a-Parameter by 10-15

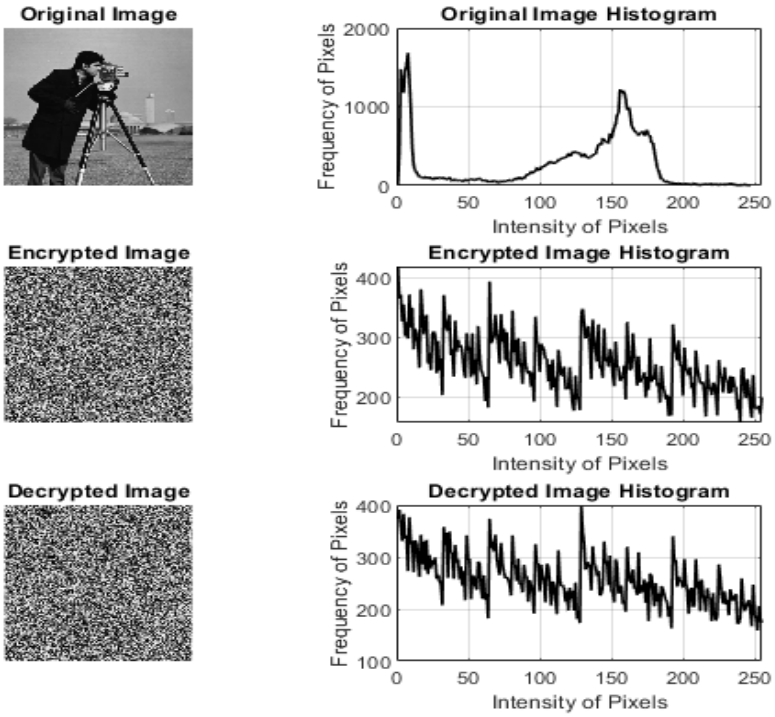


Figure 4.16:Sensitivity for Changing X(1) by 10-15

The key space of this article is at least.

$$\boldsymbol{Keyspace(1HCRS) = \prod_{1}^{D} \frac{1}{10^{-15}}} \quad (4\text{-}1)$$

$$= \left(\frac{1}{10^{-15}}\right)^{11} = \boldsymbol{10^{15*11} = 10^{165}} \quad (4\text{-}2)$$

$$\boldsymbol{Keyspace(SBHC) = 10^{165} = 2^{548}} \quad (4\text{-}3)$$

In Row-Column Scrambling based on two Hyper-chaotic (two-dimensions), the key space is equal:

$$\boldsymbol{Keyspace(SBHC_{Row-colom}) = \left(2^{548}\right)^{2} = 2^{1096}} \quad (4\text{-}4)$$

4.9 The Simulation Results of the Image Encryption Based on Three Hyper-Chaotic Signals (IE3HS) Method

The simulation results for this method are presented in Table 4.17 for the statistical encryption results for the seven images. Table 4.18 shows the Vertical (V), Horizontal (H), and Diagonal (D) appearance numeric correlation for the seven images, and Figure 4.20 shows the correlation plots for one digital encrypted image.

Figure 4.17, Figure 4.18, and Figure 4.19 show the original, encryption and decryption of three images and the histogram for each image.

Table 4.17: Encryption Simulation Results Based on IE3HS Method

Image	SSIM	Entropy	MSE	PSNR [dB]	Delay (Sec)	
					Enc.	Dec.
Image1	0.005	7.999	12728	7.083	1.193	1.081
Image2	0.005	7.999	14101	6.638	0.196	0.183
Image3	0.004	7.9993	14230	5.999	0.154	0.172
Image4	0.006	7.999	12459	7.176	0.558	0.623
Image5	0.006	7.999	13865	6.711	0.334	0.294
Image6	0.008	7.998	9557.4	7.654	0.069	0.085
Image7	0.004	7.999	16706	5.902	0.764	0.778

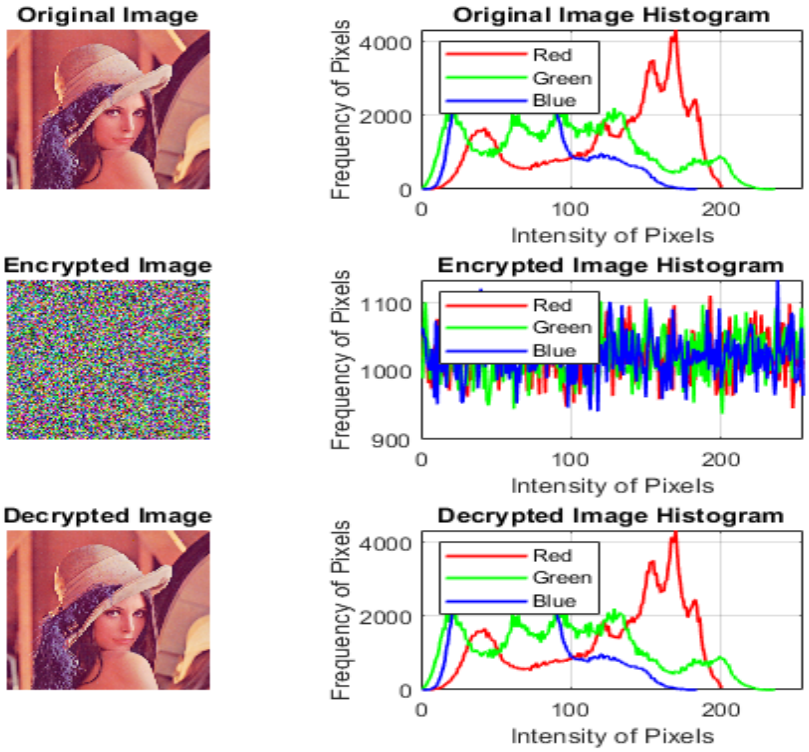


Figure 4.17: Image2 Encryption Based on IE3HS Method

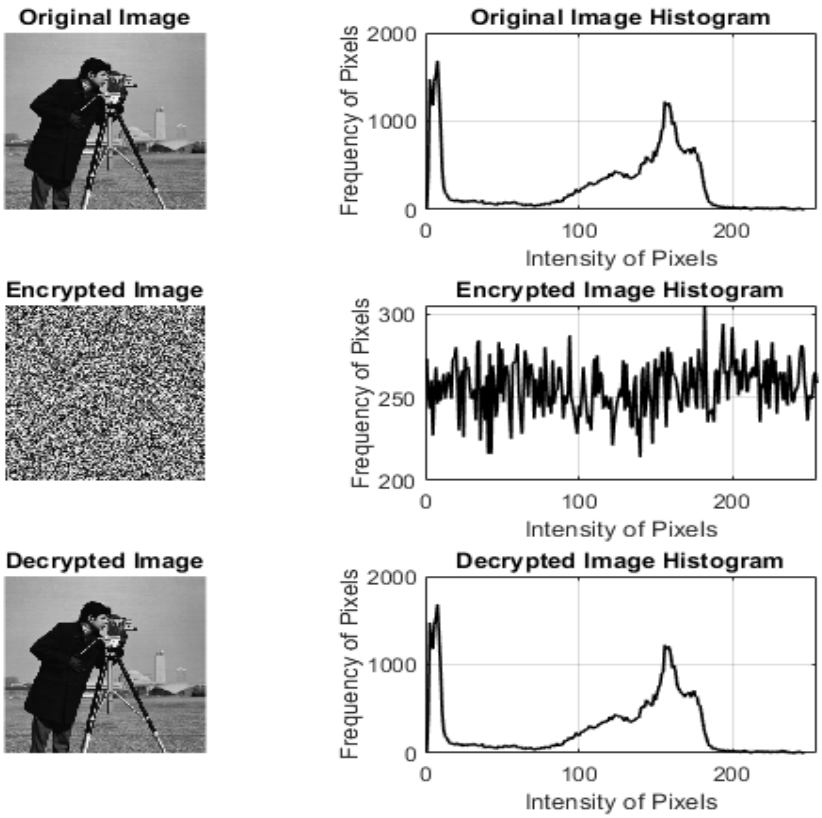


Figure 4.18: Image1 Encryption Based on IE3HS Method

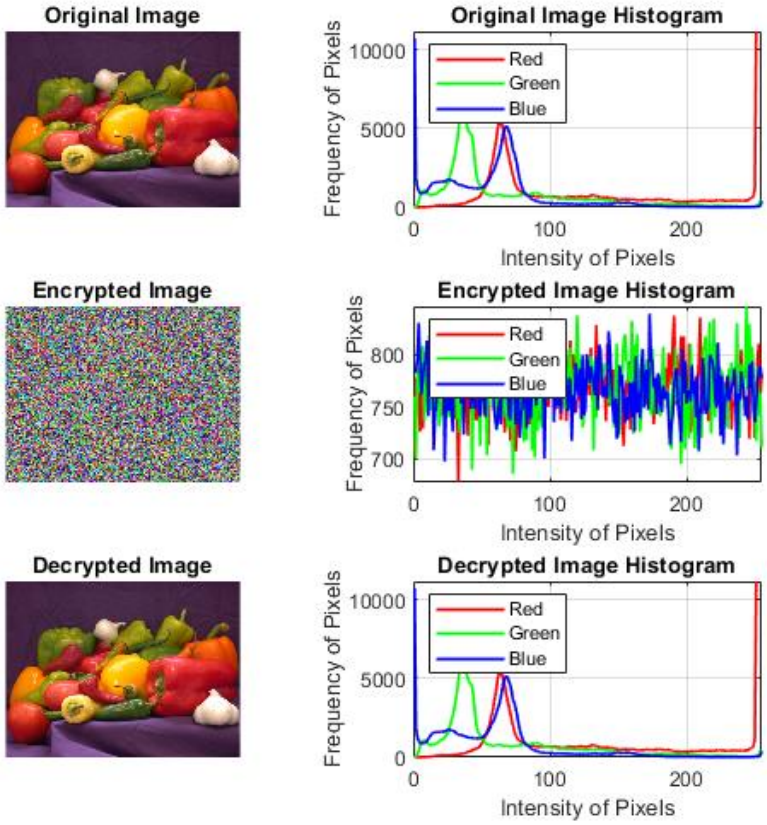


Figure 4.19: Image3 Encryption Based on IE3HS Method

Table 4.18: V, H, and D Correlation Coefficients Based on IE3HS Method

Image	V	H	D
1	0.15623	0.15386	0.02555
2	-0.00034	0.00099	-0.00219
3	0.00098	0.00091	-0.00272
4	0.00085	-0.00105	-2e-05
5	-0.00111	-0.00077	-0.00254
6	0.00367	-0.00893	0.00065
7	-0.00167	-0.00689	-0.0017

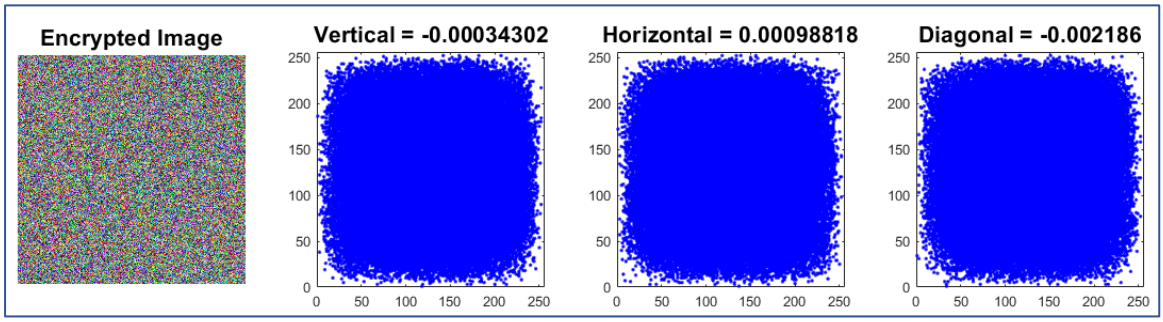


Figure 4.20: V, H, and D Correlation Coefficients Based on IE3HS Method

The simulation results from Table 4.17, Show that the testing measurements used to evaluate the encryption performance are complete (PSNR is about 8 dB, SSIM is close to zero, and entropy very relative to 8). The results based on images and histograms are shown in Figure 4.17, Figure 4.18, and Figure 4.19. The simulation results reveal that the encrypted image's histogram is flat, suggesting that virtually all conceivable pixel values happened. Regarding protecting the image, a circumstance like this is certainly important.

These correlation coefficients insignificant or zero for encrypted image. These correlation coefficients for V, H, and D are shown in Table 4.18 and Figure 4.20.

The simulation results proved that the recovered image is the identical to the original image, as shown in Figure 4.17 to Figure 4.19. The entropy of the recovered image is the same entropy of the original image, SSIM equal to 1, MSE equal to zero, PSNR equal to infinite, and the histogram of the decrypted images is identical to the histogram of the original images.

4.9.1 Discussions of The IE3HS Method (Key Sensitivity and Space)

If there is even the slightest difference in the encryption and decryption keys, the encrypted image it will not be decrypted. Sensitivity is crucial for all safe cryptosystems. This means that the Attacker cannot obtain any information by changing even a single element of the key—the starting state of X0, Y0, Z0, or

W0—by a small amount, as shown in **Error! Reference source not found.**, REF _Ref139892321 \h * MERGEFORMAT **Error! Reference source not found.**, and **Error! Reference source not found.**.

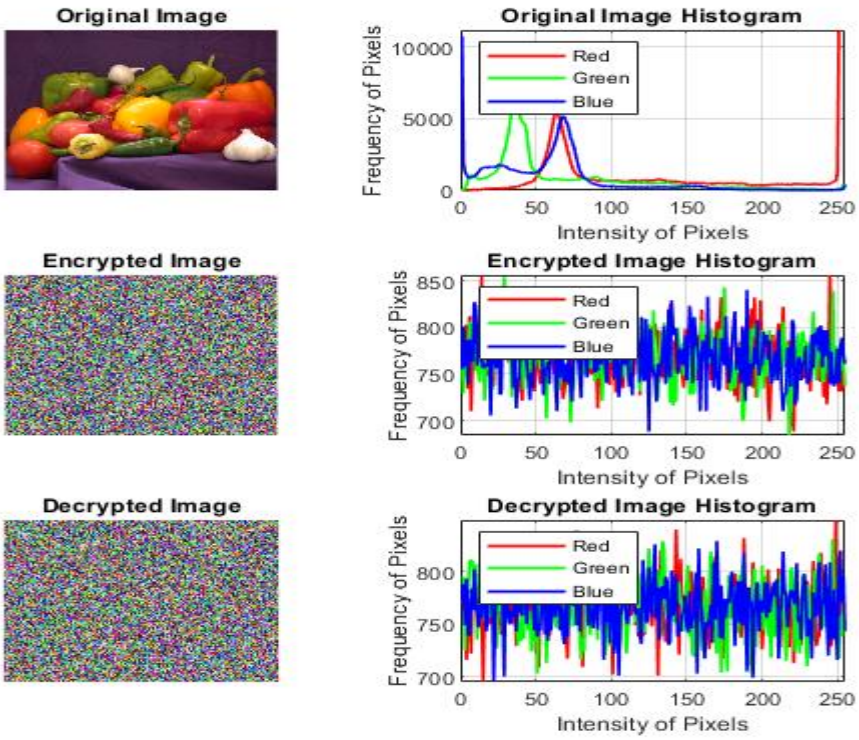


Figure 4.21:Dec. Results by Changing X(1) for First HCRS by 1e-15

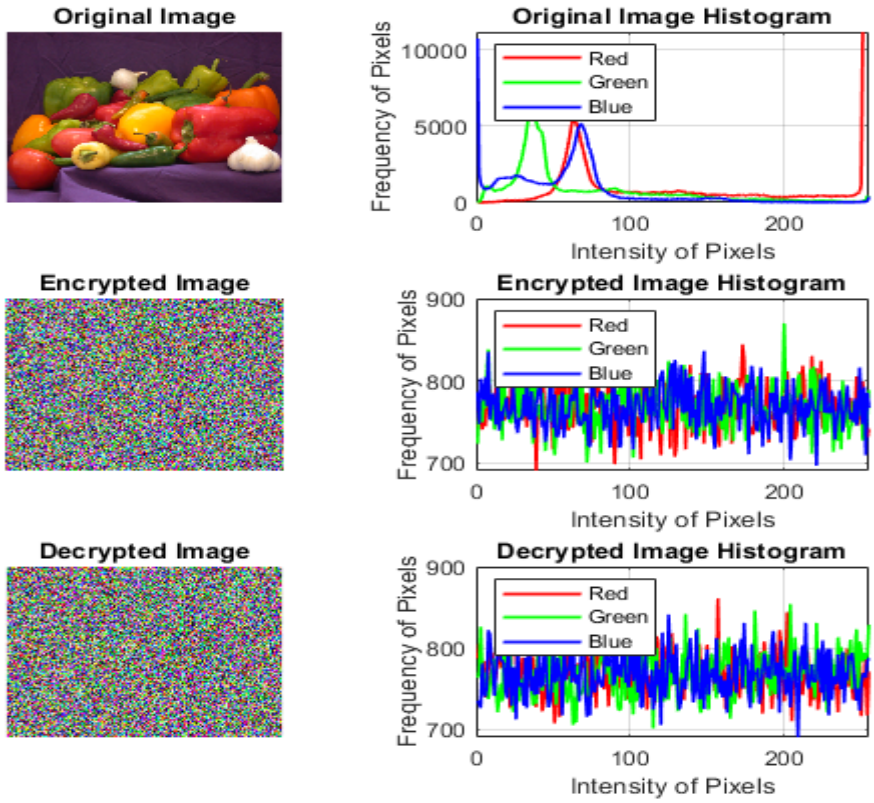


Figure 4.22:Results by Changing X(1) for Second HCRS by 1e-15

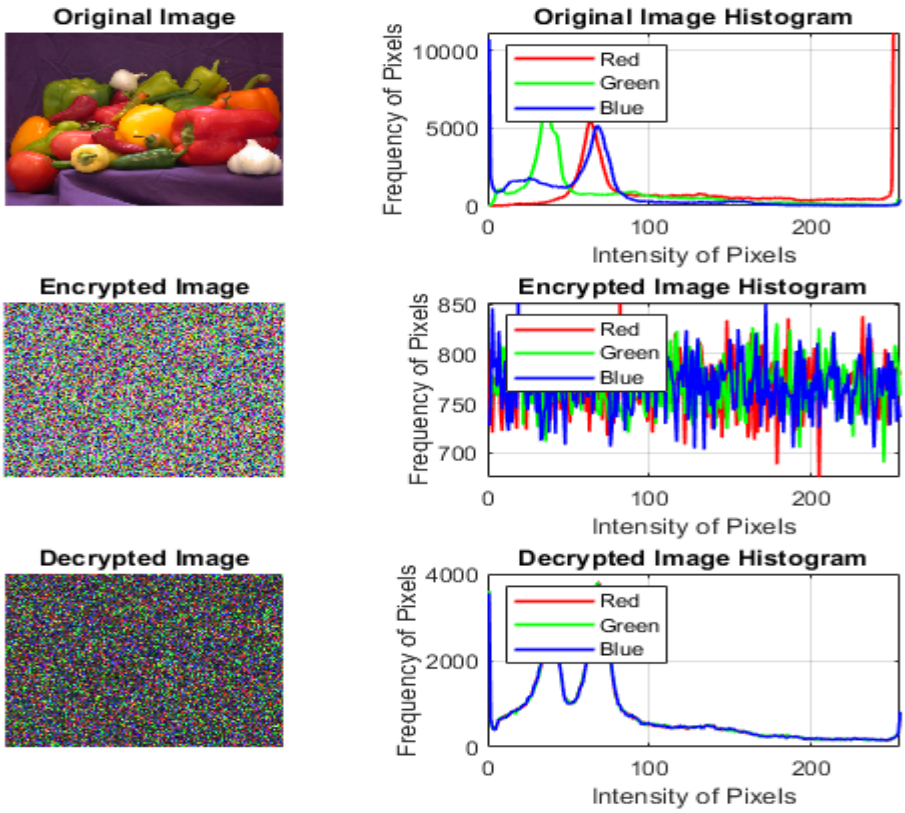


Figure 4.23:Dec. Results by Changing X(1) for Thered HCRS by 1e-15.

From the above figures, it is noticed that a tiny change is made in the first and second Hyper-chaos generators, the histogram of the decryption image will remain encrypted. In contrast, a slight change is made in the third Hyper-chaos, the histogram of the decrypted image will be returned as before encryption, but the image will not be recovered and will remain encrypted. The reason for this is that the pixels of the image are recovered but not in their correct locations.

The three Hyper-chaotic generators determine the keyspace of the image encryption system employed in this method. The key space is calculated based on the three generators:

$$\textbf{Keyspace(IE3HS Method)} = \left(2^{548}\right)^3 = 2^{1644} \qquad (4\text{-}5)$$

4.10 The Differences Between the IHESB and IE3HS Methods

IEHSB method for image encryption using Hyper-chaos Scrambling Bits. It uses a single Hyper-chaotic system to generate a sequence of random numbers, which are then used to scramble the bits of an image. The IE3HS method for Image encryption uses three Hyper-chaos systems. It uses three Hyper-chaotic system to generate three sequences of random numbers. The main difference between the two methods is the number of Hyper-chaotic systems used. IEHSB uses a single system, while IE3HS uses three systems; IE3HS has a larger key space than IEHSB, making it more secure.

The performance of the two methods is also different in the following points:

- **Keyspace:** The key space of IEHSB is 2^{548} or 2^{1096} in the Row-Column case, while the key space of IE3HS is 2^{1644}; this means there are more possible keys for IE3HS, making it more secure.
- **Security:** The IE3HS method is more secure than IEHSB because it has a larger key space. It is also more difficult to attack because the bits of the image are scrambled in three passes.
- **Complexity:** IE3HS is more complex than IEHSB because it requires three Hyper-chaotic systems; this makes it more difficult to implement and use.
- **Applications:** IE3HS is more suitable for applications where security is the primary concern. IEHSB is more suitable for applications where low complexity is also important.

The summarized difference between the two proposed methods is shown in Table 4.19.

Table 4.19: The Main Differences Between IEHSB and IE3HS Methods.

Feature	IEHSB Method	IE3HS Method
Number of hyper-chaotic systems	1 or 2	3
Keyspace	2^{548} or 2^{1096}	2^{1644}
Security	Less secure	More secure
Performance	Slower	Faster
Complexity	Less complex	More complex
Applications	In lower-complexity applications	Security is important (Higher complexity applications)

4.11 Comparison of The Proposed System with Other Research

To evaluate the encryption strength of the IEHSB and IE3HS methods by comparing the simulation results with the other articles, as shown in Table 4.20. While

Table 4.21 compares based on the key space.

Table 4.20: Comparing The Proposed Methods with the Other Articles

References	Image	MSE	PSNR (dB)	Mean Corr	Entropy
Wassim Alexan et al.[1]	Peppers	10033	8.1624	0.00175	7.9968
Xingyuan Wang ,et al. [78]	Peppers	-	-	0.0035	7.9959
Huda H. Alsaabri, et al. [32]	Cameraman	-	8.9311	0.5879	7.9973
	Peppers	-	8.6262	0.4917	7.9997
Min Long ,et al. [79]	Peppers	-	-	0.0017	7.9976
Ibrahim Yasser ,et al. [80]	Cameraman	9445	8.38	-	7.9991
	Peppers	8413	8.88	-	7.9994
IEHSB Method (one vector)	Cameraman	9388.1	8.336	-0.006	7.973
	Peppers	9959.4	8.148	0.0007	7.906
IE3HS Method	Cameraman	9474.5	8.296	-0.0006	7.378
	Peppers	11259	7.615	0.002	7.378

Table 4.21: Key Space Comparison of The Proposed Methods and Other Articles

References	Key Space
Renzhi Li , et al. [81]	2^{318}
Hikmat N. Abdullah,et al. [82]	2^{166}
Ameer K. Jawad, et al. [83]	2^{266}
Chang et al.[7]	2^{256}
Shakir et al.[8]	2^{627}
Saad S. Hreshee, et al. [84]	2^{266}
Hikmat N. Abdullah, et al. [85]	2^{319}
Proposed IEHSB Method: Except Row-Column Scrambling	2^{548}
Proposed IEHSB Method: Row-Column Scrambling Case	2^{1096}
Proposed IE3HS Method	2^{1644}

Table 4.21 shows that the proposed system has a huge keyspace compared to the other articles.

Chapter Five

Conclusions and Suggestions

5.1 Introduction

This chapter illustrates conclusions and suggestions for future works after applying the proposed system.

5.2 Conclusions

1. Lyapunov Exponents need the dynamical equations and all parameters to check whether the system is chaotic. At the same time, the CNN-Tester needed only signals to test whether it was chaotic or not.

2. The delay in testing the Rabinovitch parameters by Lyapunov is about 1.5077 seconds for one set of parameters. At the same time, the delay by the CNN-Tester is about 0.0112 seconds. Testing the chaotic signals based on CNN-Tester gives a good classification result and is faster than testing Lyapunov exponents (about 135 times)

3. The optimal accuracy of the proposed system is 100% for testing the signals of the Rabinovitch system (chaotic or not chaotic signals) by CNN- Tester.

4. Lyapunov Exponents assume a deterministic system, limiting their applicability in real-world scenarios, while CNN-Tester is acceptable to use in real-world scenarios.

5. The pixel scrambling method had the worst encryption results, with high SSIM values indicating high similarity to the original image. Encryption time was considerable, with some images taking around 50 seconds.

6. Scrambling bits in each row or column produced more acceptable results than the pixel scrambling method.

7. Row-column and one-vector scrambling methods had the best encryption results, with V, H, and D correlation values near zero. However, the key space

for the one-vector method (about 2^{548}) was smaller than row-column scrambling (about 2^{1096}).

8. The IE3HS method showed key sensitivity, with small changes in Hyper-chaotic generators retaining encryption but disrupting recovery. The key space was significant, with a value of 2^{1644}.

9. The IE3HS method demonstrated higher encryption performance than the IEHSB method cases, with lower PSNR and correlation coefficient values. IE3HS Method showed better security and efficiency, making it a more promising choice for image encryption.

5.3 Suggestions For Future Works

1. Find the best deep learning method for diverse signal kinds and applications, compare CNN-based chaos detection to another machine learning methods.

2. Explore techniques to optimize the encryption and decryption processes to reduce computational time making the methods more efficient for real-time applications.

3. Investigate the robustness of the IEHSB and IE3HS methods against various attacks, including brute-force, statistical, and adversarial attacks.

4. Evaluate the security and performance of the encryption methods using a broader range of digital image datasets and other types of images.

5. Explore hardware-based implementations of the encryption methods to achieve higher processing speeds and lower energy consumption; this could benefit resource-constrained environments and IoT applications.

References

[1] W. Alexan, M. Elkandoz, M. Mashaly, E. Azab, and A. Aboshousha, “Color Image Encryption Through Chaos and KAA Map,” *IEEE Access*, vol. 11, pp. 11541–11554, 2023.

[2] F. Castro, D. Impedovo, and G. Pirlo, “A Medical Image Encryption Scheme for Secure Fingerprint-Based Authenticated Transmission,” *Appl. Sci.*, vol. 13, no. 10, p. 6099, 2023.

[3] J. R. Lakshmi, “EXPLORING A NOVEL STRATEGY FOR ENSURING THE SECURITY OF MEDICAL IMAGES THROUGH ENCRYPTION AND QR ENCODING,” *Ind. Eng. J.*, vol. 52, no. 4, 2023.

[4] U. Erkan, A. Toktas, S. Enginoğlu, E. Akbacak, and D. N. H. Thanh, “An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN,” *Multimed. Tools Appl.*, vol. 81, no. 5, pp. 7365–7391, 2022.

[5] H. Nematzadeh, R. Enayatifar, H. Motameni, F. G. Guimarães, and V. N. Coelho, “Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices,” *Opt. Lasers Eng.*, vol. 110, pp. 24–32, 2018.

[6] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[7] H. Chang, E. Wang, and J. Liu, “Research on Image Encryption Based on Fractional Seed Chaos Generator and Fractal Theory,” *Fractal Fract.*, vol. 7, no. 3, p. 221, 2023.

[8] H. R. Shakir, S. A. Mehdi, and A. A. Hattab, “A new four-dimensional hyper-chaotic system for image encryption,” *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, p. 1744, 2023.

[9] X. Li and H. Peng, “Chaotic medical image encryption method using attention mechanism fusion ResNet model,” *Front. Neurosci.*, vol. 17, Jul. 2023, doi: 10.3389/fnins.2023.1226154.

[10] H. Mahalingam, T. Veeramalai, A. R. Menon, and R. Amirtharajan, “Dual-Domain Image Encryption in Unsecure Medium—A Secure Communication Perspective,” *Mathematics*, vol. 11, no. 2, p. 457, 2023.

[11] H. Mazin, M. Alibraheemi, Q. Al-gayem, and E. A. Hussein, “Design and FPGA Implementation of High-Speed Cryptographic System for Wireless Communications Based on Multi-Dimensional Hyperchaotic Generator,” *Neuro Quantology*, vol. 20, no. 7, pp. 559–573, 2022, doi: 10.14704/nq.2022.20.7.NQ33073.

[12] A. A. Alarood, E. Alsolami, M. A. Al-Khasawneh, N. Ababneh, and W. Elmedany, “IES: Hyper-chaotic plain image encryption scheme using improved shuffled confusion-diffusion,” *Ain Shams Eng. J.*, vol. 13, no. 3, p. 101583, 2022.

[13] N. Iqbal, M. Hanif, Z. U. Rehman, and M. Zohaib, “An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN,” *Multimed. Tools Appl.*, vol. 81, no. 6, pp. 8107–8137, 2022.

[14] R. Zhang *et al.*, “A novel plaintext-related color image encryption scheme based on cellular neural network and Chen’s chaotic system,” *Symmetry (Basel).*, vol. 13, no. 3, p. 393, 2021.

[15] X. Wang, Y. Su, C. Luo, and C. Wang, “A novel image encryption

algorithm based on fractional order 5D cellular neural network and Fisher-Yates scrambling," *PLoS One*, vol. 15, no. 7, p. e0236015, 2020.

[16] J. Zheng, Z. Luo, and Z. Tang, "An image encryption algorithm based on multichaotic system and DNA coding," *Discret. Dyn. Nat. Soc.*, vol. 2020, pp. 1–16, 2020.

[17] X.-H. Song, H.-Q. Wang, S. E. Venegas-Andraca, and A. A. Abd El-Latif, "Quantum video encryption based on qubit-planes controlled-XOR operations and improved logistic map," *Phys. A Stat. Mech. its Appl.*, vol. 537, p. 122660, 2020, doi: 10.1016/j.physa.2019.122660.

[18] L. Gong, K. Qiu, C. Deng, and N. Zhou, "An image compression and encryption algorithm based on chaotic system and compressive sensing," *Opt. Laser Technol.*, vol. 115, pp. 257–267, 2019.

[19] W. K. Lee, R. C. W. Phan, W. S. Yap, and B. M. Goi, "a novel parallel chaos-based image encryption scheme," *Springer, Nonlinear Dyn.*, vol. 92, no. 2, pp. 575–593, 2018, doi: 10.1007/s11071-018-4076-6.

[20] M. Ashtiyani, P. Birgani, and S. Madahi, "Speech Signal Encryption Using Chaotic Symmetric Cryptography," *Basic Appl. Sci. Res.*, vol. 2, no. 2, pp. 1678–1684, 2012.

[21] W. Al Nassan, T. Bonny, and A. Baba, "A New Chaos-Based Cryptoystem for Voice Encryption," *IEEE Xplore, Int. Conf. Signal Process. Inf. Secur. ICSPIS 2020*, vol. 3, pp. 1–4, 2020, doi: 10.1109/ICSPIS51252.2020.9340132.

[22] M. Ahmad, "Chaos Based Mixed Keystream Generation for Voice Data Encryption," *Int. J. Cryptogr. Inf. Secur.*, vol. 2, no. 1, pp. 39–48, 2012, doi: 10.5121/ijcis.2012.2104.

[23] M. Lawnik, L. Moysis, and C. Volos, “Chaos-Based Cryptography: Text Encryption Using Image Algorithms,” *Electron.*, vol. 11, no. 19, pp. 1–13, 2022, doi: 10.3390/electronics11193156.

[24] A. J. Mansoor, H. N. Abdullah, M. F. Al-Gailani, and H. T. Ziboon, “Chaotic encryption system based on pixel value and position transformation for color images,” *18th IEEE Int. Multi-Conference Syst. Signals Devices, SSD 2021*, no. March, pp. 433–439, 2021, doi: 10.1109/SSD52085.2021.9429368.

[25] Y. Pourasad, R. Ranjbarzadeh, and A. Mardani, “A new algorithm for digital image encryption based on chaos theory,” *Entropy*, vol. 23, no. 3, p. 341, 2021.

[26] S. F. Yousif, A. J. Abboud, and R. S. Alhumaima, “A new image encryption based on bit replacing, chaos and DNA coding techniques,” *Multimed. Tools Appl.*, vol. 81, no. 19, pp. 27453–27493, 2022.

[27] P. Ping, X. Zhang, X. Yang, and Y. A. A. Hashems, “A novel medical image encryption based on cellular automata with ROI position embedded,” *Multimed. Tools Appl.*, vol. 81, no. 5, pp. 7323–7343, 2022.

[28] Z. Liehuang, L. Wenzhuo, L. Lejian, and L. Hong, “A novel image scrambling algorithm for digital watermarking based on chaotic sequences,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 6, no. 8B, pp. 125–130, 2006.

[29] A. Mahdi, A. K. Jawad, and S. S. Hreshee, “Digital chaotic scrambling of voice based on duffing map,” *Int. J. Inf. Commun. Sci.*, vol. 1, no. 2, pp. 16–21, 2016.

[30] H. N. Abdullah, S. S. Hreshee, and A. K. Jawad, “Design of Efficient noise reduction scheme for secure speech masked by chaotic signals,” *J. Am. Sci.*,

vol. 11, no. 7, pp. 49–55, 2015.

[31] L. Dieci and E. S. Van Vleck, “Computation of a few Lyapunov exponents for continuous and discrete dynamical systems,” *Appl. Numer. Math.*, vol. 17, no. 3, pp. 275–291, 1995.

[32] H. H. Alsaabri and S. S. Hreshee, “Robust Image Encryption Based on Double Hyper Chaotic Rabinovich System,” *7th Int. Conf. Contemp. Inf. Technol. Math. ICCITM 2021*, pp. 146–152, 2021, doi: 10.1109/ICCITM53167.2021.9677722.

[33] R. B. Naik and U. Singh, “A review on applications of chaotic maps in pseudo-random number generators and encryption,” *Ann. Data Sci.*, pp. 1–26, 2022.

[34] D. Petkevičiūtė-Gerlach, R. Šmidtaitė, and M. Ragulskis, “Intermittent bursting in the fractional difference logistic map of matrices,” *Int. J. Bifurc. Chaos*, vol. 32, no. 01, p. 2230002, 2022.

[35] G. La Forgia, D. Cavaliere, S. Espa, F. Falcini, and G. Lacorata, “Numerical and experimental analysis of Lagrangian dispersion in two-dimensional chaotic flows,” *Sci. Rep.*, vol. 12, no. 1, p. 7461, 2022.

[36] A. M. Raheema, S. B. Sadkhan, and S. M. A. Satar, “Performance Enhancement of Speech Scrambling Techniques Based on Many Chaotic Signals,” in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020, pp. 308–313.

[37] H. A.-J. Al-Asady, O. Q. J. Al-Thahab, and S. S. Hreshee, “Robust encryption system based watermarking theory by using chaotic algorithms: A reviewer paper,” in *Journal of Physics: Conference Series*, 2021, vol. 1818, no. 1, p. 12086.

[38] J. Li and N. Cui, “A hyperchaos generated from Rabinovich system,” *AIMS Math.*, vol. 8, no. 1, pp. 1410–1426, 2023.

[39] H. J. Yakubu, “A More Secure Image Encryption Algorithm Using Dual 3-Dimensional Chaotic Maps for RGB Images,” *Int. J. Comput. Trends Technol.*, vol. 68, no. 10, pp. 35–43, 2020.

[40] H. Zhu, J. Ge, W. Qi, X. Zhang, and X. Lu, “Dynamic analysis and image encryption application of a sinusoidal-polynomial composite chaotic system,” *Math. Comput. Simul.*, vol. 198, pp. 188–210, 2022.

[41] R. Krishnan, P. Rajpurkar, and E. J. Topol, “Self-supervised learning in medicine and healthcare,” *Nat. Biomed. Eng.*, vol. 6, no. 12, pp. 1346–1352, 2022.

[42] S. R. Dubey and S. Chakraborty, “Average biased ReLU based CNN descriptor for improved face retrieval,” *Multimed. Tools Appl.*, vol. 80, pp. 23181–23206, 2021.

[43] I. Kouretas and V. Paliouras, “Simplified hardware implementation of the softmax activation function,” in *2019 8th international conference on modern circuits and systems technologies (MOCAST)*, 2019, pp. 1–4.

[44] U. Cinar, R. Cetin Atalay, and Y. Y. Cetin, “Human Hepatocellular Carcinoma Classification from H&E Stained Histopathology Images with 3D Convolutional Neural Networks and Focal Loss Function,” *J. Imaging*, vol. 9, no. 2, p. 25, 2023.

[45] R. S. Kuzu, E. Maiorana, and P. Campisi, “Loss functions for CNN-based biometric vein recognition,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 750–754.

[46] S. Nurmaini and A. Gani, “Cardiac Arrhythmias Classification Using Deep

Neural Networks and Principle Component Analysis Algorithm.," *Int. J. Adv. Soft Comput. Its Appl.*, vol. 10, no. 2, pp. 1–20, 2018.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[48] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient deep network architectures for fast chest X-ray tuberculosis screening and visualization," *Sci. Rep.*, vol. 9, no. 1, p. 6268, 2019.

[49] I. Banerjee *et al.*, "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification," *Artif. Intell. Med.*, vol. 97, pp. 79–88, Jun. 2019, doi: 10.1016/j.artmed.2018.11.004.

[50] M. Bouhadida *et al.*, "Neutron spectrum unfolding using two architectures of convolutional neural networks," *Nucl. Eng. Technol.*, vol. 55, no. 6, pp. 2276–2282, Jun. 2023, doi: 10.1016/j.net.2023.03.025.

[51] A. K. Ozcanli and M. Baysal, "Islanding detection in microgrid using deep learning based on 1D CNN and CNN-LSTM networks," *Sustain. Energy, Grids Networks*, vol. 32, p. 100839, 2022.

[52] S. Tian, S. Wang, and H. Xu, "Early detection of freezing damage in oranges by online Vis/NIR transmission coupled with diameter correction method and deep 1D-CNN," *Comput. Electron. Agric.*, vol. 193, p. 106638, 2022.

[53] D. Cornelisse, "An intuitive guide to convolutional neural networks," *Free Code Camp*, pp. 1–19, 2018.

[54] S. Harbola and V. Coors, "One dimensional convolutional neural network architectures for wind prediction," *Energy Convers. Manag.*, vol. 195, pp.

70–75, 2019.

[55] A. Ajit, K. Acharya, and A. Samanta, “A review of convolutional neural networks,” in *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, 2020, pp. 1–5.

[56] A. Dureja and P. Pahwa, “Analysis of non-linear activation functions for classification tasks using convolutional neural networks,” *Recent Patents Comput. Sci.*, vol. 12, no. 3, pp. 156–161, 2019.

[57] R. Prabhu, “Understanding of convolutional neural network (CNN)—deep learning,” *Mediu. Com*, vol. 1, no. 11, 2018.

[58] T.-C. Lu, “CNN Convolutional layer optimisation based on quantum evolutionary algorithm,” *Conn. Sci.*, vol. 33, no. 3, pp. 482–494, 2021.

[59] S. Saha, “A comprehensive guide to convolutional neural networks—the ELI5 way,” *Towar. data Sci.*, vol. 15, p. 15, 2018.

[60] A. Dertat, “Applied deep learning-part 4: Convolutional neural networks,” *Towar. Data Sci.*, vol. 26, 2017.

[61] A. Deshpande, “A beginner’s guide to understanding convolutional neural networks,” *Retrieved March*, vol. 31, no. 2017, 2016.

[62] S. D. Thepade, M. R. Dindorkar, P. R. Chaudhari, and S. V Bang, “Enhanced Face Presentation Attack Prevention Employing Feature Fusion of Pre-trained Deep Convolutional Neural Network Model and Thepade’s Sorted Block Truncation Coding,” *Int. J. Eng. Trans. A Basics*, vol. 36, no. 4, pp. 807–816, 2023.

[63] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, “A comprehensive survey of loss functions in machine learning,” *Ann. Data Sci.*, pp. 1–26, 2020.

[64] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, “Conceptual understanding of convolutional neural network-a deep learning approach,” *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018.

[65] A. Shrestha, H. Fang, Q. Wu, and Q. Qiu, “Approximating back-propagation for a biologically plausible local learning rule in spiking neural networks,” in *Proceedings of the International Conference on Neuromorphic Systems*, 2019, pp. 1–8.

[66] H. Pothina and K. V Nagaraja, “Artificial neural network and math behind it,” in *Smart Trends in Computing and Communications: Proceedings of SmartCom 2022*, Springer, 2022, pp. 205–221.

[67] C. Nastos, P. Komninos, and D. Zarouchas, “Non-destructive strength prediction of composite laminates utilizing deep learning and the stochastic finite element methods,” *Compos. Struct.*, vol. 311, p. 116815, 2023.

[68] L. Rice, E. Wong, and Z. Kolter, “Overfitting in adversarially robust deep learning,” in *International Conference on Machine Learning*, 2020, pp. 8093–8104.

[69] J. R. Tatz, C. Soh, and J. R. Wessel, “Towards a two-stage model of action-stopping: Attentional capture explains motor inhibition during early stop-signal processing,” *biorxiv*, pp. 2002–2021, 2021.

[70] N.-D. Hoang, “Image processing-based spall object detection using gabor filter, texture analysis, and adaptive moment estimation (Adam) optimized logistic regression models,” *Adv. Civ. Eng.*, vol. 2020, pp. 1–16, 2020.

[71] U. Erkan, A. Toktas, F. Toktas, and F. Alenezi, “2D eπ-map for image encryption,” *Inf. Sci. (Ny).*, vol. 589, pp. 770–789, 2022.

[72] U. Erkan, A. Toktas, and Q. Lai, “2D hyperchaotic system based on

Schaffer function for image encryption," *Expert Syst. Appl.*, vol. 213, p. 119076, 2023.

[73] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, and L. Vanneschi, "Structural similarity index (SSIM) revisited: A data-driven approach," *Expert Syst. Appl.*, vol. 189, p. 116087, 2022.

[74] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study," *J. Comput. Commun.*, vol. 7, no. 3, pp. 8–18, 2019.

[75] P. Rashmi, M. C. Supriya, and Q. Hua, "Enhanced lorenz-chaotic encryption method for partial medical image encryption and data hiding in big data healthcare," *Secur. Commun. Networks*, vol. 2022, pp. 1–9, 2022.

[76] A. R. Alharbi *et al.*, "A New Multistage Encryption Scheme Using Linear Feedback Register and Chaos-Based Quantum Map," *Complexity*, vol. 2022, 2022, doi: 10.1155/2022/7047282.

[77] A. H. ElSafty, M. F. Tolba, L. A. Said, A. H. Madian, and A. G. Radwan, "Hardware realization of a secure and enhanced s-box based speech encryption engine," *Analog Integr. Circuits Signal Process.*, vol. 106, no. 2, pp. 385–397, Feb. 2021, doi: 10.1007/S10470-020-01614-Z.

[78] X. Wang and Y. Su, "Color image encryption based on chaotic compressed sensing and two-dimensional fractional Fourier transform," *Sci. Rep.*, vol. 10, no. 1, pp. 1–19, 2020.

[79] M. Long and L. Tan, "A chaos-based data encryption algorithm for image/video," in *2010 Second international conference on multimedia and information technology*, 2010, vol. 1, pp. 172–175.

[80] I. Yasser, M. A. Mohamed, A. S. Samra, and F. Khalifa, "A chaotic-based

encryption/decryption framework for secure multimedia communications," *Entropy*, vol. 22, no. 11, p. 1253, 2020.

[81] R. Li, Q. Liu, and L. Liu, "Novel image encryption algorithm based on improved logistic map," *IET Image Process.*, vol. 13, no. 1, pp. 125–134, 2019.

[82] H. N. Abdullah, S. S. Hreshee, and A. K. Jawad, "Noise reduction of chaotic masking system using repetition method," *Unpublished*, pp. 1–12, 2015.

[83] A. K. Jawad, H. N. Abdullah, and S. S. Hreshee, "Secure speech communication system based on scrambling and masking by chaotic maps," in *International Conference on Advances in Sustainable Engineering and Applications, ICASEA 2018 - Proceedings*, 2018, pp. 1–6. doi: 10.1109/ICASEA.2018.8370947.

[84] S. S. Hreshee, H. N. Abdullah, and A. K. Jawad, "A high security communication system based on chaotic scrambling and chaotic masking," *Int. J. Commun. Antenna Propag.*, vol. 8, no. 3, pp. 257–264, 2018, doi: 10.15866/irecap.v8i3.13541.

[85] H. N. Abdullah, S. S. Hreshee, G. Karimi, and A. K. Jawad, "PERFORMANCE IMPROVEMENT OF CHAOTIC MASKING SYSTEM USING POWER CONTROL METHOD," *MESM'2022 Int. Program. Comm.*, pp. 19–23, 2022.

المستخلص

تقدم هذه الأطروحة نظاما جديدا لتشفير الصور الرقمية يعتمد على تقنيات متعددة الأبعاد شديدة الفوضى و "الشبكة العصبية التلافيفية (CNN)" لمعالجة مخاوف أمان البيانات والخصوصية في بيانات الصور الرقمية. يستفيد النظام من نقاط القوة في خوارزميات التشفير متعددة الأبعاد شديدة الفوضى وشبكات CNN لتحقيق أمان عال مع الحفاظ على سلامة الصور. يدرس البحث في تقنيات التشفير الحالية ويبين حدودها من حيث الأمان والكفاءة والحفاظ على جودة الصورة. يدمج النظام المقترح خوارزمية التشفير متعددة الأبعاد شديدة الفوضى والتقنيات القائمة على CNN في إطار شامل ، وتقييم أدائها من خلال تجارب مكثفة ومقاييس تقييم.

قدم النظام المقترح مكونا يختبر الإشارة شديدة الفوضى بناء على شبكة عصبية تلافيفية للتأكد من أنها فوضوية قبل إدخالها في طرق التشفير. كما تم تقديم طريقتين لتشفير الصور الرقمية: الطريقة الأولى باستخدام تقنية التجزئة والخلط الفوضوي، ويعتمد هذا النهج على تشفير الصور استنادا إلى خلط البتات بواسطة الاشارات ذات الفوضى المفرطة (IEHSB). يتم التشفير بالطريقة الثانية لتشفير الصور باستخدام ثلاثة مولدات لاشارات ذات الفوضى المفرطة (HS3IE)".

استخدمت الأطروحة اختبار القياس Lyapunov للتحقق من الإشارة الفوضوية المفرطة واختبار NIST للتحقق من عشوائية البتات المتولدة من الاشارة الفوضوية. أيضا، لتقييم التشفير وفك التشفير على الصور، تم استخدام متوسط خطأ التربيعي (MSE)، ونسبة إشارة الذروة إلى الضوضاء (PSNR)، ومعاملات الارتباط (Corr)، وقياس مؤشر التشابه الهيكلي (SSIM)، والانتروبيا.

تحقق نتائج المحاكاة التي أظهرها CNN–Tester دقة تقارب ١٠٠٪ في اختبار وتصنيف معلمات نظام رابينوفيتش. يبلغ التأخير في اختبار كل معلمة حوالي ١.٥٠٧٧ ثانية لمجموعة واحدة من المعلمات

في الاختبار التقليدي باستخدام (Lyapunov)، بينما يبلغ تأخير CNN–Tester حوالي ٠.٠١١٢ ثانية. يؤدي اختبار الإشارات الفوضوية باستخدام Tester–CNN إلى نتائج تصنيف أفضل وأسرع من الاختبار بالطريقة التقليدية. بالنسبة ل IEHSB، تم اختبار خمسة حالات في طريقة التشفير الاولى وهي: خلط بتات (البكسل، الصفوف، الاعمدة، المتجه الواحد للصورة ، واخيراً الصفوف ومن ثم الاعمدة) وكانت نتائج اخر حالتين هي الافضل من النتائج الاخرى ومساحة عدد المفاتيح في الحالة الاخيرة هو الافضل. وكذلك اضهرت نتائج التشفير بالطريقة الثانية نتائج افضل من جميع حالات الطريقة الاولى من حيث قوة التشفير على الصور وكذلك عدد المفاتيح.

توفر طريقة HS3IE أمانا وكفاءة أفضل، مما يجعلها خيارا واعدا لتشفير الصور. تبلغ مساحة المفتاح في IEHSB حوالي $٢^{٥٤٨}$ وحوالي $٢^{١٠٩٦}$ في حالة الصف والعمود ، بينما تبلغ مساحة المفتاح في طريقة HS3IE حوالي $٢^{١٦٤٤}$.

وزارة التعليم العالي والبحث العلمي
جامعة بابل كلية العلوم للبنات
قسم علوم الحاسوب

نظام لتشفير الصور الرقمية بالاعتماد على الأنظمة ذات الفوضى المفرطة متعددة الابعاد والشبكة العصبية التلافيفية

رسالة مقدمة الى مجلس كلية العلوم للبنات في جامعة بابل وهي جزء من
متطلبات نيل درجة الماجستير في كلية العلوم/ علوم الحاسبات
مقدمة من قبل

نور حيدر عبد علي وتوت

بإشراف

أ م د. علي يعكوب يوسف السلطان

١٤٤٤ هـ ٢٠٢٣ م