

Enhancing Image Security Through Memelements- Based Chaotic and Hyperchaotic Systems



Biniyam Ayele Belete

A Dissertation Submitted to
The Department of Electronics and Communication Engineering

College of Electrical Engineering and Computing

Presented in Fulfillment of the Requirement for the Degree of
Doctor of Philosophy in Electronics and Communication
Engineering (Specialization in Communication Engineering)

Office of Graduate Studies
Adama Science and Technology University

June, 2025
Adama, Ethiopia

Enhancing Image Security Through Memelements-Based Chaotic and Hyperchaotic Systems

Biniyam Ayele Belete

Supervisor: Dr. Demissie Jobir Gelmecha

Co-Supervisor: Dr. Ram Sewak Singh

A Dissertation Submitted to the Department of Electronics and
Communication Engineering
College of Electrical Engineering and Computing

Presented in Fulfillment of the Requirement for the Degree of Doctor
of Philosophy in Electronics and Communication Engineering
(Specialization in Communication Engineering)

Office of Graduate Studies
Adama Science and Technology University

June, 2025
Adama, Ethiopia

DECLARATION

I hereby declare that this Dissertation entitled “**Enhancing Image Security Through Memelements-Based Chaotic and Hyperchaotic Systems**” is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials used for this thesis have been duly acknowledged through appropriate citations.

Biniyam Ayele Belete
Name of the candidate

Signature

Date

RECOMMENDATION

We, the supervisors of this dissertation, hereby certify that we have read and revised the dissertation entitled “**Enhancing Image Security Through Memelements-Based Chaotic and Hyperchaotic Systems**” prepared under our guidance by **Biniyam Ayele Belete** submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in **Electronics and Communication Engineering (Specialization in Communication Engineering)**. Therefore, we recommend the submission of the dissertation to the department for further review and defense.

Dr. Demissie Jobir Gelmecha
Major Supervisor

Signature

Date

Dr. Ram Sewak Singh
Co-supervisor

Signature

Date

APPROVAL PAGE

We hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the dissertation entitled **“Enhancing Image Security Through Memelements-Based Chaotic and Hyperchaotic Systems”** by **Biniyam Ayele Belete**.

Dr. Demissie Jobir Gelmecha
Supervisor

Signature

Date

Dr. Ram Sewak Singh
Co-supervisor

Signature

Date

We, the undersigned, members of the Board of Examiners of the dissertation open defense by **Biniyam Ayele Belete** have read and evaluated the dissertation entitled **“Enhancing Image Security Through Memelements-Based Chaotic and Hyperchaotic Systems”** and examined the candidate during open defense. This is, therefore, to certify that the dissertation is accepted for partial fulfillment of the requirement of the degree of Doctor of Philosophy in **Electronics and Communication Engineering (Specialization in Communication Engineering)**

Chairperson

Signature

Date

Internal Examiner 1

Signature

Date

Internal Examiner 2

Signature

Date

External Examiner 1

Signature

Date

External Examiner 2

Signature

Date

Finally, approval and acceptance of the dissertation is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the candidate's Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head

Signature

Date

College Dean

Signature

Date

Office of Postgraduate Studies, Dean

Signature

Date

ACKNOWLEDGEMENTS

"No man is an island."

John Donne

First and foremost, I would like to express my deepest gratitude to God Almighty, whose grace, guidance, and unwavering presence have been my source of strength and inspiration throughout this journey. Without His blessings, this achievement would not have been possible.

I am profoundly indebted to my supervisors, Dr. Demissie Jobir Gelmecha and Dr. Ram Sewak Singh, for their invaluable guidance, patience, and encouragement. Their expertise, insightful feedback, and unwavering support have been instrumental in shaping this dissertation. I am truly honoured to have had the opportunity to learn from such esteemed mentors.

I sincerely thank the Ethiopian Air Force and Adama Science and Technology University for providing me with the resources, opportunities, and support necessary to pursue this academic endeavour. Your belief in my potential has been a driving force behind this achievement.

To my beloved family, words cannot express the depth of my gratitude. My wife, Meseret Gizaw, has been my pillar of strength, offering endless love, understanding, and encouragement during the most challenging times. My sons, Yonatan Biniyam and Etel Biniyam, have been my most significant source of joy and motivation. Your smiles and laughter have kept me going, even when the journey seemed overwhelming.

Finally, I extend my heartfelt thanks to all my friends, colleagues, and well-wishers who have supported me in countless ways throughout this journey. Your encouragement and kindness have meant the world to me.

This dissertation is a testament to the collective efforts, support, and love of all those mentioned above. To everyone who has been part of this incredible journey, I say:

"ጥቅ እሴብህ"

"በጣም አመሰግናለሁ!"

"Galatoomaa!"

"Thank you very much."

Biniyam Ayele Belete

TABLE OF CONTENTS

DECLARATION.....	I
RECOMMENDATION.....	II
APPROVAL PAGE.....	III
ACKNOWLEDGEMENTS.....	IV
TABLE OF CONTENTS.....	V
LIST OF FIGURES.....	XI
LIST OF TABLES.....	XVI
ACRONYMS.....	XIX
ABSTRACT.....	XXI
CHAPTER 1: INTRODUCTION.....	1
1.1 Motivation.....	2
1.2 Statement of the Problem.....	3
1.3 Research Questions.....	4
1.4 Objectives of the Study.....	4
1.4.1 General Objective.....	4
1.4.2 Specific Objectives.....	5
1.5 Significance of the Study.....	5
1.6 Scope of the Study.....	6
1.7 Framework of the Proposed Study.....	6
1.8 Contributions of the Dissertation.....	7
1.9 Structure of the Dissertation.....	9
CHAPTER 2: LITERATURE REVIEW.....	11
2.1 The literature review methodology.....	11
2.2 Protecting Visual Data in Smart and Connected Environments.....	12
2.2.1 Communication Security.....	13
2.2.2 Image Security.....	13
2.2.3 Memelements: A Paradigm Shift.....	15
2.2.4 Chaotic and Hyperchaotic Systems.....	17
2.2.5 DNA Coding.....	20
2.2.6 Deep Learning and Autoencoders.....	22

2.2.7	Integration of Advanced Techniques in Image Encryption	26
2.2.8	The Role of AI and Hybrid Architectures.....	26
2.3	Comparative Analysis of Traditional and Chaos-Based Encryption Methods	27
2.3.1	Traditional Image Encryption Methods	27
2.3.2	Symmetric Encryption Algorithms	28
2.3.3	Asymmetric Encryption Algorithms.....	30
2.4	Chaos-Based Encryption	32
2.4.1	Categorization of Chaotic Systems for Image Security	32
2.4.2	Memelements in Hyperchaotic Cryptography	33
2.4.3	Generation of Memelements-Based Hyperchaotic Systems.....	33
2.4.4	Image Encryption Using Memelements-Based Hyperchaotic Systems.....	33
2.5	DNA-Based Encryption	35
2.6	Deep Learning-Driven Security Enhancements	35
2.7	Security Analysis Criteria in Image Encryption	36
2.8	Related Work and Identified Research Gaps	37
2.9	Gaps in Current Image Encryption Research.....	41
2.10	Final Summary of the Literature Review	42
CHAPTER 3: MEMELEMENTS AND THE DEVELOPMENT OF CHAOTIC SYSTEMS.....		43
3.1.	Introduction:	43
3.2.	Fundamentals of Memelements	43
3.2.1.	Memcapacitor Model	44
3.2.2.	Meminductor Model	44
3.3.	Integer-Order Hyperchaotic Systems with Memelements	45
3.3.1.	Memristor-Based Hyperchaotic System	46
3.3.2.	Dual Memristor-Based System	49
3.3.3.	Dual Memristor-Band Pass Filter System.....	51
3.3.4.	Memcapacitor-Based System.....	54
3.3.5.	Hybrid Memcapacitor-Meminductor System	58
3.3.6.	Memcapacitor-Meminductor Enhanced System.....	61

3.4.	Fractional-Order Hyperchaotic Systems with Memelements	64
3.4.1.	Dissipation and Stability of the System	69
3.5.	Optimization of Hyperchaotic systems	72
3.5.1.	Optimizing the Hyperchaotic System Using CPSO.....	75
3.6.	Classical One-Dimensional Chaotic Maps and Hybrid One-Dimensional Maps	78
3.6.1.	Logistic Map	78
3.6.2.	Tent Map.....	79
3.6.3.	Sine Map	79
3.6.4.	Cubic Map.....	80
3.6.5.	General Limitations of 1D Chaotic Maps.....	80
3.6.6.	Logistic-Sine Adjusted Integrated Map	80
3.6.7.	Logistic-Sine-Cubic Iterative Map	82
3.6.8.	Sine-Logistic Integrated Map	86
3.7.	Summary	87
3.8.	Materials.....	88
3.8.1.	Datasets	88
3.8.2.	Software and Hardware Setup	88
3.8.3.	Circuit Simulation.....	88

CHAPTER 4: IMAGE ENCRYPTION USING A MEMELEMMENT-BASED

HYPERCHAOTIC SYSTEM INTEGRATED WITH DNA ENCODING SCHEMES .89

4.1	Introduction	89
4.2	Proposed Image Encryption Algorithm.....	89
4.2.1	The Encryption Algorithm's Specific Steps	90
4.2.2	The Decryption Algorithm's Specific Steps	93
4.3	Result and Discussion	96
4.3.1	Histogram Analysis.....	96
4.3.2	Correlation Analysis	96
4.3.3	Information Entropy Analysis.....	101
4.3.4	Keyspace Analysis	103

4.3.5	Key Sensitivity Analysis.....	103
4.3.6	Mean Squared Error.....	104
4.3.7	Mean Absolute Error.....	105
4.3.8	Differential Attack Analysis	107
4.3.9	Running Speed Analysis	108
4.4	Randomness Evaluation Using the NIST Statistical Test Suite.....	109
4.5	Comparative Evaluation of Security and Performance Metrics.....	110
4.6	Summary	110
CHAPTER 5: ENHANCING IMAGE ENCRYPTION THROUGH PARAMETERS		
OPTIMIZATION OF MEMRISTIVE HYPERCHAOTIC SYSTEM WITH CPSO		
ALGORITHM AND LSAIM		111
5.1	Introduction	111
5.2	Proposed Encryption Algorithm Using an Optimized Hyperchaotic System	111
5.3	Results and Discussion.....	115
5.3.1	Analysis of Performance and Security.....	116
5.3.2	Statistical Analysis.....	117
5.3.3	Analysis of Information Entropy	119
5.3.4	Differential Attack Analysis	120
5.3.5	Mean Square Error.....	121
5.3.6	Peak Signal-to-Noise Ratio.....	121
5.3.7	Mean Absolute Error.....	122
5.3.8	Structural Similarity Index Measure.....	122
5.3.9	Computational time analysis of the proposed algorithm	123
5.4	Evaluation of Randomness Using the NIST Statistical Test Suite.....	124
5.5	Performance Validation and Comparative Security Analysis.....	125
5.6	Summary	125
CHAPTER 6: DENOISING ENCRYPTED IMAGES USING ONLINE SEQUENTIAL		
EXTREME LEARNING MACHINE WITH A MEMELEMNTS-BASED		
HYPERCHAOTIC SYSTEM.....		127
6.1	Introduction	127
6.1.1	Online Sequential Extreme Learning Machine.....	127

6.1.2	OSELM Auto-Encoder	129
6.1.3	Major Image Noises	130
6.2	Proposed Methodology	130
6.2.1	Proposed Image Encryption and Anti-clipping Algorithm	131
6.2.2	The Encryption and Anti-clipping Algorithm's Specific Steps	131
6.2.3	The decryption and Denoising algorithms	134
6.2.4	The Proposed Image Decryption and Anti-clipping Algorithm's Steps	134
6.2.5	Denoising using OSELM Auto-encoder	136
6.3	Result and Discussion	137
6.3.1	Simulation Results	137
6.3.2	Image Security Analysis	139
6.3.3	Image Denoising Analysis	146
6.4	Randomness Evaluation Using NIST Statistical Test Suite	153
6.5	Performance Validation and Comparative Security Analysis	154
6.6	Summary	155
CHAPTER 7: AN ADAPTIVE DEEP CNN-BASED COMPRESSION AND HYBRID		
INTEGER-FRACTIONAL ORDER HYPERCHAOTIC ENCRYPTION		
FRAMEWORK FOR SECURE IMAGE TRANSMISSION		156
7.1	Introduction	156
7.1.1	Importance Map-Based Adaptive Compression	156
7.2	Methodology for Image Compression-Encryption Algorithm	160
7.2.1	Proposed Image Compression Algorithm	161
7.2.2	Proposed Image Encryption Algorithm	162
7.3	Result and Discussion	165
7.3.1	Image Compression and Decompression Efficiency Evaluation Metrics	168
7.3.2	Mathematical Analysis and Performance Evaluation of Image Compression, Encryption, and Decompression Using Chaotic and Hyperchaotic Systems	171
7.3.3	Keyspace analysis	182
7.3.4	Key sensitivity test	182
7.4	Evaluating Randomness with the NIST Statistical Test Suite	183

7.5	Performance Validation and Comparative Security Analysis	183
7.6	Summary	185
CHAPTER 8: CONCLUSIONS AND RECOMMENDATIONS.....		186
8.1	Conclusions	186
8.2	Recommendations	189
RESEARCH CONTRIBUTION		191
REFERENCES.....		192
APPENDIX A: MATHEMATICAL MODELS.....		217

LIST OF FIGURES

Figure 1.1: Flow Diagram of the Proposed Work.....	7
Figure 2.1: Literature Review Methodology	11
Figure 2.2: Areas of Application for Current Image Encryption.....	12
Figure 2.3: Block Diagram Illustrating the Process of Image Encryption and Decryption.	14
Figure 2.4: Overview of Image Security Techniques.	15
Figure 2.5: Basic Circuit Elements Exhibiting In-Memory and Nonlinear Responses.	16
Figure 2.6: The Four Fundamental Two-terminal Circuit Elements (Chua, 2018).	16
Figure 2.7: System of Basic Passive Memory-less and Mem-elements (Petráš, 2020, 2022).	17
Figure 2.8: Integration of Chaotic Systems and Cryptography.	20
Figure 2.9: The Structure of Part of A DNA Double Helix (Alruily et al., 2021).....	21
Figure 2.10: Architectures of CNN and Autoencoders for Image Compression and Denoising in Encryption.....	23
Figure 2.11: A Condensed Deep Neural Network (Meiliana et al., 2019)	23
Figure 2.12: Schematic of a Convolutional Neural Network (Kermany et al., 2018).	24
Figure 2.13: The Architecture of a Sample Autoencoder (Torabi et al., 2023).	25
Figure 2.14: (a) RBM Model (b) Autoencoder Model (Gupta & Vijay, 2022).	26
Figure 2.15: The Role of AI and Hybrid Architectures in Secure Image Encryption	27
Figure 2.16: Deep Learning-Driven Image Security Pipeline	36
Figure 3.1: Schematic of the Electrical Element Relations (Adapted Ref. (Petráš, 2022)).....	44
Figure 3.2: Hysteresis Loops for Memelements: (a) Memristor $q - u$ Hysteresis, (b) Memcapacitor $q - u$ Hysteresis, and (c) Meminductor $\phi - i$ Hysteresis.	45
Figure 3.3: The Memristor-Based Hyperchaotic System Circuit	47
Figure 3.4: Phase Portraits of the Hyperchaotic Attractors.	48
Figure 3.5: Bifurcation Diagrams of Hyperchaotic System for γ (8.0 to 12.0), β (0.75 to 1.25).	48
Figure 3.6: Phase Portraits of the Hyperchaotic Attractors.	48
Figure 3.7: Dual Memristor-Based Hyperchaotic Circuit.....	49
Figure 3.8: Phase Portraits of the Hyperchaotic Attractors: (a) x-y plane, (b) y-z plane, (c) z- w plane, and (d) w-x plane.....	50
Figure 3.9: Diagram Showing the Bifurcation of a Hyperchaotic System (for parameter " α ")	50
Figure 3.10: The Hyperchaotic Attractors' Phase Portraits.....	50
Figure 3.11: Memristive Chua's Hyperchaotic Circuit.....	51

Figure 3.12: 2D and 3D Phase Portrait of the Hyperchaotic System (5).....	53
Figure 3.13: Lyapunov Exponent Diagrams of the Hyperchaotic System (3.12) for Running times $T=18, 60$, and 10000 seconds.....	53
Figure 3.14: Bifurcation Diagrams, (a) $a \in 19.5, 21$, (b) $d \in 0.13, 0.20$, (c) $m_2 \in 0, 0.05$	54
Figure 3.15: A Hyperchaotic Oscillator Based on Memcapacitors.	54
Figure 3.16: The Phase Portraits of the Proposed Memcapacitor-Based Hyperchaotic System onto Different Spaces (a) x - y , (b) x - w , (c) y - w , and (d) z - w	56
Figure 3.17: Lyapunov Exponents of the Memcapacitive Hyperchaotic System.....	56
Figure 3.18: Bifurcation Plot of the Memcapacitive Hyperchaotic System for Parameter b ..	57
Figure 3.19: Bifurcation Plot of the Memcapacitive Hyperchaotic System for Parameter d ..	57
Figure 3.20: Chaotic Oscillator Circuit Using Memcapacitor and Meminductor	58
Figure 3.21: System Behavior: (a) Waveforms of State Variables (x, y, z) and (b) Chaotic Attractors of the System.....	59
Figure 3.22: Lyapunov Exponents (LE1, LE2, LE3, LE4, LE5) of the System.....	60
Figure 3.23: Bifurcation of the Memelements-Based Hyperchaotic system with Forward and Backward Parameter (a) Variation.....	60
Figure 3.24: The Hyperchaotic Oscillator Circuit of the Proposed Memcapacitor and Meminductor.....	62
Figure 3.25: 2D and 3D Phase Portraits of IOMBHCS.	63
Figure 3.26: Lyapunov Exponents of IOMBHCS for Different Operation Times.	63
Figure 3.27: Forward and Reverse Bifurcation Diagram of the IOMBHCS Concerning Parameter d	64
Figure 3.28: FOMBHCS 2D Phase Portrait of FOMBHCS x vs y for Different q Values	68
Figure 3.29: Lyapunov Exponents and Dimension of FOMBHCS: (a) Lyapunov Exponents vs Time for $q = 0.505$, (b) Lyapunov Exponents vs Fractional Order (q), (c) Lyapunov Dimension vs Fractional Order (q).	69
Figure 3.30: Forward and Reverse Bifurcation Diagram of the FOMBHCS for Parameter d ..	69
Figure 3.31: The Memristor-Based Hyperchaotic System Circuit	73
Figure 3.32: Phase Portraits of the Hyperchaotic Attractors.	73
Figure 3.33: Bifurcation Diagrams of Hyperchaotic System for γ (8.0 to 12.0), β (0.75 to 1.25).	74
Figure 3.34: Phase Portraits of the Hyperchaotic Attractors.	74
Figure 3.35: Hyperchaos Degree (h) Over Iterations.....	77
Figure 3.36: Optimized Lyapunov Exponent's Dynamics	78

Figure 3.37: Bifurcation Diagram of the Logistic Map	78
Figure 3.38: Logistic-Sine Integrated Map Bifurcation Diagrams for Different r_2 Values....	81
Figure 3.39: Lyapunov Exponent of LSIM for Different r_2 Values	82
Figure 3.40: 1D Phase Portraits of a Dynamical System for Varying Parameter r_2	83
Figure 3.41: Lyapunov Exponent Diagram of LSCIM System with Varying r_1 and Fixed r_2 Values	84
Figure 3.42: Bifurcation Diagrams of the Dynamical System for Varying r_2 and r_1	85
Figure 3.43: Logistic-Sine Integrated Map Bifurcation Diagrams for Different r_2 Values....	87
Figure 3.44: Lyapunov Exponent of LSIM for Different r_2 Values	87
Figure 4.1: Flow of the Proposed Encryption Algorithm	92
Figure 4.2: Flow of the Proposed Decryption Algorithm	95
Figure 4.3: (a.) Plane Images (b.) Encrypted Images (c.) Decrypted Images.....	97
Figure 4.4: (a) Plane Images and Plane Images <i>RGB</i> Histogram (b) Encrypted Images and Encrypted Images <i>RGB</i> Histogram.....	98
Figure 4.5: Lena's Original Test Image's Horizontal, Vertical, and Diagonal Correlations	99
Figure 4.6: Lena's Encrypted Test Image's Horizontal, Vertical, and Diagonal Correlations	100
Figure 4.7: Encryption-Decryption Results with and without Slight Key Changes;	104
Figure 5.1: Diagram Illustrating the Suggested <i>RGB</i> Image Encryption	112
Figure 5.2: Flow Diagram for the Suggested Decryption of <i>RGB</i> Images.....	114
Figure 5.3: Original, Encrypted, and Decrypted Image Results with Their Histograms.....	115
Figure 5.4: Results of an Imaging Test for Benign Tumor, Both Encrypted and Decrypted	117
Figure 5.5: Pepper's Encrypted Image and Plaintext Image Correlation Distributions	117
Figure 6.1: OSELM algorithm structure (Ou et al., 2021).....	129
Figure 6.2: Block diagram of image denoising Autoencoder	130
Figure 6.3: Encryption and Denoising Algorithm Blocks	130
Figure 6.4: The Proposed Encryption, Decryption, and Denoising Algorithm Flow Diagram	132
Figure 6.5: Original Input Images. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper	137
Figure 6.6: Encrypted Images. (a) Encrypted Image of Splash, (b) Encrypted Image of Baboon, (c) Encrypted Image of Airplane, (d) Encrypted Image of Sailboat, (e) Encrypted Image of Pepper	137

Figure 6.7: Encrypted Images Corrupted by Gaussian Noise with a Variance of $\sigma^2 = 0.5$. (a) Encrypted-Noised of Splash, (b) Encrypted-Noised of Baboon, (c) Encrypted-Noised of Airplane, (d) Encrypted-Noised of Sailboat, (e) Encrypted-Noised of Pepper	138
Figure 6.8: Decrypted Images Corrupted by Gaussian Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Noised of Splash, (b) Decrypted-Noised of Baboon, (c) Decrypted-Noised of Airplane, (d) Decrypted-Noised of Sailboat, (e) Decrypted-Noised of Pepper	138
Figure 6.9: Decrypted Images Corrupted by Sensor Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Noised of Splash, (b) Decrypted-Noised of Baboon, (c) Decrypted-Noised of Airplane, (d) Decrypted-Noised of Sailboat, (e) Decrypted-Noised of Pepper	138
Figure 6.10: Decrypted Images Corrupted by Environmental Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Noised of Splash, (b) Decrypted-Noised of Baboon, (c) Decrypted-Noised of Airplane, (d) Decrypted-Noised of Sailboat, (e) Decrypted-Noised of Pepper.....	138
Figure 6.11: Decrypted-Denoised from Gaussian Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Denoised of Splash, (b) Decrypted-Denoised of Baboon, (c) Decrypted-Denoised of Airplane, (d) Decrypted-Denoised of Sailboat, (e) Decrypted-Denoised of Pepper	138
Figure 6.12: Encryption-Decryption Results with and without Slight Key Changes;	139
Figure 6.13: Baboon's Original Test Image's Horizontal, Vertical, and Diagonal Correlations	141
Figure 6.14: Baboon's Encrypted Test Image's Horizontal, Vertical, and Diagonal Correlations.....	142
Figure 6.15: Original Input Images. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper	143
Figure 6.16: Plane Images RGB Histogram. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper	143
Figure 6.17: Encrypted Images. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper	143
Figure 6.18: Encrypted Images RGB Histogram. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper	143
Figure 6.19: Decrypted Images with Gaussian Noise Attacks: (a) 0.01 Noise Level, (b) 0.02 Noise Level, (c) 0.03 Noise Level, (d) 0.04 Noise Level.	147
Figure 6.20: Decrypted Images with Salt-and-Pepper Noise Attacks: (a) 0.01 Noise Level, (b) 0.04 Noise Level, (c) 0.1 Noise Level, (d) 0.14 Noise Level.	147
Figure 6.21: Decrypted Images with Speckle Noise Attacks: (a) 0.01 Noise Level, (b) 0.03 Noise Level, (c) 0.05 Noise Level, (d) 0.07 Noise Level.	147

Figure 6.22: Decrypted Noised Images	148
Figure 6.23: Denoised Images	150
Figure 7.1: Flow diagram of the MNV2 Architecture	157
Figure 7.2: The Flow of MNV2 Based Compression Architecture	159
Figure 7.3: Secure Image Transmission Using Adaptive Compression and Fractional-order Hyperchaotic Encryption	161
Figure 7.4: Flowchart of CNN-Based Importance Map for Adaptive Image Compression ..	162
Figure 7.5: Encryption Process for Adaptively Compressed Color Images Using Fractional-order hyperchaotic and Chaotic Systems	166
Figure 7.6: Stages of Image Processing: Adaptive Compression and Compressed Sensing with Encryption-Decryption	167
Figure 7.7: The Original Input Pepper Test Image's Horizontal-Vertical-Diagonal Correlations.....	173
Figure 7.8: Pepper's Ciphered Image Horizontal-Vertical-Diagonal Correlations	174
Figure 7.9: Input original images. (a) Crown splash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract	175
Figure 7.10: Input original images RGB Histograms. (a) Crownsplash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract.....	175
Figure 7.11: Ciphered images. (a) Crown-Splash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract.....	175
Figure 7.12: Ciphered images RGB Histograms. (a) Crown-Splash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract	175
Figure 7.13: Decryption Results Using Appropriate Key and Slightly Modified Key;.....	182

LIST OF TABLES

Table 2.1: DNA Representation of Bits	22
Table 2.2: Basic DNA Coding Operations Used in Image Encryption Schemes	22
Table 2.3: Performance Metrics of Traditional Encryption Algorithms.....	28
Table 2.4: Comparison of Different Symmetric-Key Encryption Algorithms	29
Table 2.5: Comparison of Different Asymmetric-Key Cryptography Algorithms.....	30
Table 2.6: Comparison of Chaotic Image Encryption and Traditional Methods.....	31
Table 2.7: Categorization of Chaotic Systems.....	32
Table 2.8: Security Performance Metrics for Memelements Image Encryption	34
Table 2.9: Tabulated Review of Memelements-Based Image Encryption	35
Table 2.10: Encryption Parameters for Lena Original and Encrypted Images	37
Table 2.11: Security Analysis Metrics.....	37
Table 2.12: Related Literature Review Summaries	38
Table 2.13: Summarized Literature Reviews on Advanced Image Encryption Techniques ...	40
Table 3.1: A Comparison Between the Original and Optimized Hyperchaotic System.....	77
Table 4.1: Original Test Images and Encrypted Test Images Correlation Coefficients.	100
Table 4.2: Correlation Results between Encrypted Images with Other Techniques	101
Table 4.3: Information Entropy Comparison Among Original and Cyphered Images.....	102
Table 4.4: Entropy Results Comparison of Lena Test Image with Other Techniques	102
Table 4.5: Comparative of Keyspace.....	103
Table 4.6: PSNR and SSIM Values for Original and Reconstructed Images.....	105
Table 4.7: MSE, PSNR, and MAE Values of Original and Encrypted Images.....	106
Table 4.8: Comparative MSE, PSNR, and MAE Values of Lina Image RGB Channels.....	106
Table 4.9: The UACI and NPCR Results of Test Images.....	107
Table 4.10: The UACI and NPCR Results of Test Images.....	108
Table 4.11: The Sample Times for Encryption and Decryption	109
Table 4.12: NIST STS Results for the Proposed Encrypted Image Bit Stream.....	109
Table 4.13: Performance Summary of the Proposed Image Encryption Scheme.....	110
Table 5.1: An Analysis of Key Spaces Comparison.....	116
Table 5.2: Pixel Correlation Values for Test Images that are Encrypted and the Source.....	118
Table 5.3: Information Entropy Comparison Between Cyphered and Original Images.....	119
Table 5.4: Comparing Pepper Test Image Entropy Results with Alternative Methods.....	119
Table 5.5: Comparative Values of UACI and NPCR for Cyphered Image	120
Table 5.6: MSE and PSNR Between Original and Encrypted Test Images	122

Table 5.7: MAE and SSIM of Original and Encrypted Test Images	123
Table 5.8: MSE, PSNR, and SSIM of Decrypted Images	123
Table 5.9: Computational Time for Six Iterations and Average.....	124
Table 5.10: NIST STS Results for the Encrypted Image Bit Stream.....	124
Table 5.11: Summary of Performance Evaluation Metrics	125
Table 6.1: Pseudo-code of the Encryption and Anti-clipping Steps.....	136
Table 6.2: Comparative of Keyspace.....	139
Table 6.3: Entropy of Original and Cyphered Images	140
Table 6.4: Comparative Entropy Results of Test Images Using Different Techniques.....	140
Table 6.5: Correlation Coefficients Between Original and Encrypted Version	141
Table 6.6: Correlation Coefficients for Encrypted Images Using Different Techniques	142
Table 6.7: The NPCR and UACI Results of Input Images	144
Table 6.8: NPCR and UACI Results for Ciphred Baboon Test Image	144
Table 6.9: MSE and PSNR Values of Original and Encrypted Images without Noise	145
Table 6.10: Comparative MSE and PSNR Values of RGB Channels	146
Table 6.11: PSNR and SSIM Values for Original and Reconstructed Images.....	146
Table 6.12: Assessment of Gaussian Noise Qualification in Decrypted Images.....	147
Table 6.13: Assessment of Salt-and-Pepper Noise in Decrypted Images.....	149
Table 6.14: Assessment of Speckle Noise Qualification in Decrypted Images.....	149
Table 6.15: Performance Metrics for Noisy and Denoised Images	151
Table 6.16: The Encryption and Decryption Running Speed of Various Test Images.....	152
Table 6.17: Comparison of the Encryption Time With Other Schemes	153
Table 6.18: NIST STS Results for the Encrypted Image Bit Stream.....	153
Table 6.19: Performance Evaluation Summary of the Proposed Encryption Scheme	154
Table 7.1: The Proposed Adaptive Compression Pseudocode and Compressed Sensing	163
Table 7.2: Metrics Comparison after Encryption, Decryption, and Decompression.....	170
Table 7.3: Compression Time and Ratio for Various CNN Models and Images	172
Table 7.4: Performance Metrics of Adaptive Encryption Using CNN Models.....	174
Table 7.5: Adaptive CS Encryption Results with CNN Models Across Datasets	176
Table 7.6: Decryption Results of Adaptive Compression with CNN Models.....	177
Table 7.7: The Results Discussion of UACI and NPCR for Encrypted Image	178
Table 7.8: MSE and PSNR Comparative Values of Input and Encrypted Emage	178
Table 7.9: The Proposed Adaptive Compression and Compressed Sensing	179
Table 7.10: Image Decryption Results of Adaptively CS Decrypted Image Images	180

Table 7.11: Decryption Results of Adaptively CS Decompressed and Decrypted Images ...	181
Table 7.12: Keyspace Comparison with Current Research Works.....	182
Table 7.13: Proposed Framework for NIST STS Results on the Encrypted Image Bitstream	183
Table 7.14: Performance Evaluation Summary of the Proposed Encryption Scheme	184
Table 8.1: Proposed methods comparison	188

ACRONYMS

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
4D	Four-Dimensional
5D	Five-Dimensional
A	Adenine
AE	Autoencoder
AES	Advanced Encryption Standard
ADM	Additive Decomposition Method
AI	Artificial Intelligence
ANN	Artificial Neural Network
BPF	Band Pass Filter
C	Cytosine
CNN	Convolutional Neural Network
CPSO	Chaotic Particle Swarm Optimization
CR	Compression Ratio
CS	Compressed Sensing
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
DL	Deep Learning
DNA	Deoxyribonucleic Acid
DWT	Discrete Wavelet Transform
ECC	Elliptic Curve Cryptography
EE	Electrical Engineering (assumed context)
ELM	Extreme Learning Machine
FFT	Fast Fourier Transform
FOMBHCS	Fractional-Order Memelements-Based Hyperchaotic System
FPGA	Field-Programmable Gate Array
GA	Genetic Algorithm
G	Guanine
GF	Galois Field
HMCO	Hyperchaotic Memcapacitive Oscillator
IDEA	International Data Encryption Algorithm

IEEE	Institute of Electrical and Electronics Engineers
IOMBHCS	Integer-Order Memelements-Based Hyperchaotic System
IoT	Internet of Things
LEA	Lightweight Encryption Algorithm
LEs	Lyapunov Exponents
LSAIM	Logistic Sine-Adjusted Integrated Map
LSCIM	Logistic-Sine-Cubic Iterative Map
MAE	Mean Absolute Error
MNV2	MobileNetV2
MSE	Mean Squared Error
NIST	National Institute of Standards & Technology Special Publication
NPCR	Number of Pixels Change Rate
OSELM	Online Sequential Extreme Learning Machine
PRNG	Pseudo-Random Number Generator
PSNR	Peak Signal-to-Noise Ratio
PSO	Particle Swarm Optimization
R	Red
RC4	Rivest Cipher 4
RDNN	Recurrent Deep Neural Network
RSA	Rivest–Shamir–Adleman
SAE	Stacked Autoencoder
SLIM	Sine-Logistic Integrated Map
SSIM	Structural Similarity Index Measure
T	Thymine
UACI	Unified Average Changing Intensity
USC-SIPI	University of Southern California Signal & Image Processing Institute
VGG16	Visual Geometry Group (16-layer CNN model)
VGG19	Visual Geometry Group (19-layer CNN model)
WSN	Wireless Sensor Network
XOR	Exclusive OR
5G	5th Generation Mobile Network
6G	6th Generation Mobile Network

ABSTRACT

In today's digitally interconnected world, the volume and sensitivity of visual data continue to grow rapidly, highlighting the critical importance of secure image transmission, particularly in cloud computing, surveillance, healthcare, and military systems. Conventional encryption algorithms often fail to account for the high correlation between image pixels and the large size of image data, making them vulnerable to modern attacks. To overcome these limitations, this study presents several novel image encryption frameworks that incorporate hyperchaotic systems with memristive, memcapacitive, and meminductive devices, DNA-based encoding, and intelligent optimization techniques. First, a new encryption approach integrates DNA coding with a logistic map and a hyperchaotic memcapacitor-based model, achieving high NPCR values (99.587%–99.634%) and UACI values (27.474%–33.935%), as well as entropy values close to 8, indicating strong randomness and statistical resistance. The key space of 2^{531} surpasses brute-force thresholds threshold of 2^{100} . Recognizing the limitations of traditional Particle Swarm Optimization (PSO), the study introduces Chaotic PSO (CPSO) to fine-tune parameters of a four-dimensional memristor-based hyperchaotic system. This method, combined with DNA coding and a Logistic Sine Adjusted Integrated Map (LSAIM), achieves enhanced image encryption performance, including a key space of 2^{1116} , MSE of 10676.01, and PSNR of 7.85 dB. To address noise interference in transmission, an integrated encryption-denoising scheme is proposed using an Online Sequential Extreme Learning Machine (OSELN) autoencoder paired with a two-memristor-based hyperchaotic system, a 2D sine map and DNA coding. This approach shows resilience against multiple noise types (Gaussian, salt-and-pepper, quantization, speckle, etc.), achieving PSNR between 23.63 and 37.45 dB for denoised images, and maintaining strong security metrics (NPCR up to 99.64%, UACI $\sim$ 33.92%, entropy $\approx$ 8, key space $\approx 2^{797}$). Finally, a dual-stage architecture is proposed for image compression and encryption. It combines MobileNetV2-based CNN compression with a memelement-driven fractional-order hyperchaotic encryption scheme enhanced by a Logistic-Sine-Cubic Iterative Map (LSCIM), DNA coding, and dynamic diffusion. This method maintains near-lossless compression quality (PSNR $>$ 55 dB, SSIM $\approx$ 0.9999, MSE $\approx$ 0.2) while achieving a key space of 2^{1276} , ensuring high robustness against brute-force and statistical attacks. Together, these contributions form a comprehensive solution for secure, efficient, and noise-resilient image processing, making them highly applicable in sensitive domains such as medicine, defense, astronomy, and biological imaging.

Keywords: Hyperchaotic Systems, Memristor-based Circuits, Fractional-Order Dynamics, CNN-based Compression, Secure Multimedia Communication, Permutation-Diffusion Architecture, Compressed Sensing, Autoencoders.

CHAPTER 1

INTRODUCTION

In the modern digital era, the rapid growth of multimedia technologies has significantly increased the generation and storage of visual data. Images are crucial in healthcare, cloud storage, surveillance, social media, and military communications. As reliance on digital images grows, so does the need to protect their integrity and confidentiality. With increasing cyber threats, securing visual information is essential for maintaining data privacy and resilience in interconnected systems.

Image encryption is an effective way to secure image data from unauthorized access. However, traditional text or binary data encryption often faces challenges when applied to images due to their unique characteristics, such as high redundancy, strong inter-pixel correlation, and large file sizes. These challenges can lead to inefficiencies, making encrypted images vulnerable to various attacks, including brute-force, statistical, and differential attacks (Hadj Brahim et al., 2024; Malik et al., 2023; Rashid & Hussein, 2023). Additionally, many existing techniques do not consider other important factors related to image transmission, such as compression efficiency and resilience to channel noise (Charmouti et al., 2022; J. Chen et al., 2020; Rashid & Hussein, 2023; Wen et al., 2022).

This research systematically addressed challenges by designing a unified framework for secured image transmission, integrating memelements-based chaotic systems, hybrid one-dimensional maps, deep learning, and biological computing models. The central objective of this dissertation is to design novel, high-performance image encryption algorithms that are computationally efficient, highly secure, and resilient to noise while incorporating image compression for efficient transmission and storage and as a preprocessing stage for practical deployment.

The study introduces five new integer-order hyperchaotic systems that utilize memelements, specifically, memristors, memcapacitors, and meminductors. These nonlinear devices, enable complex dynamical behaviour and high-dimensional chaos, essential for building secure cryptographic primitives.

Fractional-order derivatives were introduced to enhance the chaotic dynamics further, providing more precise control over system memory and nonlinearity. Additionally, Chaotic Particle Swarm Optimization (CPSO) was used to optimize the parameters of the proposed systems. CPSO demonstrated superior performance to standard Particle Swarm Optimization

(PSO) and Genetic Algorithms (GA), significantly increasing the unpredictability necessary for secure encryption.

In addition to system design, the study introduces and assesses two advanced hybrid chaotic maps: the Logistic-Sine Adjusted Integrated Map (LSAIM) and the Logistic-Sine-Cubic Iterative Map (LSCIM). These maps utilize permutation techniques for row and column shifts. By integrating multiple sources of nonlinearity, they enhance chaotic behaviour and increase resistance against known attacks.

A dual-stage adaptive compression-encryption framework was developed to support the practical transmission of encrypted images. This framework uses Convolutional Neural Networks (CNNs) such as VGG19 and MobileNetV2 to adaptively compress images by distinguishing non-critical and essential regions. This pre-encryption compression reduces data size while preserving crucial visual content. The compressed images are then encrypted using multilayered chaotic methods.

Moreover, recognizing the noise challenge in real-world transmission channels, a noise-resilient encryption architecture was proposed by integrating an Online Sequential Extreme Learning Machine (OSELM) autoencoder with chaotic encryption and DNA coding. This hybrid scheme improves both noise tolerance and encryption security.

1.1 Motivation

The motivation for this research stems from the convergence of three urgent needs in modern image communication systems:

- As more sensitive and mission-critical images are transmitted over insecure networks, the need for robust, lightweight encryption mechanisms is growing.
- Large image sizes can burden storage and transmission systems, particularly in resource-constrained environments such as mobile networks or satellite communications.
- In practical scenarios, images often encounter noise, loss, or distortion, compromising their quality and security.

Despite extensive research in image encryption, few methods simultaneously address all three needs in a unified and adaptive framework. This dissertation was driven by the aim of filling that gap. Using memelements-based chaotic systems and DNA coding enhances complexity and sensitivity, making the encryption stronger and more challenging to predict. Integrating

deep learning-based compression offers intelligent preprocessing, which is vital for scalability. Including noise-resilient mechanisms using OSELM ensures practical viability in real-world communication systems.

Together, these contributions not only fulfil the research objectives but also open new directions for designing adaptive, intelligent, and secure multimedia transmission systems, especially in sensitive applications such as healthcare diagnostics, surveillance networks, military communications, and remote data acquisition platforms.

1.2 Statement of the Problem

In an era of exponential data generation, particularly through visual content enabled by 5G, IoT, healthcare, and defense applications, secure image transmission has become a critical concern. Conventional cryptographic algorithms, such as AES, RSA, and DES, although robust for text or general-purpose data, exhibit significant shortcomings in handling image data due to inherent properties like high data redundancy, pixel correlation, and structural patterns. Even classical chaos-based schemes, while offering unpredictability and key sensitivity, often suffer from limited key-space, inadequate robustness against noise, slow encryption speed, or incompatibility with modern resource-constrained environments [1]–[9].

Furthermore, existing image encryption methods largely neglect adaptive intelligence and fail to integrate with compression, noise resilience, and dynamic complexity, all of which are critical for real-time secure transmission. In addition, contemporary threats such as statistical attacks, differential cryptanalysis, and brute-force techniques necessitate the design of multi-layered, lightweight, and highly secured encryption frameworks [5], [10]–[15].

Given these challenges, this research addresses a multidimensional problem: designing an intelligent, robust, and computationally efficient image encryption system that effectively integrates hyperchaotic dynamics, bio-inspired techniques (e.g., DNA coding), and artificial intelligence mechanisms (e.g., CNNs and OSELM). The aim is to outperform traditional and chaos-only models in terms of entropy, attack resistance, key sensitivity, and processing speed. Addressing this gap is crucial for developing a unified, scalable framework that can effectively handle real-world constraints, including image noise, high compression requirements, and evolving attack strategies.

Despite recent advancements in cryptography, the field of image encryption continues to grapple with limitations in adaptability, complexity, integration, and real-time performance. This study addresses these issues by proposing innovative image encryption frameworks that

incorporate memelement-based high-dimensional and fractional-order chaotic systems, hybrid one dimensional chaotic maps, DNA-based coding, and deep learning models. These solutions are designed to provide secure, efficient, and resilient image encryption, suitable for critical domains such as military surveillance, medical imaging, and next-generation communication systems.

1.3 Research Questions

- How can novel chaotic and hyperchaotic systems that incorporate memristors, memcapacitors, and meminductors be mathematically modelled and analysed to enhance the complexity and unpredictability of image encryption?
- What encryption framework can be designed using a mem-element-based hyperchaotic system and DNA coding to ensure high security in colour image encryption?
- Which optimisation strategies are most effective in tuning hyperchaotic parameters to maximise security metrics such as NPCR, UACI, and entropy in the proposed image encryption algorithms?
- How can denoising autoencoders be integrated with mem-element-based hyperchaotic systems and DNA coding to enhance robustness against noise and improve visual data integrity after decryption?
- What design methodology enables adaptive image compression using deep learning techniques, fractional-order hyperchaotic systems, and DNA coding to achieve secure and efficient image transmission?
- How do the proposed image encryption algorithms perform under various cryptographic attack models, and how do their evaluation metrics (e.g., PSNR, SSIM, NPCR, UACI) validate their effectiveness for real-world applications?

1.4 Objectives of the Study

1.4.1 General Objective

The general objective of this study is to design and evaluate advanced image encryption techniques for enhancing the security of sensitive visual data in personal, business, medical, and military applications by developing chaos-based systems using memelements, identifying the most effective hyperchaotic models according to performance criteria, and formulating hybrid encryption algorithms that integrate the developed hyperchaotic system, DNA coding, one-dimensional chaotic maps, neural networks, and autoencoders for robust image protection.

1.4.2 Specific Objectives

This study aims to enhance image encryption for secure transmission and storage through the following specific objectives:

- To develop and analyze chaotic and hyperchaotic systems utilizing memory elements such as memristors, memcapacitors, and meminductors.
- To design a color image encryption technique based on a Memelements-based hyperchaotic system combined with DNA coding strategies.
- To optimize memelement-based hyperchaotic systems and develop image encryption algorithms that integrate optimized hyperchaotic systems, DNA coding, and a hybrid one-dimensional chaotic map for enhanced encryption performance.
- To create an efficient image encryption algorithm that integrates denoising Autoencoders, a Memelements-based hyperchaotic system, and DNA coding procedures.
- To establish a novel image encryption scheme that integrates adaptive image compression through neural networks, memelements-based fractional-order hyperchaotic systems, and DNA coding, enhancing both security and transmission efficiency.

Finally, each developed image encryption algorithm will be evaluated through comprehensive testing, analyzing various image encryption metrics and assessing resilience against cryptographic attacks, including noise, statistical, differential, and brute-force attacks.

1.5 Significance of the Study

Image communication is vital in healthcare, surveillance, and defence. This study proposed advanced encryption solutions that address the limitations of conventional methods, offering strong protection for sensitive images through innovative techniques.

- Addresses shortcomings of traditional image encryption methods: This study highlights the significant weaknesses of conventional algorithms, particularly their inefficiency with high pixel correlation and large image sizes, by introducing encryption techniques specifically designed for visual data.
- Introduces Hyperchaotic and Mem-Element-Based Security Models: This research enhances unpredictability and increases resistance to brute-force and differential attacks by utilizing hyperchaotic systems driven by memristors and memcapacitors.

- Enhances encryption complexity with DNA-inspired coding: DNA coding introduces biological-level intricacies to the encryption, making the encoded images substantially more challenging to decipher through statistical or plaintext attacks.
- Utilizes Deep Learning for Smart, Adaptive Processing: The combination of CNNs and Autoencoders facilitates intelligent image compression and encryption that adjusts to image features while preserving security and visual quality..
- Combining Compression and Encryption for Efficiency: Compressed Sensing and CNNs significantly reduce image data sizes by up to 3.75 times without compromising encryption strength, making them ideal for real-time or bandwidth-limited applications.
- Demonstrates Strong Security Metrics and Noise Resilience: All methods exhibit high entropy (~ 7.999), strong NPCR ($>99.6\%$), and resilience to noise and transmission errors, ensuring reliable image protection in dynamic environments.

This study proposed a modular and scalable framework for secure image handling, setting a new standard in the field. This advancement promotes further research and practical applications in sensitive areas worldwide, including healthcare, defence, and cloud services, particularly in Ethiopia.

1.6 Scope of the Study

This research develops efficient image encryption techniques using hybrid methods with hyperchaotic systems to enhance security, privacy, and image quality while addressing algorithm limitations. Incorporating mem-element-based hyperchaotic systems and DNA coding with deep learning shows promise in cybersecurity, medical imaging, cloud storage, military applications, and protecting multimedia communications and IoT devices. Efforts are focused on improving efficiency and resilience against cyber threats in secure image processing.

1.7 Framework of the Proposed Study

This project workflow diagram outlines a clear, structured pathway for developing and testing secure image encryption algorithms. It begins with analyzing memelements models and the design of chaotic systems, followed by selecting appropriate hyperchaotic systems using Lyapunov exponents. Input images are then processed through four distinct branches of algorithm development, each incorporating advanced techniques such as DNA coding, optimization, autoencoders, convolutional neural networks (CNNs), and compressed sensing.

These algorithms are evaluated based on performance metrics and resilience to attacks. Finally, the results are compared to draw conclusions and recommendations for future improvements, ensuring robust, adaptive, and efficient image encryption. Figure 1.1 shows the flow diagram outlining the proposed work on mem-element-based chaotic and hyperchaotic systems for image encryption.

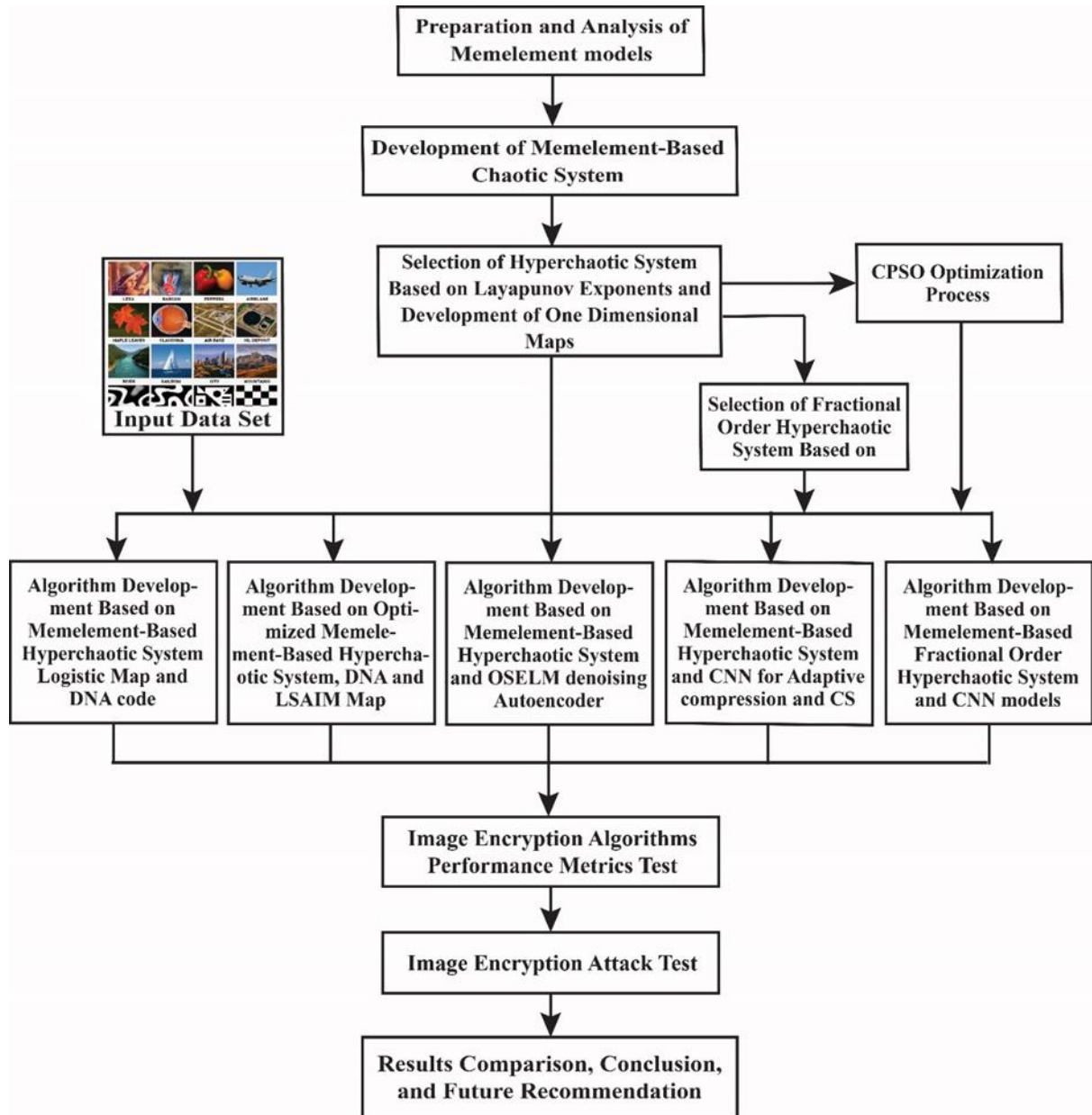


Figure 1.1: Flow Diagram of the Proposed Work.

1.8 Contributions of the Dissertation

This study highlights advancements in chaotic digital colour image encryption. It aims to improve the strength and quality of encryption and decryption techniques by combining chaos theory with advanced cryptographic principles. Key contributions include:

Contribution 1: A Hybrid Image Encryption Framework Combining Memelement-Hyperchaos, DNA Computing, and Logistic Maps: This research developed a novel chaotic image encryption technique tailored for the secured transmission of digital images. A hybrid cryptosystem is integrated into a four-dimensional memelements-based hyperchaotic system, DNA coding, and logistic maps. This fusion expands the key space, enhances sensitivity to initial conditions, and significantly improves resistance to brute-force, statistical, and differential attacks.

Contribution 2: Design and Implementation of a Memelement-Based Hyperchaotic System with Unique Dynamic Properties: A new memelements-based hyperchaotic system is developed, exhibiting coexisting attractors, infinite equilibria, multi-stability, and controllable initial states. These dynamic properties are leveraged to strengthen encryption schemes for secure communication. Simulation results demonstrate superior performance metrics (e.g., entropy, NPCR, UACI) compared to conventional image encryption techniques.

Contribution 3: Optimized Hyperchaotic Sequence Generation via CPSO and LSAIM for Robust Encryption: The parameters of a four-dimensional memristor-based hyperchaotic system are optimized using Chaotic Particle Swarm Optimization (CPSO) to generate highly unpredictable and complex behaviour. A novel Logistic-Sine Adjusted Integrated Map (LSAIM) is introduced to produce chaotic sequences with higher Lyapunov exponents. This approach significantly increases resistance to cryptanalytic attacks when combined with DNA coding.

Contribution 4: Enhanced Robustness through Noise-Resilient and Anti-Clipping Mechanisms: This study introduces an OSELM autoencoder to enhance decryption accuracy in distorted conditions and address noise, intensity saturation, and data loss in image encryption. An anti-clipping mechanism that uses dynamic pixel permutation based on chaotic sequences was also proposed. This method improved entropy, increased pixel variability, and enhanced resilience in noisy environments.

Contribution 5: Integrated Deep Learning-Based Compression and Chaotic Encryption for Efficient Secure Transmission: A context-aware image encryption-compression framework is developed to generate saliency-driven importance maps using pre-trained CNNs (VGG16, VGG19, MNV2). These guided adaptive compression while maintaining critical visual information. The system combined a 5D hyperchaotic map, a Logistic-Sine-Cubic Iterative

Map (LSCIM), and DNA operations for multi-layer encryption. It offered a vast keyspace (2^{1276}) and high processing efficiency, ideal for modern image transmission applications.

Contribution 6: CNN-Based Adaptive Compression, Integrated Framework Using Fractional-Order Memelement-Based Hyperchaotic System with Hybrid Chaotic Maps, and DNA Coding: This study introduced a framework for image compression and encryption that combined a fractional-order memelement-based hyperchaotic system with a hybrid one-dimensional chaotic map to enhance image encryption. It utilized dynamic DNA encoding and decoding for improved security and employs CNN-based adaptive compression guided by deep learning models like VGG16, VGG19 and MobileNet to create saliency-aware quantization maps. This integration enhanced encryption strength and key sensitivity, making the framework ideal for bandwidth-constrained and security-critical applications.

1.9 Structure of the Dissertation

This dissertation is structured into seven main chapters to provide a coherent flow of ideas and a logical progression of the research. Each chapter builds upon the previous one to present a comprehensive study of advanced image encryption frameworks. The organisation is as follows:

Chapter 2 presented a literature review on image encryption, highlighting its importance for secure communication. The methodology section outlines the systematic approach used in the study. This review contrasts traditional image encryption methods with chaotic systems and DNA coding, highlighting key security criteria like statistical evaluations and noise resistance. It discusses advancements in AI-based preprocessing, DNA encryption, and memelements-based hyperchaotic systems and identifies future research gaps.

Chapter 3 discussed the developments of hyperchaotic systems using memelements like memristors, memcapacitors, and meminductors, highlighting their nonlinear and memory-dependent properties. The chapter covers the basics of memelements, integer-order and fractional-order hyperchaotic systems, and optimization techniques like chaotic particle swarm optimization. It concludes with a brief overview of classical one-dimensional chaotic maps and innovative hybrid maps to improve system performance.

Chapter 4 presented a new image encryption algorithm that enhances security by combining a memelement-based hyperchaotic system with DNA coding. It features DNA encoding rules, a logistic map, and a hyperchaotic memcapacitor model to improve robustness and attack resistance. This chapter described the development of encryption and decryption techniques,

assessing performances using performance metrics such as histogram analysis, correlation, key sensitivity, and metrics like MSE and PSNR, demonstrating adequate security for image data during transmission.

Chapter 5 presented a colour image encryption method utilizing a memristive hyperchaotic system optimized by the Chaotic Particle Swarm Optimization (CPSO) algorithm. This method enhanced security by optimizing fourteen parameters with DNA coding and a logistic sine-adjusted integrated map (LSAIM). It demonstrated strong resistance to attacks and features ample key space, with security metrics such as NPCR, UACI, MSE, PSNR, and MAE. Additionally, the chapter discussed military communications, medical imaging, and secure data storage applications and suggests future enhancements for hyperchaos-based encryption.

Chapter 6 introduced a method for image encryption and denoising that integrates an OSELM autoencoder with a memelements-based hyperchaotic system. This approach effectively decreased noise in encrypted images while confirming strong security, sensitivity and image quality for key variations. Simulation results showed an extended key space, strong resistance to attacks, and protection against different images, highlighting its potential for secure digital image transmission and storage in fields like medicine, astronomy, and defence.

Chapter 7 introduced a new framework for securely transmitting sensitive images using adaptive compression and robust encryption. The framework utilized Convolutional Neural Networks (CNNs) such as VGG19 and MobileNetV2 to generate an efficient importance map and ensure high-quality compression. This hybrid system combined integer-order and fractional-order hyperchaotic mechanisms with a Logistic-Sine-Cubic Iterative Map (LSCIM) and DNA coding to enhance security. As a result, the framework achieves significant compression with near-lossless quality, making it ideal for real-time medical imaging and secure surveillance.

CHAPTER 2

LITERATURE REVIEW

In today's digital environment, protecting visual data is essential in healthcare, defense, and consumer technology. Images may contain sensitive information, making them vulnerable to breaches via 5G networks, cloud platforms, and Internet of Things (IoT) devices. Traditional encryption methods are often inadequate, highlighting the need for specialized solutions. This literature review explores advanced image security strategies, such as hyperchaotic systems, DNA-based encoding, and deep learning, highlighting their benefits for secure image transmission and storage.

2.1 The literature review methodology

The literature review methodology outlines the systematic procedure for the literature review, providing a detailed and impartial examination of research directions, improvements, and gaps in chaos-based image encryption, memelements-based hyperchaotic systems image encryption, and AI-driven image security methods. A systematic literature search was performed across various multidisciplinary academic databases to collect peer-reviewed journal articles, conference proceedings, preprints, and patents.

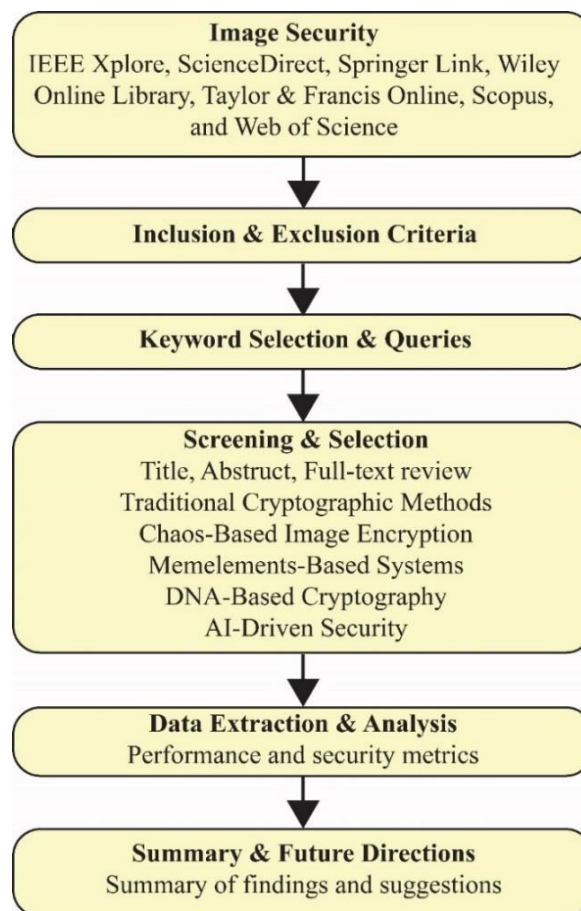


Figure 2.1: Literature Review Methodology

The systematic literature review process outlined in the flowchart provides a thorough examination of image security research. It begins by sourcing articles from reputable databases, such as IEEE Xplore, ScienceDirect, and Scopus. Inclusion and exclusion criteria are then applied to filter relevant studies. Keyword searches identify papers on key topics, including traditional cryptography, chaos-based encryption, and AI-driven security. After rigorous screening, data extraction focuses on performance and security metrics. The review yields a synthesis of findings, identifies research gaps, and provides recommendations for future developments in secure image transmission. The complete methodology for the literature review is organized and illustrated in Figure 2.1.

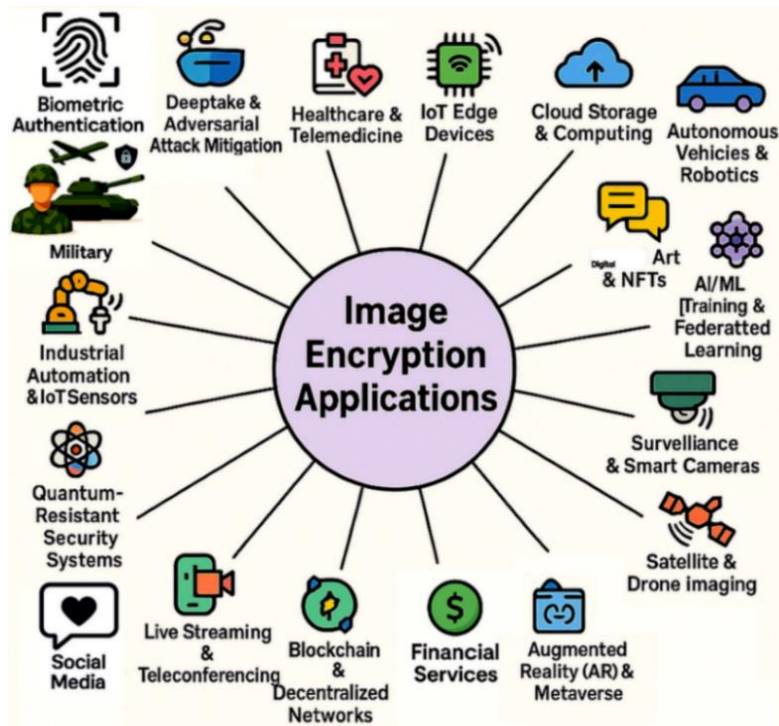


Figure 2.2: Areas of Application for Current Image Encryption

Figure 2.2 illustrates the current application areas of image encryption (Alawida, 2024; Alexan et al., 2024; Brabin et al., 2022; Z. Chen et al., 2022; Ejaz et al., 2018; El-Shafai et al., 2021; Fan, 2024; Kakkad et al., 2019; Y. Li et al., 2021; W. Liang et al., 2021; Mahadik et al., 2024; K. Manikandan et al., 2011; Nadimpalli & Rattani, 2024; Roy et al., 2021; Sekar et al., 2023; Shah et al., 2021; Shetty et al., 2024; Tian & Jiang, 2024; X. Yang et al., 2021; Yocam & Wang, n.d.).

2.2 Protecting Visual Data in Smart and Connected Environments

Images play a crucial role across various fields due to their ability to visualize information. In the context of the IoT (Hussain et al., 2024) and portable devices such as smartphones and surveillance drones, images can contain private information, making them vulnerable to

privacy breaches if lost or stolen. For example, in medical contexts, patient privacy must be safeguarded. In business organizations, it's essential to protect graphical codes, patents, and trademarks. In terms of national security, operational visual information needs to remain confidential. With the advancement of cloud computing, more image data is being processed and stored in the cloud. Additionally, the rollout of 5G technology makes image storage and transmission security even more critical. The security of images is vital for advancing numerous fields, including medicine (Bencherqui et al., 2024; Sadia et al., 2024), photography (J. Yu et al., 2023), biology (Belete et al., 2025b), defence (Charmouti et al., 2022), astronomy (Yao et al., 2023), and so on (F. Yu et al., 2022).

2.2.1 Communication Security

The rapid evolution of digital communication technologies has revolutionized the transmission of multimedia data, including images, videos, and documents. However, this progress has introduced significant security challenges, particularly in safeguarding sensitive information from unauthorized access, tampering, and cyberattacks.

Conventional block cyphers are often inadequate (J. Chen et al., 2020; Wen et al., 2022). The encryption standards (e.g., DES, AES, RSA), optimized for textual data, fail to address the unique challenges of images, high redundancy, pixel correlation, and computational inefficiency (Alexan, Gabr, et al., 2023; Fang et al., 2021; Hadj Brahim et al., 2024; Z. T. Hu et al., 2024; Iqbal et al., 2021; H. Lin et al., 2022; Malik et al., 2023). For instance, AES exhibits NPCR values of approximately 99% and small key space 2^{128} for encrypted images (Malik et al., 2020; Y. Zhang, 2018), rendering it susceptible to differential and brute force attacks. These limitations have spurred interest in chaos-based encryption, leveraging nonlinear dynamics to generate pseudorandom sequences resistant to cryptanalysis (Akif et al., 2021). They provide a more effective solution for secure communication, as their pseudo-randomness and sensitivity to initial conditions enhance cryptographic processes (X. Gao, 2021).

2.2.2 Image Security

Images play a vital role in various fields such as healthcare (e.g., MRI scans), the Internet of Things (e.g., surveillance systems), defence (e.g., satellite imagery) (Bencherqui et al., 2024; Charmouti et al., 2022; Hussain et al., 2024; Sadia et al., 2024), and government (Xuejing & Zihui, 2020). Due to their sensitive content, images are often primary targets for cyber threats (Rashid & Hussein, 2023). Chaotic encryption techniques are emerging as options to traditional methods like Data Encryption Standard (DES), Advanced Encryption Standard (AES), and

Rivest–Shamir–Adleman (RSA), which often have limitations. By employing deep neural networks, image encryption can improve security using cryptography’s trapdoor feature, resulting in more effective protection methods (X. Gao, 2021; Maniyath & V, 2020). Digital images, central to modern applications like medical imaging, autonomous vehicles, and satellite surveillance, are increasingly vulnerable to adversarial attacks due to their transmission over unsecured networks (Dantas et al., 2024; Lu et al., 2022; Sakthi kumar & Revathi, 2024). Securing image transmission continues to be a significant challenge (Rashid & Hussein, 2023). Traditional encryption algorithms often perform poorly with two-dimensional images, as seen in metrics like entropy and NPCR. In contrast, chaos-based encryption uses the properties of chaotic systems, such as sensitivity to initial conditions, to generate secure cryptographic keys and improve data diffusion.

Figure 2.3 shows image encryption and decryption operation. A plain image (P) is converted into a ciphered image (C) using an encryption function and key. The ciphered image looks like random noise and can be decrypted to its original form using a decryption function and critical.

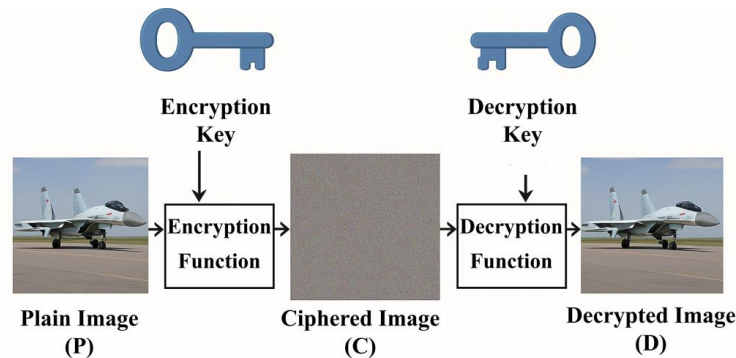


Figure 2.3: Block Diagram Illustrating the Process of Image Encryption and Decryption.

Figure 2.4 categorises image security techniques into two main groups: encryption and data-hiding techniques. Traditional algorithms, such as AES, RSA, and ElGamal, are based on frameworks established by Wolfgang and Delp (1998)(297.Pdf, n.d.), and they contrast with modern methods that incorporate chaos-based encryption and frequency-domain techniques, including DCT, DWT, and FFT.

Current approaches for image security possess steganography, which conveys information within other data, and watermarking, which protects copyrights and detects tampering. Image encryption techniques range from conventional methods to advanced ones like post-quantum cryptography and homomorphic encryption. Chaos-based encryption utilizes chaotic maps and pseudo-random generators. Additionally, emerging data-hiding techniques include blockchain-based encryption and zero-knowledge proofs, all of which aim to enhance image security.

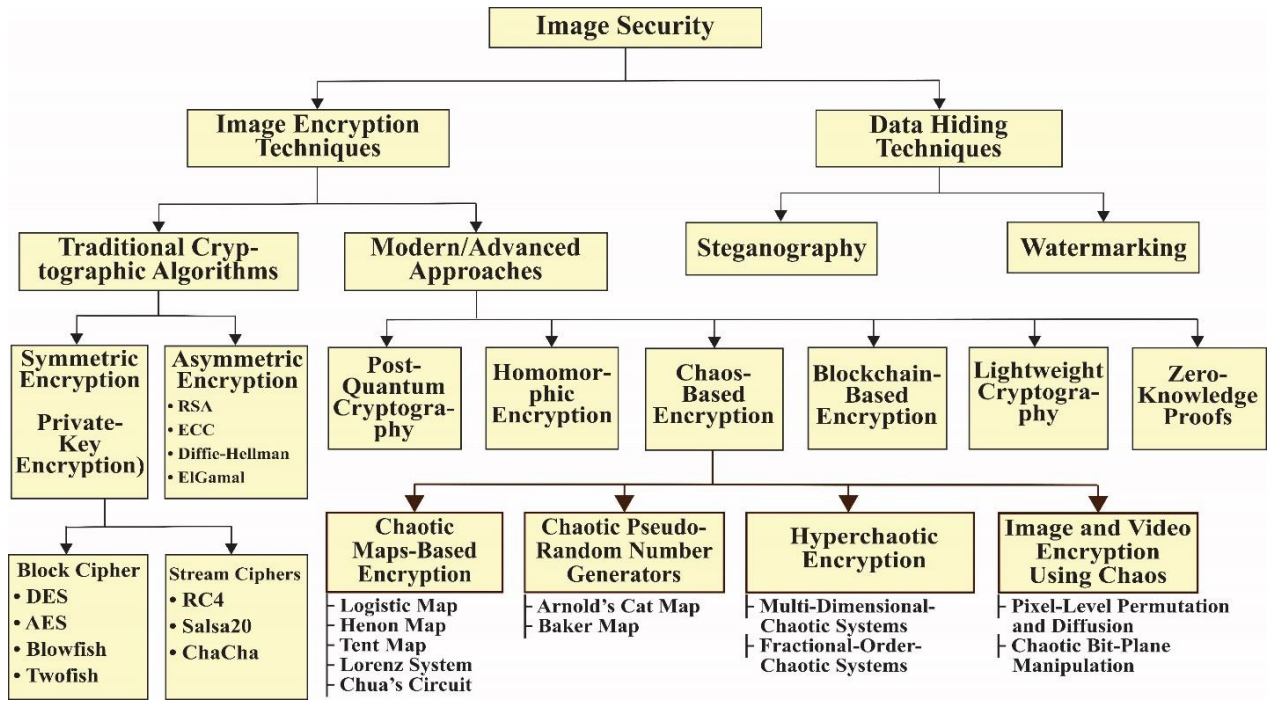


Figure 2.4: Overview of Image Security Techniques.

2.2.3 Memelements: A Paradigm Shift

The advancement of bioengineering, neural networks, Electronic engineering, communication engineering, and computer science has been significantly influenced by memelements (Ersoy & Kacar, 2023; Shanmugam et al., 2020). Memelements refer to electronic components that store and manipulate information beyond the limitations of traditional linear resistors, capacitors, and inductors. This includes memristor, memcapacitor, and meminductor elements that exhibit memory behaviour in certain conditions. These components are an expansion of fundamental circuit elements and are denoted as shown in Figure 2.5 (Bhardwaj & Srivastava, 2022; Marco et al., 2021; Massimiliano et al., n.d.; Raj et al., 2021). Figure 2.5 illustrates that the fundamental circuit elements and their nonlinear versions are considered “singular” as they lack memory. This is indicated by the dashed curves representing “cuts” in the plane of memelements (Massimiliano et al., n.d.).

The memristor was predicted in 1969 based on principles from the textbook "Nonlinear Network Theory." It was mathematically confirmed a year later as the fourth fundamental circuit element. On November 25, 1970, Leon O. Chua published his paper, "Memristor: The Missing Circuit Element," in the journal IEEE Transactions on Circuit Theory. It has been noted that only three two-terminal circuit elements were defined by relationships between current, voltage, charge, and flux linkage (Chua, 1971). Since the relationship between flux

linkage and charge was lacking, he proposed the memristor, which behaves like a nonlinear resistor with memory (Chua, 1971).

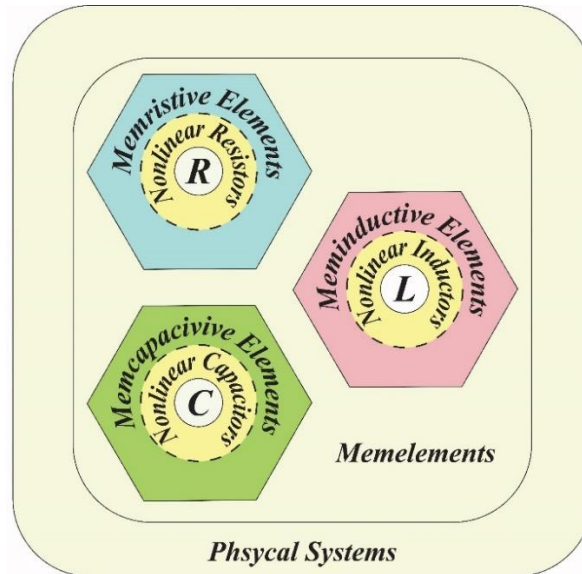


Figure 2.5: Basic Circuit Elements Exhibiting In-Memory and Nonlinear Responses.

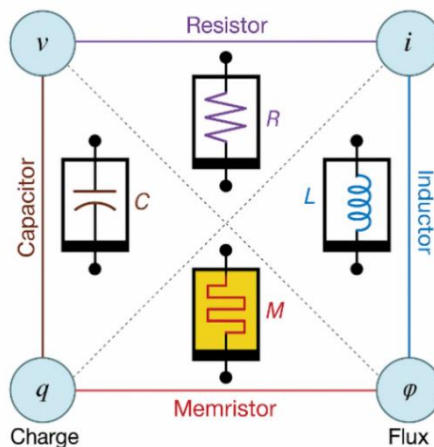


Figure 2.6: The Four Fundamental Two-terminal Circuit Elements (Chua, 2018).

The four fundamental two-terminal elements are depicted in Figure 2.6: a resistor links voltage (v) and current (i), an inductor links flux (ϕ) and current (i), a capacitor links charge (q) and voltage (v), and a memristor links flux (ϕ) and charge (q) (Chua, 2018). Chua introduced the memcapacitor and meminductor by applying memory circuit concepts to capacitive and inductive systems. Researchers established connections between the time-integral of charge (σ) and magnetic flux (ϕ) for the memcapacitor, and between the time-integral of magnetic flux

(ρ) and charge (q) for the meminductor, as shown in Figure 2.7 (Petráš, 2020, 2022; Romero et al., 2021; X. Y. Wang et al., 2012).

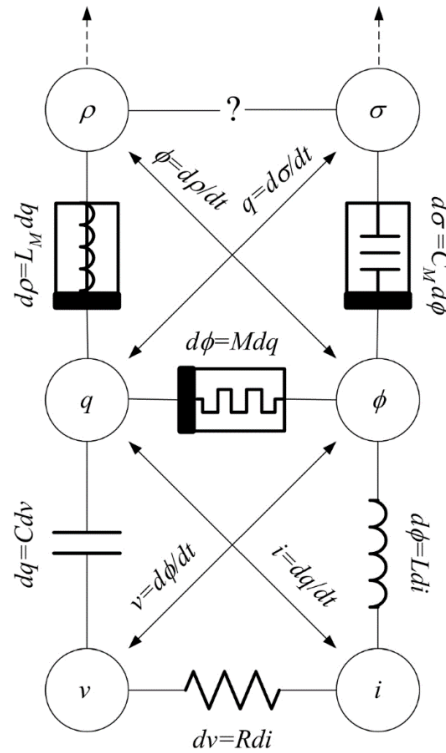


Figure 2.7: System of Basic Passive Memory-less and Mem-elements (Petráš, 2020, 2022).

Memelements, which include memristors, memcapacitors, and meminductors, significantly enhance the design of nonlinear systems by incorporating memory-dependent dynamics that promote chaotic behavior (Di Ventra et al., 2009; Massimiliano et al., n.d.; Raj et al., 2021; Yuan & Li, 2019). Unlike traditional components such as resistors and capacitors, memelements retain historical state information, enabling the creation of complex hyperchaotic systems characterized by higher-dimensional attractors. For example, memristor circuits have demonstrated entropy values exceeding 7.9 and positive Lyapunov exponents greater than zero. However, meminductors are still underexplored due to challenges in their fabrication (Akgül et al., 2021; Ersoy & Kacar, 2023; Sharma et al., 2020).

2.2.4 Chaotic and Hyperchaotic Systems

Chaos is a widely investigated phenomenon in scientific, engineering, and mathematical fields. Its defining characteristic is sensitivity to initial conditions, causing neighbouring trajectories to diverge exponentially over time. While this makes chaos challenging to predict, the emergence of chaos synchronization has opened up new interdisciplinary research avenues

with potential applications in physics, engineering, secure communications, biomedical sciences, and other fields (Alibraheemi et al., 2022; Almatroud Othman et al., 2016).

A chaotic system characterises an inherently complex phenomenon within nonlinear systems described by unpredictable trajectories, sensitivity to initial values, chaos control parameters, and ergodic trajectory paths (Akif et al., 2021; Cheng et al., 2020; X. Gao et al., 2022; S. Yan et al., 2023). These features align perfectly with the essential requirements of cryptography. The structural similarities between chaotic systems and cryptography have encouraged researchers to discover the application of chaotic systems in cryptographic methods. Therefore, chaotic cryptography has become a crucial part of modern encryption techniques. Due to the exceptional properties of non-linear chaotic systems, it is extensively used in image encryption (J. Cai et al., 2023; C. M. Kumar et al., 2022; S. Zhang & Liu, 2021), secure communication (Akgül et al., 2021; Bonny et al., 2023; F. Yu et al., 2020), and other fields of study.

Early efforts focused on low-dimensional systems (e.g., Logistic map (Fadhil et al., 2021)), but their small key space (R. Li et al., 2019) and predictability prompted the adoption of high-dimensional hyperchaotic systems (e.g., Lorenz's, Rossler's, Chen's, Liu's) and their fractional-order variants (Ma et al., 2020; Sánchez-López, 2020).

The Lorenz attractor, Logistic map, Rössler attractor, Tent map, Hénon map, and Piecewise linear chaotic map are notable chaotic systems. The Lorenz system is the first chaotic attractor model in a three-dimensional autonomous system, while Chen's system features a more complex topological structure (G. H. Li, 2007). Rössler first reported hyperchaos in 1979, identifying the Lorenz attractor, Logistic map, Rössler attractor, and Tent map as examples of hyperchaotic systems (Pang & Liu, 2011). Although creating hyperchaotic attractors is appealing in theory, it is challenging in practice. Researchers have studied hyperchaotic attractors in multidimensional systems and circuits based on memristors (Rossler, 1979; Teng et al., 2020).

One-dimensional logistic chaotic mappings in encryption have limited key spaces and are vulnerable to attacks. Due to their susceptibility to brute-force threats, researchers are turning to high-dimensional hyperchaotic systems to enhance security (Zolfaghari & Koshiba, 2022). The later investigation used the Lorenz system, while the earlier employed the Rossler chaotic system. In references (Zolfaghari & Koshiba, 2022) and (Alexan, Chen, et al., 2023), PRNG encryption keys are generated using the Rossler system.

Research on chaotic systems has expanded to include hyperchaotic systems like the hyperchaotic Chen system, which offers complex behaviour and better feedback control (G. Chen, 1999; H. Gao & Wang, 2021). It improves encryption by modifying pixel values and using fractal images, increasing key space and sensitivity while resisting common attacks (Mohamed et al., 2020). Zhou et al. introduced a hyperchaotic oscillator with a logarithmic memcapacitor in a Chua circuit, enabling various attractors based on initial conditions (Njitacke et al., 2022; W. Zhou et al., 2020). In 2010, Hu presented a disordered oscillator with a memcapacitor, and in 2017, Wang developed a turbulent oscillator with both a memcapacitor and a meminductor.

Hyperchaotic systems have more than one positive Lyapunov exponent and provide increased complexity and security (H. P. Ren et al., 2017)(J. Zhang & Xu, 2023). Recent advancements have combined fractional-order calculus and mem-elements to enhance chaotic dynamics further (Abro et al., 2023; Gabr et al., 2023; Ma et al., 2020; Petráš, 2022). For example, fractional-order Chen systems achieve an entropy greater than 7.9 (Alexan, Gabr, et al., 2023), while the fractional-order Lorenz and Rossler systems have greater complexity than the conventional Lorenz or Rossler systems(X. Zhang et al., 2015). Chaotic systems can be optimized using nonlinear analysis (Setoudeh & Sedigh, 2021; W. Zhou et al., 2020), fractional order methods (Alexan, Gabr, et al., 2023; Y. G. Yang et al., 2024), and optimization methods (Ghazvini et al., 2020; Houssein et al., 2021a; Shami et al., 2022; L. Wang et al., 2017).

In 2019, Yuan created a simple chaotic oscillator using a combination of a memristor, memcapacitor, and meminductor (Z. Guo et al., 2023; Z. Hu et al., 2010). Memristors and memcapacitors are vital for low-power, high-density memory systems and enable diverse chaotic events through fractional-order circuits (Akgül et al., 2021; Bhardwaj & Srivastava, 2022; Sharma et al., 2020; Q. Zhao et al., 2019).

A new method for encrypting colour images uses dynamic DNA and four-dimensional memristive hyper-chaos to create chaotic matrices. It encodes the image into three DNA matrices for enhanced security and generates the final encrypted image by decoding the DNA and combining the components while assessing system stability (M. Guo et al., 2021; Murillo-Escobar et al., 2023).

A new chaotic circuit, based on Chua's dual circuit and using a mem-element emulator, can generate hyper-chaos with memristors, memcapacitors, or meminductors. By connecting

different mem-elements and adjusting their parameters, this design is valuable for studying chaotic circuits (Q. Zhao et al., 2019).

The study analyzed a fractional-order, four-dimensional hyperchaotic memristive system using phase diagrams, Lyapunov exponents, and bifurcation diagrams. It also introduces a secure colour image encryption scheme based on DNA sequence operations to improve the security of transmitted images (P. Li et al., 2019). Fractional-order memristive, meminductive, and memcapacitive systems exhibit diverse, dynamic behaviours, including multiple and hidden attractors and transitions between chaotic and hyperchaotic states. These features indicate potential applications in oscillator circuits and secure communication (Borah & Roy, 2021).

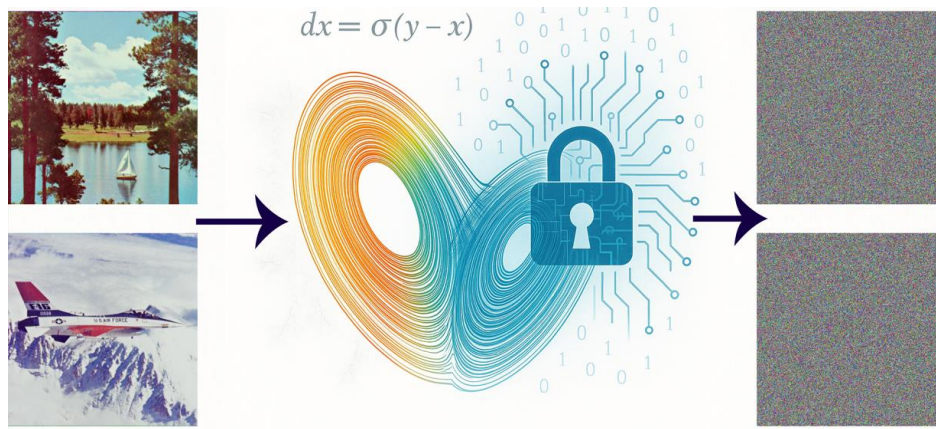


Figure 2.8: Integration of Chaotic Systems and Cryptography.

Figure 2.8 shows the connection between chaotic system theory and cryptography. It features a colourful attractor that highlights sensitivity to initial conditions and the role of chaos in secure encryption. The figure shows the importance of high-dimensional chaotic systems for modern encryption and image security, particularly with enhancements from fractional-order calculus and memelements.

2.2.5 DNA Coding

DNA computing, inspired by biological processes, introduces a new layer of security through nucleotide-based encoding, utilizing the bases A, T, C, and G, with operations such as XOR and addition (Alruily et al., 2021; M. Liu & Ye, 2021a). Dynamic DNA coding, driven by hyperchaotic sequences, addresses the limitations of static methods (Alexan, Gabr, et al., 2023; Erkan et al., 2022). However, despite its potential, the biological feasibility of these methods remains a challenge, as few studies utilize actual DNA sequences (Iqbal et al., 2021; Sakran et al., 2023).

Many researchers have recently developed image encryption techniques that incorporate DNA coding alongside chaotic mapping. One approach involves using DNA segment coding, where odd and even columns are swapped to increase security. This combination of DNA computing and chaotic encryption aims to enhance overall protection (Chai et al., 2017; X. Chen et al., 2023).

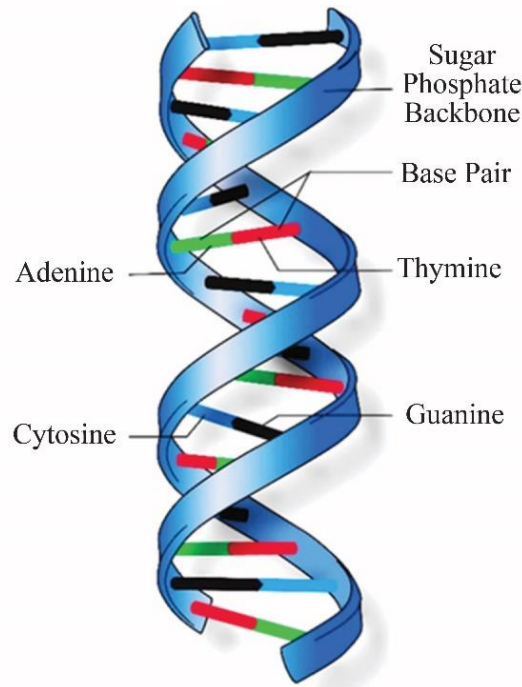


Figure 2.9: The Structure of Part of A DNA Double Helix (Alruily et al., 2021).

A chaotic image encryption algorithm enhances security by using dynamic DNA coding and RNA computing to convert image data into pseudo-random sequences, creating ciphertext images (Cun et al., 2023). While DNA-based encryption is innovative for data transmission, few systems encrypt text with DNA, as most prioritize data concealment over actual encryption.

Recent advancements in asymmetric DNA encryption have strengthened the security of simple scripts, increasing the popularity of DNA cryptography (Alruily et al., 2021). DNA is a long-lasting storage medium, capable of preserving information for millions of years, and is essential in protein production. It consists of four bases: adenine (A), thymine (T), cytosine (C), and guanine (G), which are represented by the binary codes A (00), G (01), C (10), and T (11) (Alruily et al., 2021; Chai et al., 2017). Table 2.1 shows the pairing rules for these bases (Alruily et al., 2021).

Table 2.1: DNA Representation of Bits

DNA sequence	Bit 0	Bit 1	Binary
A	0	0	00
G	0	1	01
C	1	0	10
T	1	1	11

The operations follow standard DNA computation rules described in Table 2.2 (Kengnou Telem et al., 2021).

Table 2.2: Basic DNA Coding Operations Used in Image Encryption Schemes

DNA Addition Table (Addition is modulo-4 on binary)				
Addition (+)	A (00)	C (01)	G (10)	T (11)
A	A	C	G	T
C	C	G	T	A
G	G	T	A	C
T	T	A	C	G
DNA Subtraction Table (Subtraction is modulo-4 on binary)				
Subtraction (-)	A (00)	C (01)	G (10)	T (11)
A	A	T	C	G
C	C	A	T	G
G	G	C	A	T
T	T	G	C	A
DNA XOR Table (Bitwise XOR on binary values)				
XOR	A (00)	C (01)	G (10)	T (11)
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A
DNA XNOR Table (Bitwise XNOR on binary values)				
XNOR	A (00)	C (01)	G (10)	T (11)
A	A	T	C	G
C	T	A	G	C
G	C	G	A	T
T	G	C	T	A

2.2.6 Deep Learning and Autoencoders

Deep learning algorithms, specifically Convolutional Neural Networks (CNNs) and autoencoders, are primarily used in image processing and help alleviate computational challenges associated with encryption. Lightweight CNNs reduce encryption times through data compression (Alqahtani et al., 2021; Dantas et al., 2024), while Extreme Learning Machine (ELM) autoencoders improve denoising performance, enhancing Peak signal-to-noise ratio (PSNR) in noisy environments (Jessica, 2020; N. Zhang et al., 2022). Additionally, hybrid

frameworks that combine chaos theory with artificial intelligence optimize key generation and parameter tuning and improve attack resistance (W. Chen, n.d.; Fang et al., 2022; Raja et al., 2023).

Figure 2.10 illustrates how Convolutional Neural Networks (CNNs) and Autoencoders work together for image encryption. CNNs extract essential features to reduce data size and encryption time. At the same time, the Extreme Learning Machine (ELM) variant of the Autoencoder improves image quality and Peak signal-to-noise ratio (PSNR) through denoising, ensuring secure encryption in noisy conditions.

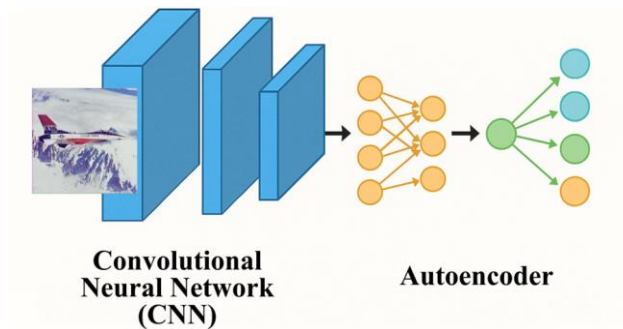


Figure 2.10: Architectures of CNN and Autoencoders for Image Compression and Denoising in Encryption.

2.2.6.1 Deep Neural Network

This approach is particularly effective for large datasets where features are difficult to discern. It allows the system to differentiate individual data points and learn intricate functions. Deep learning (DL) differs from traditional feature engineering by automatically identifying relevant data features without human input. It reveals hidden patterns through multiple layers of nonlinear operations, transforming raw data into abstract representations for learning complex functions (Esteva et al., 2019). Information flows through the system layers, transforming the input space to reveal individual data points, as shown in Figure 2.11 (Meiliana et al., 2019).

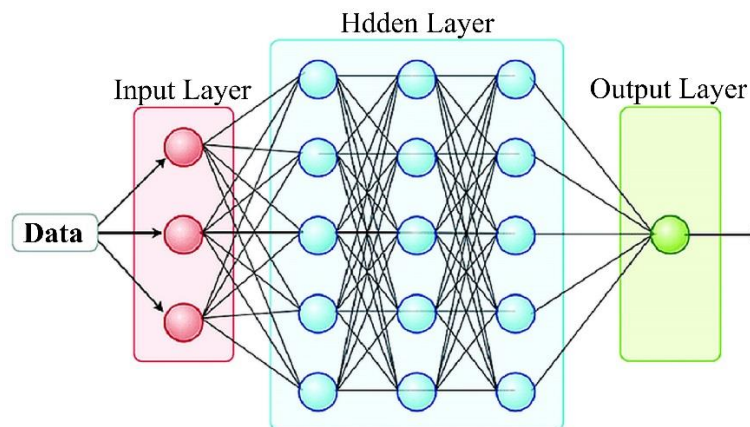


Figure 2.11: A Condensed Deep Neural Network (Meiliana et al., 2019)

Convolutional neural networks (CNNs) have greatly improved image categorization and object recognition by applying multiple filters to images and creating feature maps (Krizhevsky et al., 2017). The input consists of pixel data, while the output is image classification (see Figure 2.12) (Kermany et al., 2018).

Most deep learning algorithms are based on artificial neural networks (ANNs), with CNNs featuring a fully connected final layer compared to the interconnected structure of ANNs (Meiliana et al., 2019). Additionally, memristors are well-suited to serve as synapses between neurons in neural networks (Hong et al., 2021, 2022).

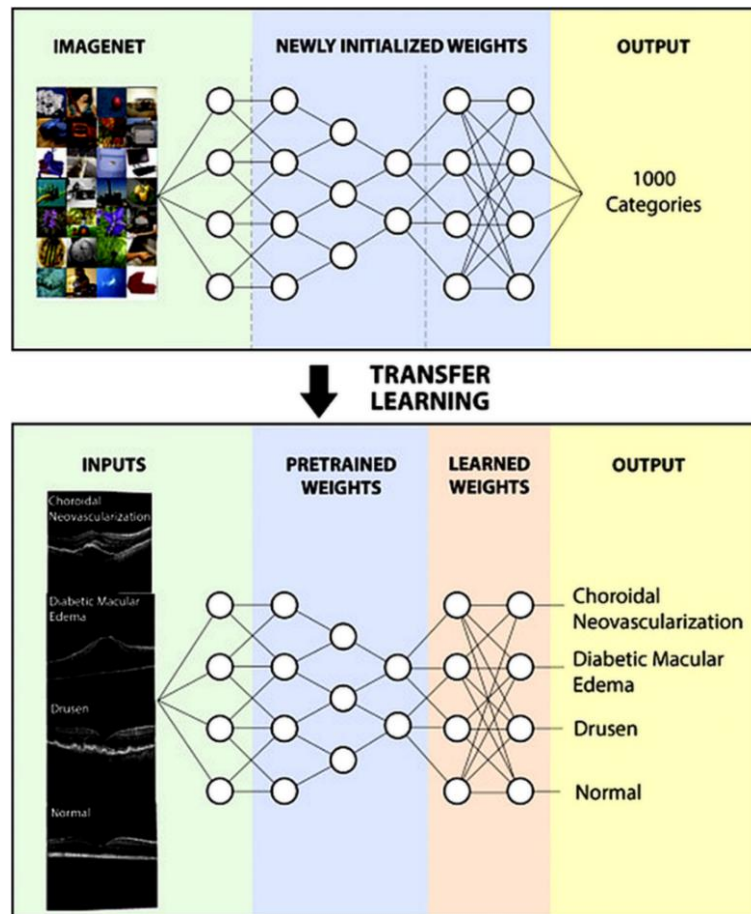


Figure 2.12: Schematic of a Convolutional Neural Network (Kermany et al., 2018).

The authors have developed a method for creating adaptive controllers for reaction-diffusion neural networks (RDNNs) with time-varying delays. They also enhanced image encryption and security using deep convolutional neural networks (CNNs) (Shanmugam et al., 2020). Additionally, a neural network with a nonlinear memristor was proposed for medical image watermarking, utilizing adjustable parameters to balance watermark robustness and invisibility (F. Yu et al., 2022; J. Yu et al., 2023).

2.2.6.2 Autoencoder

An autoencoder (AE) is a feed-forward neural network (Sakurada & Yairi, 2014) where the input and output dimensions are the same, as shown in Figure 2.13 (Torabi et al., 2023). It uses an unsupervised approach to reconstruct input vectors. The main components are the encoder, which transforms the input vector (X) into a hidden representation (H), and the decoder, which reconstructs the output.

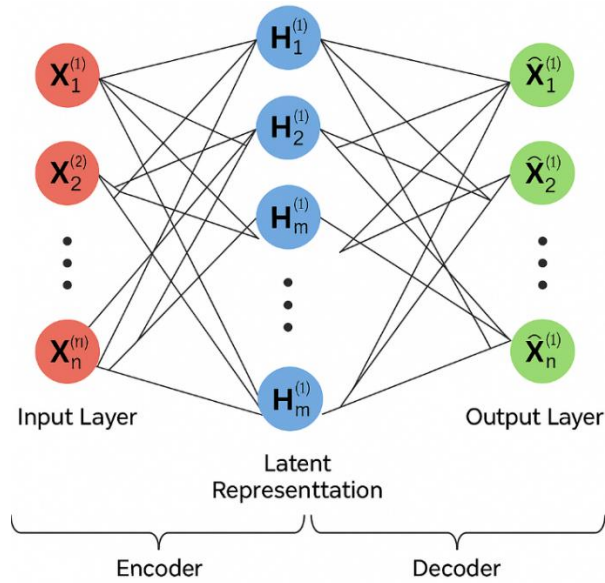


Figure 2.13: The Architecture of a Sample Autoencoder (Torabi et al., 2023).

Figure 2.13 illustrates the typical architecture of an autoencoder. H is represented as:

$$H = \sigma(W_{xh} X + b_{xh}) \quad (2.1)$$

where W_{xh} is a weight matrix, b is a bias vector, and σ is an activation function, such as a sigmoid function or rectified linear unit. The hidden representation H is subjected to the transformation operation in order to use a decoder to recreate the initial input space.

Effective compression and encryption are essential to securely transmit images over low-capacity channels. This method combines deep learning, specifically an autoencoder with the logistic map, to create a hybrid compression and encryption scheme (Gupta & Vijay, 2022).

2.2.6.3 Stacked Autoencoder

An autoencoder (AE) is a nonlinear structure that extracts features from an image at multiple levels. The unsupervised AE model consists of two Restricted Boltzmann Machine (RBM) models connected back-to-back, as illustrated in Figure 2.14. By stacking several AE models together, a stacked autoencoder model is created in which the output of one AE serves as the input for the next.

The stacked autoencoder model includes multiple hidden layers, necessitating the design of a three-stage structure comprising input and output layers and various hidden layers in between. The number of nodes in each layer can vary, but the number of nodes in a hidden layer is always less than that in the corresponding input layer. The Stacked Autoencoder (SAE) captures features at multiple levels, with each level representing a compressed version of the original image (Gupta & Vijay, 2022; F. Hu et al., 2016).

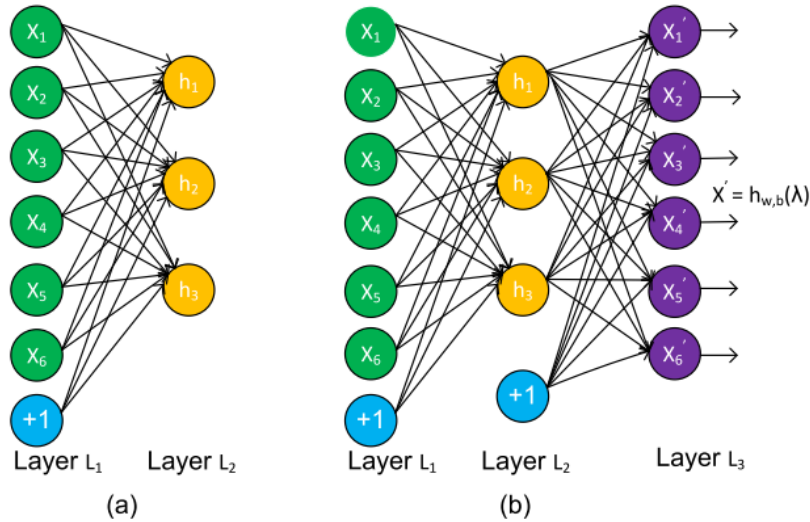


Figure 2.14: (a) RBM Model (b) Autoencoder Model (Gupta & Vijay, 2022).

2.2.7 Integration of Advanced Techniques in Image Encryption

This study introduced a method for encrypting color images using a multi-dimensional hyperchaotic system, dynamic DNA coding, and robust chaotic maps to enhance security and efficiency (Singh, 2024),(C. S. Manikandan et al., 2023). It employs Chaotic Particle Swarm Optimization (CPSO) and Convolutional Neural Networks (CNNs) for fast feature extraction and segmentation, while lightweight CNNs facilitate data compression, reducing encryption times without losing robustness (Sandler et al., 2018),(Mingxing Tan, 2013). Additionally, Extreme Learning Machine (ELM) autoencoders improve denoising and image reconstruction (Alizamir, Ahmed, et al., 2023; J. Wu & Zhang, 2021).

Compressive sensing boost efficiency and security (Isaac et al., 2022; Y. Li et al., 2021). Given the increasing need for secure image transmission in healthcare, astronomy, and military sectors, this research emphasizes the value of integrating these technologies for better compression, encryption, and denoising of digital images.

2.2.8 The Role of AI and Hybrid Architectures

The fusion of chaos theory with deep learning and DNA coding has addressed longstanding efficiency-security trade-offs. Convolutional neural networks (CNNs) enable adaptive

compression, reducing encryption time without compromising robustness (Maharaja Vannia Moorthi et al., 2024), while extreme learning machines (ELM) Autoencoder denoise images pre-encryption, improving PSNR (X. Yang et al., 2019) and MAE (Kaur et al., 2023).

Dynamic DNA coding, which utilizes base-pair transformations like XOR operations (M. Liu & Ye, 2021a), achieves NPCR values exceeding 99%, outperforming static methods. However, fragmented workflows, where denoising, encryption, and compression operate in isolation limit scalability. Figure 2.15 shows how AI and combined architectures improve secure image encryption. Convolutional Neural Networks (CNNs) allow adaptive compression, Extreme Learning Machine (ELM) Autoencoders perform image denoising, and dynamic DNA coding strengthens encryption. This approach boosts workflow efficiency, performance, and scalability.

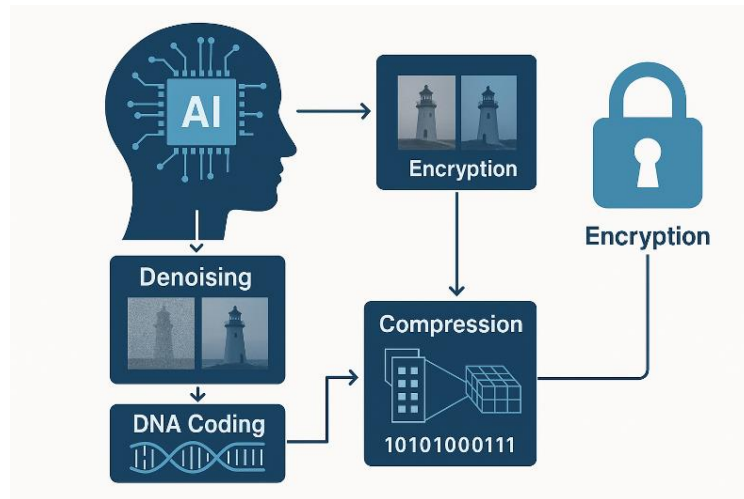


Figure 2.15: The Role of AI and Hybrid Architectures in Secure Image Encryption

2.3 Comparative Analysis of Traditional and Chaos-Based Encryption Methods

2.3.1 Traditional Image Encryption Methods

Conventional encryption algorithms for securing digital image encryption have limitations due to high redundancy and pixel correlation (Chai et al., 2020; T. Wang & Wang, 2020b). This section reviews techniques such as the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Rivest-Shamir-Adleman (RSA), which have been widely employed to secure digital images (Prerna et al., 2013).

AES is effective with strong diffusion and confusion properties, with NPCR values approaching 99% and entropy exceeding 7.8. However, it has a limited key space (2128), making it vulnerable to brute-force attacks and differential cryptanalysis (Y. Zhang, 2018). DES, with its 56-bit key, is now considered obsolete due to advances in computing power (Devi

et al., 2015a). While RSA is valid for key exchange, its computational complexity makes it impractical for real-time image encryption (Prerna et al., 2013).

Table 2.3 compares traditional encryption algorithms based on image entropy, correlation coefficients, and PSNR values. Higher entropy and lower correlation coefficients indicate stronger encryption, while PSNR reflects the distortion introduced by encryption.

Table 2.3: Performance Metrics of Traditional Encryption Algorithms

Encryption Algorithm	Image Entropy (bps)	Maximum Correlation Coefficient	PSNR (dB)
RC4 (Scripcariu et al., 2019)	7.99680	0.00390	8.37810
DES (Scripcariu et al., 2019)	7.99590	0.05580	8.46980
3DES (Scripcariu et al., 2019)	7.99660	0.04550	8.45480
AES (Scripcariu et al., 2019)	7.99650	0.00450	8.40210
Chaotic Map (Scripcariu et al., 2019)	7.99720	0.00860	8.26740
LEA (Scripcariu et al., 2019)	7.99900	0.00470	8.03640
ECC (Scripcariu et al., 2019)	6.76940	0.35020	7.90170

The Lightweight Encryption Algorithm (LEA) surpasses algorithms such as RC4, DES, Triple DES, AES, ECC, and Chaotic Map in terms of increasing image entropy. However, its peak signal-to-noise ratio (PSNR) is lower than that of ECC. While LEA's simplicity and use of precomputed substitution values may expose it to algebraic or side-channel attacks compared to more robust standards like AES or ECC, it maintains low correlation coefficients similar to those of AES. It features low computational complexity, performing two rounds of permutation, substitution, and key addition in Galois Field (GF)(256) (Scripcariu et al., 2019).

2.3.2 Symmetric Encryption Algorithms

Symmetric encryption algorithms, such as the Advanced Encryption Standard (AES), the Data Encryption Standard (DES), and Blowfish, utilize a single key for both encryption and decryption. AES is one of the most secure standards, resisting brute-force attacks due to its large key sizes (128-bit, 192-bit, and 256-bit). However, AES and DES have weak image encryption, as they poorly diffuse pixel values, leading to predictable cypher images. They also face challenges with high entropy and slow performance on large image datasets (Devi et al., 2015b; Kumar Sharma et al., 2021).

Table 2.4 compares symmetric-key encryption algorithms. AES is preferred for its speed and security, while DES and 3DES are obsolete due to brute-force vulnerabilities. Blowfish proposes flexible key sizes, but IDEA poses key management challenges.

Table 2.4: Comparison of Different Symmetric-Key Encryption Algorithms

Properties/Factor	DES	3DES	AES	IDEA	Blowfish
Key Size (Devi et al., 2015a; Kumar Sharma et al., 2021)	56 bits	112 or 168 bits	128, 192, or 256 bits	128 bits	32–448 bits
Block Size (Kumar Sharma et al., 2021)	64 bits	64 bits	128 bits	64 bits	64 bits
Rounds (Kumar Sharma et al., 2021)	16	48	10, 12, 14	8.5	16
Structure (Kumar Sharma et al., 2021)	Balanced Feistel	Feistel	Substitution-Permutation Network (SPN)	Lai-Massey Scheme	Feistel
Developed By (Kumar Sharma et al., 2021)	IBM, 1975	IBM, 1978	Joan Daemen & Vincent Rijmen, 2000	James Massey & Xuejia Lai, 1991	Bruce Schneier, 1993
Objective (Kumar Sharma et al., 2021)	Small key size makes it vulnerable to brute force attacks. It provides strong encryption but is considered insecure.	More secure than DES but significantly slower due to multiple encryption rounds.	Faster and more secure than 3DES. Widely used in modern encryption applications.	Strong security, but vulnerable to certain attacks due to weak key scheduling.	Fast encryption, flexible key sizes, but key changes can slow performance.
Encryption Speed (Devi et al., 2015a)	Slow	Slower than DES	Fast	Moderate	Fast
Decryption Speed (Devi et al., 2015a)	Slow	Slower than DES	Fast	Moderate	Fast
Speed Dependence on Key (Kumar Sharma et al., 2021)(Devi et al., 2015a)	Yes	Yes	No	Yes	No
Security (Devi et al., 2015a; Kumar Sharma et al., 2021)	Insecure (vulnerable to brute-force)	More secure than DES, but still weak	Highly secure, widely used	Secure but weaker than AES	Considered secure, fewer cryptanalysis attempts than AES
Hardware & Software Implementation (Devi et al., 2015a)	Designed for hardware, slow in software	Slower than DES in software but still used	Efficient in both hardware and software	Primarily designed for software	Efficient in software, adaptable to hardware
Objective (Additional) (Kumar Sharma et al., 2021)(Devi et al., 2015a)	Basic encryption but obsolete due to weak key size	Stronger than DES but computationally expensive	Modern encryption standard, secure and efficient	Used in PGP encryption, balanced speed and security	Flexible encryption with variable key sizes, fast processing
Weaknesses/Gaps	Vulnerable to brute-force attacks, weak key size, outdated	Slower due to multiple encryption rounds, still vulnerable to some attacks	Susceptible to side-channel attacks (e.g., timing attacks), requires large key sizes for maximum security	Weak key scheduling, vulnerable to specific attacks like chosen plaintext attacks	Small block size (64 bits), slow performance with large key sizes, vulnerable to related-key attacks

2.3.3 Asymmetric Encryption Algorithms

Asymmetric cryptographic methods, such as RSA and ECC, utilize distinct public and private keys for encryption and decryption (Alegro et al., 2019). These techniques show strong security, but their increased computational complexity makes them inoperable for real-time image encryption applications. RSA, for instance, requires large prime numbers for key generation, making it computationally expensive compared to symmetric encryption methods.

Table 2.5: Comparison of Different Asymmetric-Key Cryptography Algorithms

Properties	RSA	ElGamal	ECC
Key Size (Kumar Sharma et al., 2021)	1024 bits (commonly 2048+ for security)	1024 bits (can be larger)	160 bits (equivalent to 1024-bit RSA)
Structure (S. Kumar et al., n.d.; Kumar Sharma et al., 2021)	Integer Factorization	Discrete Logarithm	Elliptic Curve
Developed By (Kumar Sharma et al., 2021)	Rivest, Shamir, and Adleman, 1977	Taher ElGamal, 1985	Neal Koblitz & Victor S. Miller, 1985
Aim (S. Kumar et al., n.d.; Kumar Sharma et al., 2021)	Uses two keys (public and private) for encryption and decryption. Security relies on the difficulty of factoring large prime numbers. Efficient for large data encryption but computationally intensive.	Based on the discrete logarithm problem, decryption is challenging without the private key. It produces long ciphertexts and is slower than RSA.	It uses elliptic curve mathematics to provide strong security with smaller key sizes. Efficient and secure but complex to implement.
Security (Kumar Sharma et al., 2021; Perna et al., 2013)	Secure with large key sizes but vulnerable to quantum attacks.	Secure under current cryptanalysis, but ciphertext expansion reduces efficiency.	Strong security even with smaller key sizes, but requires careful implementation.
Limitations	Requires large key sizes (2048+ bits) for strong security, which increases computational cost. - Slower compared to symmetric encryption. - Vulnerable to quantum computing (Shor's algorithm).	- Slower than RSA due to large ciphertext size. - Ciphertext expansion makes it inefficient for encrypting large amounts of data. - Computationally expensive.	- More complex to implement correctly. - Requires specialized knowledge in elliptic curve mathematics. - Vulnerable to poorly chosen curves (weak curve attacks).

Table 2.5 compares RSA, ElGamal, and ECC (Elliptic Curve Cryptography) in the context of asymmetric cryptography, focusing on key size, design, and security features. RSA relies on integer factorization and requires large key sizes, making it computationally intensive. ElGamal uses discrete logarithms but is slower due to ciphertext expansion. In contrast, ECC (Elliptic Curve Cryptography) provides strong security with smaller key sizes and greater efficiency, though it is more complex to implement.

Table 2.6: Comparison of Chaotic Image Encryption and Traditional Methods

Criteria	Traditional (AES/DES/RSA)	Basic Chaotic (Logistic, Sine, Cubic, Tent Maps) Lena	Hyperchaotic (Lorenz, Rossler, Chen Systems)
Key Space	AES: 2^{128} , 2^{192} , 2^{256} (Hamza & Kumar, 2020) DES: 2^{56} (Paradesi Priyanka et al., 2022) RSA: 2^{1024} (Imam et al., 2021)	Parameters (r, x ₀) can each provide 10^{15} values, resulting in a total key space of 10^{30} (Ahuja & Doriya, 2023; Pareek et al., 2006; X. Wang & Guan, 2023). Logistic Map: $\approx 10^{30}$ Sine Map: $\approx 10^{30}$ Cubic Map: $\approx 10^{30}$ Tent Map: $\approx 10^{30}$	Lorenz: 2^{232} (C. Zou et al., 2020) Rossler+: 2^{362} (W. Huang et al., 2021) Chen+: 2^{256} (H. Gao & Wang, 2021)
Entropy	AES: 7.959 (Science & Phenomena, 2017) DES: 7.996 (Mohammad et al., 2017) RSA: 7.965 (Gollagi et al., 2021)	Logistic Map: 7.9666 (Mandal et al., 2012) Sine Map: 7.9635 (Gebereselassie & Roy, 2024) Cubic Map: 7.9653 (Gebereselassie & Roy, 2024) Tent Map: 7.9679 (Gebereselassie & Roy, 2024)	Lorenz: 7.9991 (C. Zou et al., 2020) Rossler+: 7.90238 (W. Huang et al., 2021) Chen+: 7.9991 (H. Gao & Wang, 2021)
Image Encryption NPCR/UACI	AES: 99.63/ 31.87(Science & Phenomena, 2017) DES: 0.0216/0.0040(Mohammad et al., 2017) RSA: 99.70/33.30(Gollagi, 2021)	Logistic Map: 99.625/28.332 (Mandal et al., 2012) Sine Map: 95.3903/33.3676(Gebereselassie & Roy, 2024) Cubic Map: 99.510/37.932 (Gebereselassie & Roy, 2024) Tent Map: 97.417/33.423 (Gebereselassie & Roy, 2024)	Lorenz: 99.6246/33.5118 (C. Zou et al., 2020) Rossler+: 99.63/33.41(C. Zou et al., 2020) Chen+: 99.61/33.42(H. Gao & Wang, 2021)
Computational Speed	AES: Fast(Dhanalaxmi, 2017) DES: Slow(Dhanalaxmi, 2017) RSA: Slowest(Dhanalaxmi, 2017)	Efficient for image data (Aouissaoui et al., 2022).	Lorenz system to meet the need for fast computing and quantum encryption (Arshad et al., 2019). Relatively low speed (H. Gao & Wang, 2021; W. Huang et al., 2021).
Attack Resistance	RSA algorithm attacks are more prone in terms of data security (Imam et al., 2021). The time required for a brute-force attack increases proportionally with the number of rounds needed to find a key (Mohammad et al., 2017). AES has a Side-channel attack.	Moderate resistance; some weaknesses due to low-dimensional predictability (L. Liu & Miao, 2018).	High resistance to statistical and chosen-plaintext attacks (C. Zou et al., 2020). Resist multiple attacks(W. Huang et al., 2021). Chen's system can effectively resist ciphertext-only attacks, differential attacks, cropping attacks, and noise attacks (H. Gao & Wang, 2021).
Sensitivity	The AES algorithm is susceptible to noise. So, AES is not suitable for noisy environments (Ahmad & Ahmed, 2012).	High sensitivity (small key changes yield significant output differences) (Pareek et al., 2006).	Extreme sensitivity due to multiple positive Lyapunov exponents (H. Gao & Wang, 2021; Q. Zou et al., 2023).
Encryption Time Complexity	AES: 312KB 1.8Sec.(Padmavathi & Kumari, 2013) DES: 312KB 3.0Sec.(Padmavathi & Kumari, 2013) RSA: 312KB, 7.8Sec.(Padmavathi & Kumari, 2013)	Logistic Map: 6.26–6.32 sec(1024x1024) (Mandal et al., 2012). Sine Map: 0.017564sec(Zhong et al., 2018) Logistic Map: 0.6436sec(Zhong et al., 2018)	It often requires additional synchronization overhead (Alibraheemi et al., 2022). High time complexity (W. Huang et al., 2021).
Hardware Implementation	AES: Faster and Efficient DES: Better in hardware than in software. RSA: Not very efficient (Padmavathi & Kumari, 2013).	It may require custom design and optimization for efficient hardware use (Aouissaoui et al., 2022).	Requires specialized design (FPGA/ASIC) for real-time performance (Peng & Zeng, 2020). Using Lorenz-256x256-192KB-953mS
Resistance to Brute-Force	DES is vulnerable to brute-force attacks with weak keys, while AES (Paradesi Priyanka et al., 2022) and RSA are more resistant (Imam et al., 2021).	Generally lower due to a smaller key space(Aouissaoui et al., 2022)	The keyspace is sufficiently large to resist brute force attacks (C. Zou et al., 2020)(H. Gao & Wang, 2021).

2.4 Chaos-Based Encryption

Standard cryptographic algorithms such as RSA, DES, and AES are insufficient for image encryption. Chaos-based cryptographic schemes use chaotic maps' sensitivity to initial conditions and pseudo-randomness to generate complex encryption keys, offering enhanced security and a larger key space than traditional methods (Alexan, Gabr, et al., 2023; Jiang et al., 2023; M. Zhang et al., 2020).

One-dimensional chaotic maps are often used in security applications due to their simplicity and chaotic properties. However, one-dimensional chaotic maps, such as the Logistic, Sine, Cubic, and Tent maps, exhibit predictability because of their low-dimensional characteristics (Aouissaoui et al., 2022; X. Wang et al., 2020).

Hyperchaotic systems, such as the Lorenz, Rossler, Chen, Lu, attractors, introduce additional positive Lyapunov exponents, which enhance key generation complexity and strengthen cryptographic systems (Q. Liu & Liu, 2020). Traditional cryptographic methods like DES and RSA have shown vulnerabilities in penetration testing. Although AES and RSA offer strong security, they may not be suitable for all applications due to high resource demands, particularly for low-power devices. For more information, see Table 2.6.

2.4.1 Categorization of Chaotic Systems for Image Security

Chaotic encryption techniques differ in complexity and security. One-dimensional chaotic maps show basic pixel permutation but have a limited key space (X. Wang et al., 2020). In contrast, multidimensional hyperchaotic systems, such as the Lorenz and Rössler systems, offer superior protection (V. Kumar & Girdhar, 2021). Fractional-order hyperchaotic systems enhance unpredictability and key sensitivity, resulting in higher entropy, particularly when combined with memristors, thereby improving cryptographic resistance. As presented in Table 2.7, fractional-order systems provide enhanced security; however, they require hardware optimization.

Table 2.7: Categorization of Chaotic Systems

References	Chaotic System	Strengths	Weaknesses
(Aouissaoui et al., 2022; L. Liu & Miao, 2018; X. Wang et al., 2020)	1D Maps (Logistic)	Simple implementation	Predictable, small key space
(V. Kumar & Girdhar, 2021)(C. Zou et al., 2020)(R. Lin & Li, 2021)	Hyperchaotic (Lorenz)	Multiple Lyapunov exponents	Computational complexity
(Alexan, Gabr, et al., 2023; Dong et al., 2020; Feng et al., 2023; Rahman et al., 2022)	Fractional-Order	Higher entropy (>7.9)	Requires advanced hardware

2.4.2 Memelements in Hyperchaotic Cryptography

Memristors, memcapacitors, and meminductors, known as memelements, have transformed cryptographic applications with their memory-dependent nonlinear dynamics. These elements bring unique properties that enhance hyperchaotic behaviour, making encryption schemes more secure.

Memelements have transformed cryptographic applications with their memory-dependent dynamics, enhancing hyperchaotic behavior and making encryption schemes more secure. Memelements enhance cryptography by improving key sensitivity, enabling hyperchaotic systems, boosting entropy (over 7.9) for better randomness, and supporting compact, low-power hardware.

2.4.3 Generation of Memelements-Based Hyperchaotic Systems

Hyperchaotic systems are characterized by multiple positive Lyapunov exponents, which lead to highly unpredictable behaviour. Memristor-based hyperchaotic systems can typically be designed in 2D, 3D, 4D, 5D, or 6D dimensions (D. Huang et al., 2024; Kou et al., 2022; Mezzatio et al., 2020; J. Sun et al., 2023; J. Zhang et al., 2024; S. Zhang et al., 2023). Memelements enhance these chaotic dynamics by utilizing memory-dependent differential equations. These systems must show at least two positive Lyapunov exponents, ensuring their hyperchaotic properties. Likewise, the memory characteristics of these systems improve entropy and randomness (Cun et al., 2023; H. System & Filtering, 2020).

Research indicates that memristor-based systems generate more complex attractors than traditional chaotic maps, making them particularly suitable for encryption applications (Kengne et al., 2017; D. Yan et al., 2021). Refer to Table 2.8 for an assessment of security metrics in memelements-based image encryption, including its resistance to cryptographic attacks.

2.4.4 Image Encryption Using Memelements-Based Hyperchaotic Systems

The encryption process begins with key generation using a hyperchaotic system that produces pseudo-random sequences for encryption keys, enhanced by mem-elements for improved sensitivity. In the permutation stage, image pixels are scrambled with these hyperchaotic sequences to disrupt correlations. The diffusion stage modifies pixel values through DNA encoding and chaotic operations, ensuring high randomness. Finally, decryption reverses the encryption using the duplicate chaotic keys (Sha et al., 2021; J. Sun et al., 2020; Yildirim, 2020).

Table 2.8 shows that memcapacitor-based and combined memristor encryption methods outperform traditional hyperchaotic techniques. An entropy value near 8.0 indicates excellent randomness, and NPCR and UACI values of 99.6% and 33.5% demonstrate strong resistance to differential attacks. A correlation coefficient close to zero further emphasises the effective decorrelation between the original and encrypted images, highlighting the robustness of these advanced techniques.

Table 2.9 outlines how memelement-based hyperchaotic encryption improves security, randomness, and resistance to attacks. Memristor encryption is the most studied, displaying high entropy and key sensitivity. While memcapacitor and meminductor encryption have benefits, they need more optimization. Combining all three methods offers superior security, though fabrication challenges complicate implementation.

Table 2.8: Security Performance Metrics for Memelements Image Encryption

Metric	Memristor-Based	Memcapacitor-Based	Meminductor-Based	Combined Memelements-Based		
				Dual-Memristors	Memristor-Meminductor	Another Combination
Entropy	7.9991(Y. Gao, Liu, Zhang, et al., 2024)	7.9996(F. M. H. System et al., 2022)	It has not been addressed yet.	7.9993(S. Zhang et al., 2023)	7.9997(Lan & Wang, 2025)	It has not been addressed yet.
Key Sensitivity	High (Y. Gao, Liu, Zhang, et al., 2024)	High (F. M. H. System et al., 2022)	It has not been addressed yet.	High (S. Zhang et al., 2023)	Extremely High(Lan & Wang, 2025)	It has not been addressed yet.
NPCR (%)	99.63 (Y. Gao, Liu, Zhang, et al., 2024)	99.814 (F. M. H. System et al., 2022)	It has not been addressed yet.	99.6043(S. Zhang et al., 2023)	99.606(Lan & Wang, 2025)	It has not been addressed yet.
UACI (%)	33.36 (Y. Gao, Liu, Zhang, et al., 2024)	33.625 (F. M. H. System et al., 2022)	It has not been addressed yet.	33.4668(S. Zhang et al., 2023)	33.5(Lan & Wang, 2025)	It has not been addressed yet.
Correlation Coefficient	~0.002 (Y. Gao, Liu, Zhang, et al., 2024)	~0.0004 (F. M. H. System et al., 2022)	It has not been addressed yet.	~0.0009(S. Zhang et al., 2023)	~0.01(Lan & Wang, 2025)	It has not been addressed yet.
Resistance to Differential Attacks	Highly resistant to differential attacks (Y. Gao, Liu, Zhang, et al., 2024)	Demonstrate strong resistance to various types of attacks (F. M. H. System et al., 2022)	It has not been addressed yet.	It is resilient against correlation, statistical, differential, data loss, and noise attacks (S. Zhang et al., 2023).	Resilience against differential, brute force, statistical attacks, cropping, noise, chosen-plaintext attacks, and combined attacks (Lan & Wang, 2025).	It has not been addressed yet.

Table 2.9: Tabulated Review of Memelements-Based Image Encryption

References	Encryption Scheme	Key Features	Security Enhancements	Challenges
(Z. Liu et al., 2019; Raj et al., 2021)	Memristor-Based Encryption	Non-volatile memory, history-dependent behavior	High entropy, excellent key sensitivity	Complex fabrication, variability issues
(Romero et al., 2021; J. Sun et al., 2020)	Memcapacitor-Based Encryption	Charge-storage properties, nonlinear response	Reduces power consumption, enhances randomness	Limited research, stability concerns
(Marco et al., 2021; Romero et al., 2021)	Meminductor-Based Systems	Magnetic flux memory, high-frequency response	Improves frequency-domain security characteristics	Difficult fabrication, underexplored
(An et al., 2021; Çam Taşkıran et al., 2020; Lan & Wang, 2025)	Combination of Memristor, Memcapacitor, and Meminductor	Hybrid memory effects, maximum nonlinearity	Synergistic security improvements, ultra-complexity	Manufacturing and integration challenges

2.5 DNA-Based Encryption

DNA cryptography, inspired by biological encoding, uses nucleotide transformations (A, T, C, G) to enhance encryption (Shakir et al., 2022). The combination of dynamic DNA coding and hyperchaotic maps improves diffusion, achieving NPCR over 99% and UACI above 33% (Z. Liu et al., 2019). Additionally, dynamic DNA coding addresses static vulnerabilities, though the potential of meminductors remains untapped (X. Chen et al., 2023; Cun et al., 2023; Z. Liu et al., 2019).

2.6 Deep Learning-Driven Security Enhancements

Machine learning techniques, such as convolutional neural networks (CNNs) and autoencoders, help improve encryption efficiency and security (Guang et al., 2023; F. Hu et al., 2016; Velho, 2009; Y. Yang et al., 2018). CNNs enable quicker real-time encryption with image compression, while extreme learning machine (ELM) autoencoders enhance resilience against noise for reliable decryption. Combining these methods creates a more effective encryption framework (F. Hu et al., 2016; Man et al., 2021; D. Wu & Wang, 2021; J. Zhang et al., 2020).

A technique that combines a Convolutional Neural Network (CNN) with an Extreme Learning Machine (ELM) achieves 97.50% accuracy, surpassing that of state-of-the-art methods (S. Zhou & Tan, 2020). Compression, encryption, and denoising enhance data security and efficiency by reducing storage needs and protecting against unauthorised access (Charmouti et al., 2022; Mahdaoui et al., 2022; Pathak et al., 2024).

AI-driven encryption techniques for IoT and smart devices enhance efficiency and scalability through the use of machine learning. Lightweight convolutional neural network models enable real-time encryption with low computational demands, making them suitable for resource-constrained environments.

These solutions enhance security and facilitate rapid data transmission. Figure 2.16 illustrates a deep learning encryption framework that integrates Convolutional Neural Networks (CNNs) with Extreme Learning Machine (ELM) Autoencoders.

The CNNs provide adaptive image compression for faster encryption and reduced data size, while the ELM Autoencoders enhance decryption with improved denoising. This approach boosts security and efficiency, making it ideal for resource-constrained IoT and smart devices.

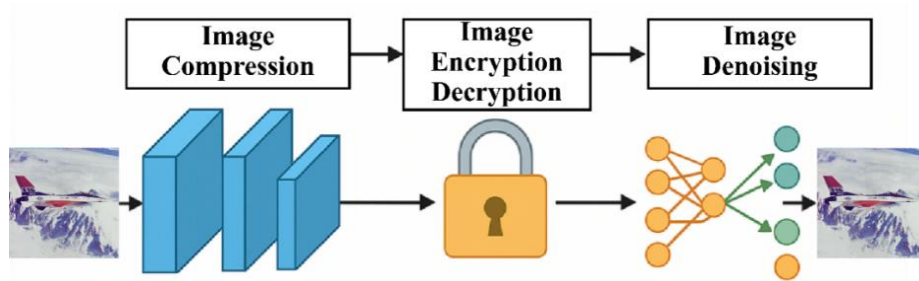


Figure 2.16: Deep Learning-Driven Image Security Pipeline

2.7 Security Analysis Criteria in Image Encryption

Table 2.10 provides a comparative analysis of encryption performance metrics for the Lena image, including minimum values, literature-reported values, ideal targets, and notable remarks. The key parameters correlation, entropy, keyspace, MSE, PSNR, SSIM, MAE, NPCR, and UACI are vital for assessing the security and effectiveness of the encryption scheme.

Encrypted images must withstand standard cryptanalysis techniques, including statistical analysis, chosen-plaintext attacks, and known-plaintext attacks. Chaos-based methods, especially hyperchaotic systems, exhibit strong resistance due to their inherent unpredictability. See Table 2.11 for a summary of security metrics, highlighting encryption effectiveness against various attacks.

Chaos-based and AI-driven encryption offers strong security, but standardisation is vital for reliability. Key metrics like NPCR (over 99%), UACI (around 33%), and entropy (about 8) help evaluate strength, with AI methods typically outperforming traditional AES. Standard benchmarks are essential for enhancing security and consistency.

Table 2.10: Encryption Parameters for Lena Original and Encrypted Images

References	Parameter	Minimum	Optimum	Remarks (Ideally)
(X. Zhang & Ye, 2021)	Correlation	0.0065(GV)	0.001	0
(Alexan et al., 2022)	Entropy	7.9973(G)	7.99	8
(D. Lin et al., 2022)	Keyspace	2^{100}	2^{500}	Infinite
(Alexan, Chen, et al., 2023)	MSE	8890.05	9000	Large
(Alexan, Chen, et al., 2023)	PSNR	8.64176	8.8	Low
(Theramban & Ravi, 2021)	SSIM	0.001	0.0009	0
(Alexan et al., 2022)	MAE	77.3752	80	Large
(Lawnik & Berezowski, 2022)	NPCR	99.23%	99.40%	100%
(Lawnik & Berezowski, 2022)	UACI	32.60%	32.80%	33%

Table 2.11: Security Analysis Metrics

Metric	Ideal Value	Traditional	Chaos-Based	AI-Driven-Chaos	References
Entropy	8	7.959 (AES)	7.9981 (memristor)	7.9994 (CNN)	(Erkan et al., 2022; Njitacke et al., 2022; Pathak et al., 2024; Science & Phenomena, 2017)
NPCR (%)	> 99	99.63 (AES)	99.8173 (memristive)	99.6094 (AI-DNA hybrid)	(Erkan et al., 2022; Njitacke et al., 2022; Pathak et al., 2024; Science & Phenomena, 2017)
UACI (%)	~33	31.87 (AES)	33.437 (memristive)	33.4622(AI-DNA hybrid)	(Erkan et al., 2022; Njitacke et al., 2022; Pathak et al., 2024; Science & Phenomena, 2017)
Key Sensitivity	High	Moderate	Moderate – High	High	(Erkan et al., 2022; Njitacke et al., 2022; Science & Phenomena, 2017)

2.8 Related Work and Identified Research Gaps

Table 2.12 summarizes image encryption techniques like chaotic methods (logistic, Hénon, Lorenz maps), DNA encoding, and deep neural networks. While these methods enhance security, they have vulnerabilities such as limited key space and slow execution times, indicating a need for further research to improve randomness and efficiency, especially with wavelet transforms and hyperchaos.

This study addresses several limitations of conventional encryption algorithms, including longer execution times, low security, vulnerability to cyberattacks, low entropy values, limited key space in Logistic chaotic mapping, and weaknesses in fixed DNA encoding. Additionally, there is a lack of a practical trapdoor function and randomness issues with pixel shuffling. To tackle these challenges, the recommendation suggests using memelements-based hyperchaotic systems combined with DNA, CNN models, and Autoencoders.

Table 2.12: Related Literature Review Summaries

Author	Title	Contribution	Gap
Abdullah A. Rashid, Khalid Ali Hussein Year: 2023. (Rashid & Hussein, 2023)	“Image encryption algorithm based on the density and 6D logistic map”	Sharing and exchanging multimedia with privacy and security by using cryptographic approaches based on chaotic logistic Image encryption methods.	Traditional block ciphers are not suitable for multimedia data.
X. Gao Year: 2021 (X. Gao, 2021)	“A color image encryption based on an improved Hénon map”	DES and AES, two common text encryption techniques, provide enough security.	Conventional encryption algorithms like DES, AES, and RSA can't produce ideal results.
G. Arthi, V. Thanikaiselvan, and R. Amirtharajan Year: 2022. (Arthi et al., 2022)	“4D Hyperchaotic map and DNA encoding combined image encryption”	Uses a cryptosystem that combines 4D Lorenz-type hyper-chaos and DNA encoding mechanisms for enhanced encryption.	DES, AES, and IDEA have limitations like longer execution time, low security, and various cyberattacks.
Q. Cai, H. Li, W. Cao, and Guifen Jie Year: 2023. (Q. Cai et al., 2021)	“Photoinduced-electron transfer coupled with DNA cross-chain displacement amplification”	Uses multiple DNA computing approaches and DNA biological operations during encryption.	Weaknesses in fixed DNA encoding that can be easily deciphered.
Z. Liang, Q. Qin, C. Zhou, and S. Xu Year: 2023. (Z. Liang et al., 2023a)	“Color Image Encryption Algorithm Based on Four-Dimensional Multi-stable Hyper Chaotic System”	The algorithm uses a basic one-dimensional Logistic chaotic mapping.	Small key space and susceptibility to exhaustive attacks.
P. N. Lone, D. Singh, and U. H. Mir Year: 2022. (P. N. Lone et al., 2022)	“Image encryption using DNA coding and three-dimensional chaotic systems”	Introduces a simple image encryption method using 3D maps and DNA operations.	Future work should include wavelet transforms and hyperchaos to improve speed and efficiency.
J. Yu, X. Yu, L. Zhang, and W. Xie Year: 2023. (J. Yu et al., 2023)	“A two-stage chaotic encryption algorithm for color face image based on circular diffusion”	Proposes a hyperchaotic algorithm with good speed and security analysis.	Needs development of real-time and secure face encryption algorithms for videos.
P. Li, J. Xu, J. Mou, and F. Yang Year: 2019. (P. Li et al., 2019)	“Fractional-order 4D hyperchaotic memristive system and application in color image encryption,”	It shows increased complexity and randomness and is suitable for encryption.	Entropy values are relatively low.

Author	Title	Contribution	Gap
S. R. Maniyath and T. V Year: 2020. (Maniyath & Thanikaiselvan, 2020)	“An efficient image encryption using deep neural network and chaotic map”	Combines chaos and NNs, using autoencoders to enhance performance and key generation.	It lacks an effective trapdoor function and has unresolved security problems.
H. Torabi, S. L. Mirtaheri, and S. Greco Year: 2023. (Torabi et al., 2023)	“Practical autoencoder-based anomaly detection by using vector reconstruction error”	Develops an AE-based model for anomaly detection in cloud computing.	May have limitations in practical use.
N. Gupta and R. Vijay Year: 2022. (Gupta & Vijay, 2022)	“Hybrid image compression-encryption scheme based on multilayer stacked autoencoder and logistic map”	Combines a stacked autoencoder and a logistic map for image encryption.	The randomness of pixel shuffling based on positions is limited.

The literature review in Table 2.13 covers Memelements, Memelements-based hyperchaotic systems, chaotic systems, DNA coding, system optimization, and autoencoder-based denoising in image encryption and cryptography.

Key highlights include Memristor-Based Hyperchaotic Systems for Encryption (An et al., 2021; Qian et al., 2023; Singh, 2024; B. Wang et al., 2018; G. Yang et al., 2023), Chaotic Systems for Image Encryption (Feng et al., 2022; Kengnou Telem et al., 2021; H. Li et al., 2017; Y. Liu et al., 2020; Susanto et al., 2020; Wan et al., 2020), DNA Coding and Chaotic Systems (Aditya et al., 2020; Alawida & Teh, 2023; Kengnou Telem et al., 2021; Patel et al., 2020; Wan et al., 2020; Xuejing & Zihui, 2020), Optimization in Cryptography Using PSO and Chaotic PSO (Fang et al., 2021; Gad, 2022; Houssein et al., 2021b; Y. Sun et al., 2009; Tutueva et al., 2020), Online Extreme Learning Machine (OSELM) and Autoencoders for Denoising (Al-Amri et al., 2023; Fang et al., 2021; Gebereselassie & Roy, 2022; He et al., 2021), and Hybrid-Cryptographic Systems (Al-Amri et al., 2023; Heltberg et al., 2019; Iqbal et al., 2021; Wen et al., 2022; J. Yu et al., 2023), Advancements in these areas aim to enhance image encryption.

Table 2.13: Summarized Literature Reviews on Advanced Image Encryption Techniques

References	Category	Summary	Identified Problems	Proposed Solutions
(An et al., 2021),(Qian et al., 2023), (G. Yang et al., 2023),(Singh, 2024)	Memristor-Based Systems	Application of memristor-based systems for encryption using hyperchaotic dynamics.	Conventional systems lack adequate randomness.	Use hyperchaotic memristors to improve non-linear dynamics and encryption strength.
(B. Wang et al., 2018),(Singh, 2024), (An et al., 2021),(Qian et al., 2023), (G. Yang et al., 2023)	Memristor-based Hyperchaotic Systems	Combine memristors with chaotic dynamics for robust encryption.	Lack of sufficient randomness in conventional systems.	Enhance randomness and cryptographic security with hyperchaotic memristors.
(Kengnou Telem et al., 2021),(H. Li et al., 2017), (Wan et al., 2020),(Susanto et al., 2020), (Y. Liu et al., 2020),(Kengnou Telem et al., 2021), (Feng et al., 2022),(Y. Liu et al., 2020), (Susanto et al., 2020)	Chaotic Systems	Use of chaotic maps like logistic and tent maps.	High correlation and vulnerability in traditional encryption.	Use chaotic systems to increase randomness and reduce predictability.
(Kengnou Telem et al., 2021),(Xuejing & Zihui, 2020), (Patel et al., 2020),(Aditya et al., 2020), (Wan et al., 2020),(Alawida & Teh, 2023)	DNA Coding and Chaotic Systems	Combine DNA encoding with chaotic maps for pixel scrambling.	Predictability and weak resistance in static methods.	Use dynamic encoding and higher-dimensional maps.
(Tutueva et al., 2020),(Gad, 2022), (Y. Sun et al., 2009),(Fang et al., 2021), (Houssein et al., 2021b)	Optimization in Cryptography	Apply PSO and CPSO for optimization in encryption.	Standard PSO can converge prematurely.	CPSO adds chaotic diversity, improving convergence and encryption performance.
(Fang et al., 2021),(He et al., 2021), (Gebereselassie & Roy, 2022),(Al-Amri et al., 2023)	OSELM & Autoencoders	Use OSELM with autoencoders for cryptographic denoising.	ELM shows slow convergence in complex tasks.	OSELM speeds up convergence and robustness in denoising.
(Iqbal et al., 2021),(Wen et al., 2022), (J. Yu et al., 2023),(Heltberg et al., 2019), (Al-Amri et al., 2023)	Hybrid Chaotic and Cryptographic Systems	Combine chaotic maps, DNA coding, memristors, and PSO.	AES and DES lack robustness for image data.	Use hybrid models for high randomness, security, and speed.

Challenges like low randomness and slow PSO convergence remain. Future research should explore hybrid models combining Chaotic PSO with memristor chaos, DNA coding, and optimised neural networks to enhance cryptographic security and efficiency.

2.9 Gaps in Current Image Encryption Research

Despite advancements in image encryption, several challenges hinder the creation of secure and efficient systems for multimedia data. Key limitations and areas needing further research include:

- Traditional encryption algorithms, such as DES, AES, and RSA, are ineffective for multimedia applications due to their inability to efficiently process large files and their susceptibility to multimedia-specific attacks (Rashid & Hussein, 2023).
- Chaotic systems employed in encryption, especially low-dimensional chaotic maps, often have limited key spaces and low entropy. These limitations increase susceptibility to brute-force and statistical attacks. To enhance randomness and security, there is a need for high-dimensional or multi-stable chaotic systems (Z. Liang et al., 2023a; X. Wang et al., 2020).
- DNA-based encryption schemes often rely on fixed encoding patterns, which reduces randomness and makes them easier to decipher. Dynamic and adaptive DNA encoding mechanisms are needed to enhance security (Cai et al., 2021).
- Execution speed is another concern, as traditional cyphers and basic chaotic systems can be slow, particularly with high-resolution images, thus limiting real-time application (Lone et al., 2022).
- Additionally, many encryption algorithms lack robustness against common attacks, such as noise interference and cropping, leaving images vulnerable to tampering (Zhang et al., 2023). Deep neural network-based approaches often lack secure trapdoor functions, which complicates key management (Maniyath & Thanikaiselvan, 2020).
- Pixel shuffling techniques also fall short, as simple shuffling may not eliminate spatial redundancy, resulting in exploitable patterns (Gupta & Vijay, 2022).
- Ultimately, the emergence of quantum computing presents new risks that many existing encryption systems cannot withstand. Incorporating quantum-resistant techniques is vital for long-term security (Torabi et al., 2023).

In summary, image encryption research must address key issues, including inefficiencies in traditional cyphers, security limitations of low-dimensional chaotic systems, the predictability of fixed DNA encoding, and the latency of current methods. Additionally, improvements are needed in trapdoor mechanisms for deep learning, pixel-shuffling techniques, and resistance to quantum attacks. Addressing these challenges is crucial for developing secure next-generation image encryption frameworks.

2.10 Final Summary of the Literature Review

Conventional encryption algorithms, such as DES, AES, and RSA, are not effective for image encryption due to high pixel correlation and large file sizes, which expose them to vulnerabilities in real-time environments like 5G networks. One-dimensional chaotic maps, while offering alternatives, often have limited complexity, resulting in smaller key spaces and lower entropy, which makes them susceptible to attacks. As a result, research is shifting towards high-dimensional and hyperchaotic systems, especially those utilizing mem-elements (memristors, memcapacitors, and meminductors) to enhance security and nonlinearity.

DNA coding provides inherent parallelism and randomness for image encryption; however, many models rely on static encoding rules, which reduce unpredictability. Traditional compression techniques, such as JPEG, fail for encrypted images where fidelity is crucial, particularly in medical or military contexts. Deep learning, particularly Convolutional Neural Networks (CNNs) and Autoencoders, has shown promise for adaptive compression and denoising but is underutilized in secure encryption frameworks.

Many encryption schemes also lack resilience against common attacks and often do not provide comprehensive performance metrics, such as PSNR, MSE, NPCR, UACI, and SSIM. Optimization methods, such as Particle Swarm Optimisation (PSO), have been applied to improve parameter tuning; however, challenges like premature convergence remain.

Current research in image encryption encounters several challenges, including fragmentation and key limitations, such as weak randomness and insufficient performance evaluation. Future efforts should focus on developing unified frameworks that integrate hyperchaotic memory elements, dynamic DNA coding, and deep learning. This approach will ensure rigorous benchmarking and quantum resistance, ultimately enhancing the security of multimedia communications.

CHAPTER 3

MEMELEMENTS AND THE DEVELOPMENT OF CHAOTIC SYSTEMS

3.1. Introduction:

Chaos theory has gained significant attention in various scientific and engineering areas due to its applications in secure communication, cryptography, nonlinear systems, and intricate dynamical systems. At the core of chaotic scenarios lies nonlinearity, which can be introduced through memelements such as memristor, memcapacitor, and meminductor. These elements exhibit memory-dependent characteristics, allowing them to accumulate and process information dynamically and be suitable for producing advanced chaotic and hyperchaotic systems. This chapter provides an overview of chaos theory and it highlights the role of memelements such as memristors, memcapacitors, and meminductors in introducing nonlinearity and memory-dependent dynamics. The objectives of this chapter are to explore memelements, design hyperchaotic systems, analyze their dynamics, and present novel hybrid one-dimensional chaotic maps.

3.2. Fundamentals of Memelements

Memelements, including memristors, memcapacitors, and meminductors, are nonlinear circuit components with memory-dependent properties influenced by past inputs, such as voltage and charge (Kebbat, 2022; Raj et al., 2021). They enhance classical circuit theory by showing hysteresis and state retention, making them valuable in neuromorphic computing, adaptive systems, secure communication, and energy-efficient memory applications.

These components introduce memory-dependent nonlinearity by extending traditional concepts of resistors, capacitors, and inductors. Figure 3.1 shows the interrelationships between the memristor, memcapacitor, and meminductor, highlighting their basic equations. It indicates how the state variables (ρ , σ) relate to charge (q), flux (ϕ and p), current (i), and voltage (v) through equations such as $d\rho = M dq$ for the memristor, $d\sigma = M d\phi$ for the memcapacitor, and $d\phi = M dq$ for the meminductor, where M represents memory-dependent resistance, capacitance, or inductance. These equations highlight the dynamic behavior of these components.

The mathematical formulation of the flux-controlled memristor is found in the relationship between current $i(t)$ and voltage $u(t)$, controlled by the memductance $W(\phi(t))$ (Kebbat, 2022).

2022; X. Wang et al., 2019). The memductance is represented as a function of the flux $\phi(t)$, derived from a nonlinear charge function $q(\phi)$. The memductance model is formulated as:

$$W(\phi(t)) = m_1 + 3m_2\phi^2(t) \quad (3.1)$$

where m_1 and m_2 are memductance constants defining the device's nonlinearity. This quadratic dependence on flux $\phi(t)$ (where $\phi(t) = \int u(t)dt$) ensures pinched hysteresis under periodic excitation, a hallmark of memristive systems. The memristor exhibits a pinched hysteresis loop in the q - u plane, as shown in Figure 3.2(a), when excited with a sinusoidal charge input $q(t) = Q_C \sin(2\pi ft)$.

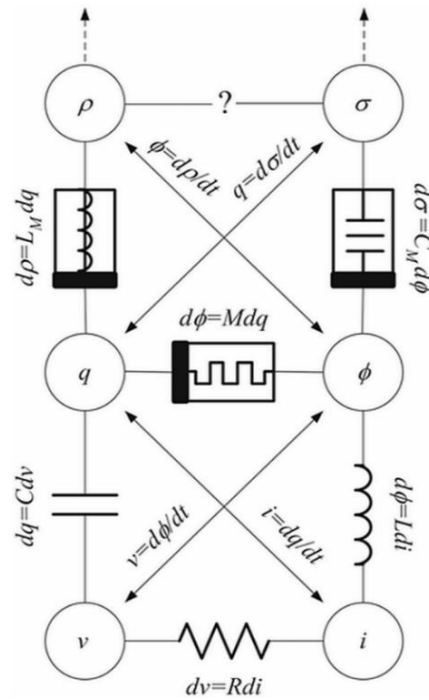


Figure 3.1: Schematic of the Electrical Element Relations (Adapted Ref. (Petráš, 2022))

3.2.1. Memcapacitor Model

For a charge-controlled memcapacitor (Kebbaty, 2022; X. Wang et al., 2019), the voltage $u(t)$ depends nonlinearly on the charge $q(t)$ and its time integral σ . The voltage is represented as:

$$u(t) = (\alpha + \beta\sigma^2)q(t) \quad (3.2)$$

Where $\sigma = \int q(t)dt$ represents the charge's integral, and α and β are parameters characterizing the device's nonlinear behavior. Under sinusoidal excitation $q(t) = Q_C \sin(2\pi ft)$ ($Q_C = 1C, f = 1Hz$), the $q - u$ hysteretic loops (Figure (3.2b)) validate the model's ability to replicate memory-driven capacitive effects.

3.2.2. Meminductor Model

The flux-controlled meminductor relates current $i(t)$ to flux $\phi(t)$ and its integral ρ :

$$i(t) = (\alpha' + \beta'\rho)\phi(t) \quad (3.3)$$

where $\rho = \int \phi(t)dt$ is the integral of the flux, and α' and β' are parameters specific to the meminductor. This equation illustrates the dependence of the current on both the flux and its integral, capturing the nonlinear characteristics of the meminductor (Kebbat, 2022; X. Wang et al., 2019). When excited by $\phi(t) = Q_L \sin(2\pi ft)$ ($Q_L = 1Wb, f = 1Hz$), the $\phi - i$ pinched hysteresis loop (Figure (3.2c)) confirms its memory-inductive properties.

This comparative analysis of the memristor, memcapacitor and meminductor under sinusoidal excitation highlights their respective roles in modeling hysteretic behaviors within electronic circuits. Figure 3.2 illustrates the voltage-charge and current-flux hysteresis loops for three types of Memelements: Memristor, Memcapacitor, and Meminductor. Each plot illustrates how the respective element's behaviour varies with varying parameters (Q_C for Memristor and Memcapacitor, Q_L for Meminductor), highlighting their response's non-linear and history-dependent nature.

The Memristor's resistance changes with accumulated charge, the Memcapacitor's capacitance relies on the voltage applied, and the Meminductor's inductance is controlled by the flux, all of which direct to different hysteresis characteristics that are required for applications in memory storage, adaptive systems, and so on.

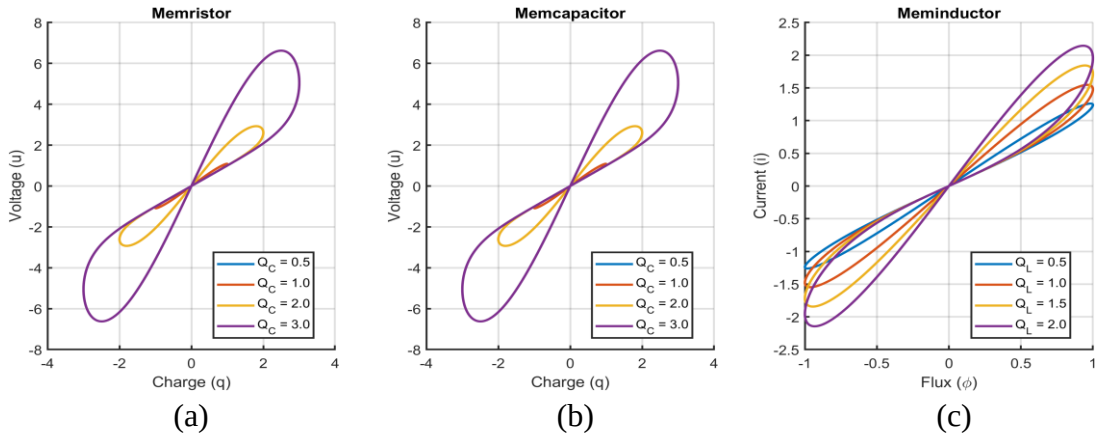


Figure 3.2: Hysteresis Loops for Memelements: (a) Memristor $q - u$ Hysteresis, (b) Memcapacitor $q - u$ Hysteresis, and (c) Meminductor $\phi - i$ Hysteresis.

3.3. Integer-Order Hyperchaotic Systems with Memelements

Hyperchaotic systems, distinguished by multiple positive Lyapunov exponents and intricate nonlinear dynamics, hold immense potential for secure communication, image encryption, cryptography, and neuromorphic computing. Integrating Memelements, such as memristors, memcapacitors, and meminductors, into these systems introduces unprecedented nonlinearities and state-dependent memory effects, thereby amplifying their complexity and adaptability.

This section proposed six novel integer-order hyperchaotic systems that leverage memelements to achieve enhanced multistability, broad parameter sensitivity, and robust chaotic signatures:

- **Memristor-Based Hyperchaotic System (Proposed System-1):** A foundational model incorporating a single memristor to generate flux-controlled nonlinearity, validated through phase portraits and Lyapunov exponents (LEs: $L_1=0.35$, $L_2=0.03$).
- **Dual Memristor-Based System (Proposed System-2):** Dual memristive interactions induce competing dynamics, exhibiting hyperchaos with three positive Lyapunov exponents ($L_1 = 25.62$, $L_2 = 7.26$, $L_3 = 7.61$) and wide bifurcation ranges.
- **Dual Memristor-Band Pass Filter (BPF) System (Proposed System-3):** Enriched with a BPF, this technique enables tunable spectral properties and multi-scroll attractors, making it ideal for secure signal modulation.
- **Memcapacitor-Based System (Proposed System-4):** Charge-history-dependent memcapacitance drives nonvolatile state transitions, yielding hyperchaotic behaviour ($L_1=0.30$, $L_2=0.08$) and extreme multistability.
- **Hybrid Memcapacitor-Meminductor System (Proposed System-5):** This system combines memcapacitive and meminductive dynamics for bidirectional energy coupling, validated by bifurcation hysteresis and five-dimensional chaos.
- **Memcapacitor-Meminductor Enhanced System (Proposed System-6):** This system integrates adaptive feedback to optimize hyperchaotic bandwidth, achieving stabilized LEs ($L_1 = 0.16$, $L_2 = 0.002$) over extended durations.

Each system is rigorously analyzed via numerical simulations (Runge-Kutta methods), bifurcation diagrams, and Lyapunov spectra to confirm hyperchaotic behaviour. Key innovations include tunable parameters (e.g., a , d , m_2) for chaos control, multi-scroll attractors for high entropy, and **hardware-inspired circuit designs** for real-world applications.

These techniques demonstrated excellent performance in image encryption, with computational efficiency (less than 18 seconds for sequence generation) and sensitivity to initial conditions, making them strong candidates for next-generation secure communication frameworks.

3.3.1. Memristor-Based Hyperchaotic System

The Memristive hyperchaotic modelling is derived from the Memristive-based hyperchaotic circuit, as illustrated in Figure 3.3. This circuit comprises a memristor, two capacitors, a resistor, and an inductor. The improved mathematical model of the entire circuit is produced by using cubic memristor modelling (Belete et al., 2025a; Sahin et al., 2020).

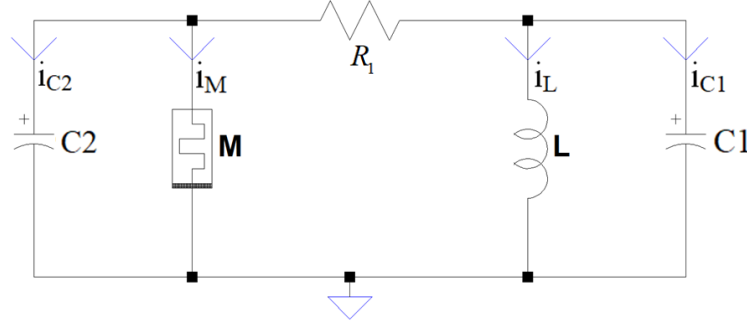


Figure 3.3: The Memristor-Based Hyperchaotic System Circuit

Kirchhoff's rules can be used to get the model equation set, which is provided by:

$$\begin{aligned}
 \dot{v}_{C1} &= \frac{Kv_{C1}(t) - v_{C3}(t) - v_{C1}(t)}{R_3C_1} + \frac{v_{C2}(t) - v_{C1}(t)}{R_1C_1} - \frac{v_{C1}(t)}{R_2C_1} \\
 \dot{v}_{C2} &= \frac{v_{C1}(t)}{R_1C_2} - \frac{v_{C2}(t)}{R_1C_2} - i_M(t) \\
 i_L &= \frac{Kv_{C1}(t) - v_{C3}(t) - v_{C1}(t)}{R_3C_1} \\
 \dot{v}_{CM} &= \frac{v_{C2}(t)}{R_MC}
 \end{aligned} \tag{3.4}$$

where C , R_2 , R_M , and R_3 are capacitance and resistance values of the memristor along with memory characteristics. The new modified memristive hyperchaotic system mathematical model was obtained by transforming the parameters such that $v_{C1} = x$, $v_{C2} = y$, $i_L = z$, and $v_{CM} = w$, $\gamma=1/C_1$, $\alpha=1/C_2$, $\beta=1/R_1$, $\delta=1/L$, $\epsilon=1/R_2$, $f=1/R_MC$, and $\epsilon=1/R_3$. The normalized equation set for the new scenario is obtained in equation (3.5) as follows.

$$\begin{aligned}
 \dot{x} &= \alpha\beta y - \alpha\beta x - \alpha f z + \alpha x((K-1)f - \epsilon) \\
 \dot{y} &= \gamma\beta x - \gamma\beta y - \gamma y(n_0 + n_1 w^2) \\
 \dot{z} &= \delta f x(K-1) - \delta f z \\
 \dot{w} &= \epsilon x
 \end{aligned} \tag{3.5}$$

Equation (3.5) is subjected to the numerical analysis using the Runge-Kutta ode23s method in MATLAB. The initial states are given as $[x, y, z, w] = [0.10, 0.0, 0.0, 0.0]$ and each of the parameter values which are capable of generating hyperchaotic dynamics is analyzed using bifurcation diagrams and the best parameters are selected as ($\alpha=1$, $\beta=1$, $\gamma=9.98$, $\delta=1$, $\epsilon=3$, $f=3.75$, $\epsilon=-3$, $n_0=-1.2$, $n_1=0.1$, and $K=3.05$). Phase portraits (x_z , x_w , y_z , and y_w plane) of the hyperchaotic attractors for the improved hyperchaotic system based on memristor are displayed in Figure 3.4.

We look into a few system parameters that serve as bifurcation parameters. To investigate the system's sensitivity to specific parameters, We assessed the diagrams of bifurcations by raising

the parameter to be used within a designated range while maintaining fixed values for all other values and the initial states.

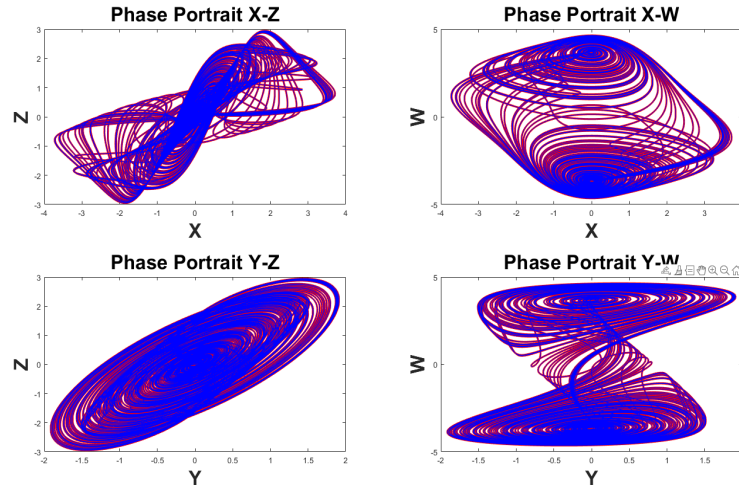


Figure 3.4: Phase Portraits of the Hyperchaotic Attractors.

The bifurcation diagrams are displayed in Figure 3.5. The diagrams show that the modified memristor-based hyperchaotic system manifests considerable hyperchaos throughout each bifurcation interval.

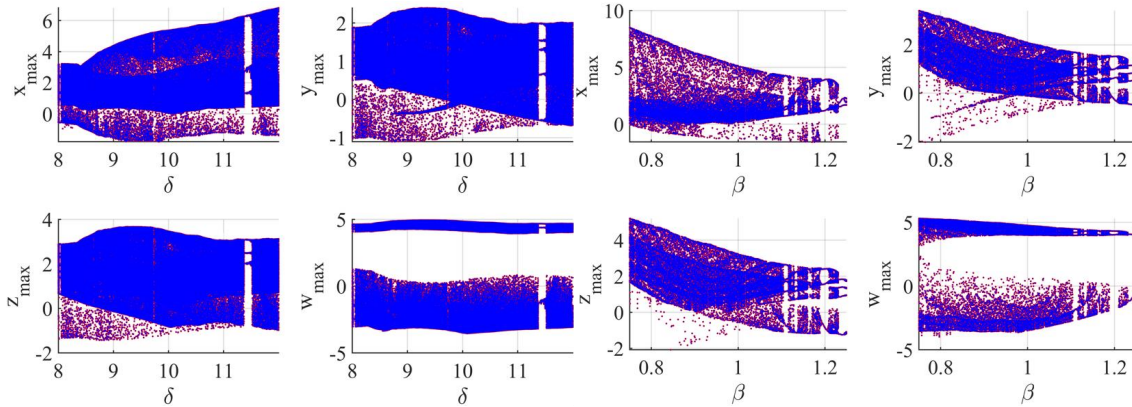


Figure 3.5: Bifurcation Diagrams of Hyperchaotic System for γ (8.0 to 12.0), β (0.75 to 1.25).

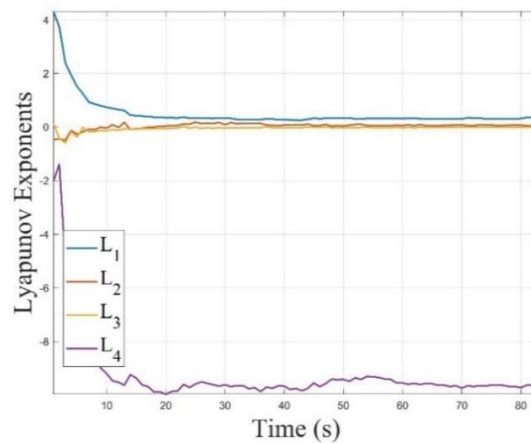


Figure 3.6: Phase Portraits of the Hyperchaotic Attractors.

The phase portraits and the Lyapunov exponent diagrams show that the chaotic system has a hyperchaotic state with two positive Lyapunov exponents ($L_1=0.349936$, $L_2=0.031855$, $L_3=-0.010692$, and $L_4=-9.615182$), large key, and good randomness. Figure 3.6 shows Lyapunov exponents over time, indicating a hyperchaotic system.

3.3.2. Dual Memristor-Based System

The circuit shown in Figure 3.7 is a novel hyperchaotic system model with four independent, autonomous dual memristors: C stands for capacitor, L for inductor, W for voltage-controlled memristor, and M for charge-controlled memristor (Belete, Gelmecha, Singh, et al., 2024; Oresanya et al., 2021).

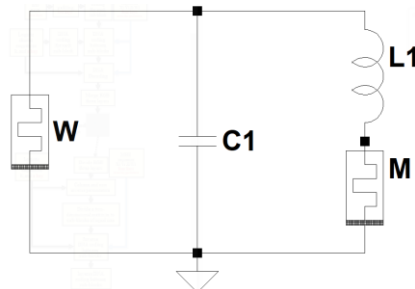


Figure 3.7: Dual Memristor-Based Hyperchaotic Circuit

Equation (3.6) gives the model equation set, which may be obtained using Kirchhoff's methods.

$$\begin{aligned}
 \dot{v}_C &= -\frac{1}{C} [i_L(t) + v_C(t)W(\phi, t)] \\
 \dot{i}_L &= \frac{1}{L} [v_C(t) - i_L(t)M(q, t)] \\
 \dot{q} &= i_L(t) \\
 \dot{v}_{CM} &= v_C(t) - 40\phi + 2v_C(t)^2\phi
 \end{aligned} \tag{3.6}$$

where, $W(\phi, t) = -0.33 + \phi + 0.5\phi^2$, and $M(q, t) = 0.01 - 3q$

Equation (3.7) provides the normalized equation set for the new case. It is obtained by substituting $x=v_C$, $y=i_L$, $z=q$, $w=\phi$, $\alpha=1/C$, and $\beta=1/L$ into Equation (3.6):

$$\begin{cases} \dot{x} = -\alpha[y + xW(w, t)] \\ \dot{y} = \beta[x - yM(z, t)] \\ \dot{z} = y \\ \dot{w} = x - 40w + 2x^2w \end{cases} \tag{3.7}$$

where, $W(w, t) = -0.33 + w + 0.5w^2$, and $M(z, t) = 0.01 - 3z$

A system's potential stable states in phase space are represented by a group of points or dots called attractor sets. Phase portraits are created using the MATLAB Runge Kutta "ode.m"

technique. The outcomes of this system are derived by using Equation (3.7). Figure 3.8 displays the chaotic attractor parameters when starting conditions $x(0)=0.001$, $y(0)=0.00$, $z(0)=2.00$, and $w(0)=0.00$ are fixed. The parameters are $\alpha=201$, and $\beta=0.79$ Figure 3.8 displays the two-dimensional graphs of the attractors of the DMHS.

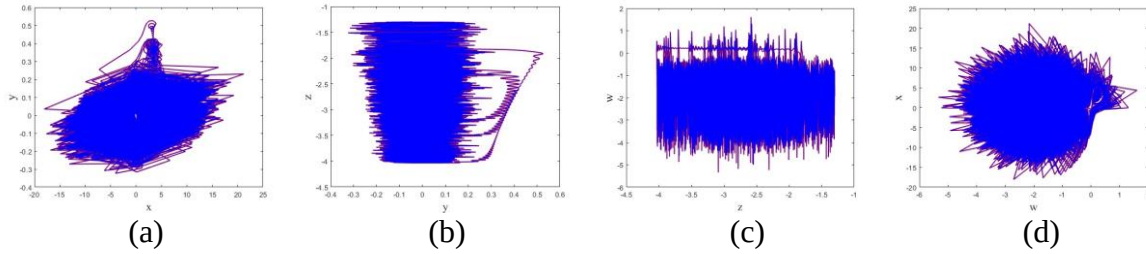


Figure 3.8: Phase Portraits of the Hyperchaotic Attractors: (a) x-y plane, (b) y-z plane, (c) z-w plane, and (d) w-x plane

By varying the chosen parameter within a predetermined range while keeping the other values and initial states constant, we investigate important system parameters as bifurcation parameters and analyze the resulting bifurcation diagrams. The notable hyperchaotic behaviour of the improved memristor-based hyperchaotic system is depicted in Figure 3.9 for the bifurcation interval. The image illustrates that the system will exhibit unpredictable behaviour for α (120-250). The finding is quite intriguing as it remains in a chaotic condition throughout a wide range of α values.

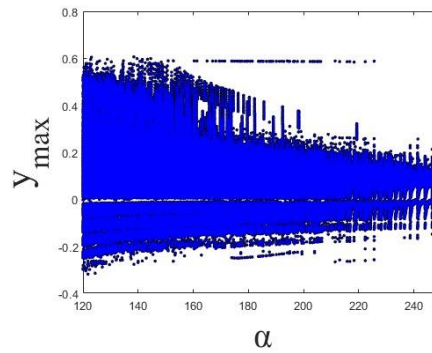


Figure 3.9: Diagram Showing the Bifurcation of a Hyperchaotic System (for parameter " α ")

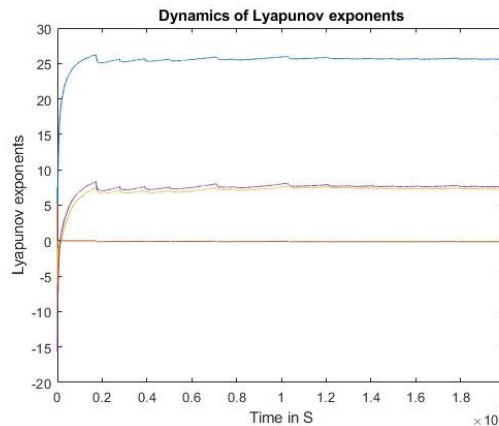


Figure 3.10: The Hyperchaotic Attractors' Phase Portraits

The chaotic features are determined by the number of positive Lyapunov exponents: one positive Lyapunov exponent indicates chaos, while two indicate hyperchaos. Lyapunov exponent diagrams in Figure 3.10 indicate that the system is in an exceptional hyperchaotic state, characterized by very high unpredictability, with three large positive Lyapunov exponents ($L1 = 25.6192$, $L2 = -0.1299$, $L3 = 7.2645$, and $L4 = 7.6140$).

3.3.3. Dual Memristor-Band Pass Filter System

We proposed a memristor-based hyperchaotic system with two memristors and a band-pass filter (BPF) for our image encryption and denoising algorithm (Belete et al., 2025b), as shown in Figure 3.11.

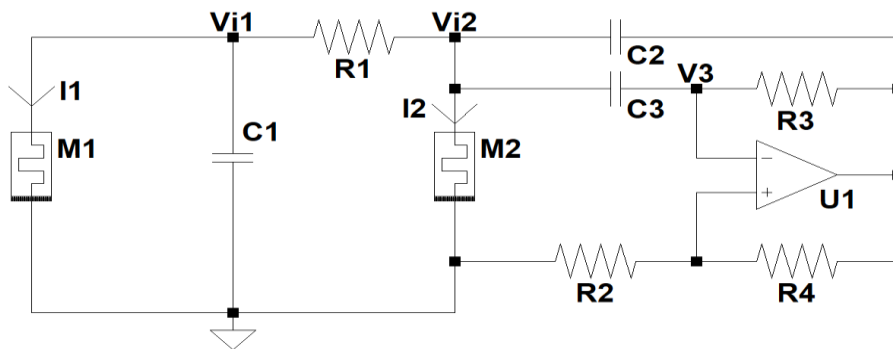


Figure 3.11: Memristive Chua's Hyperchaotic Circuit

The mathematical models for the two memristors, M_1 and M_2 , are outlined in (X. Gao et al., 2021) and can be expressed as follows:

$$\begin{cases} I_1 = M_1(V_{n1})V_{i1} = -\frac{1}{R_a}(1 - m_1V_{n1}^2)V_{i1} \\ \frac{dV_{n1}}{dt} = f(V_{i1}) = -\frac{1}{R_bC_{M1}} \end{cases} \quad (3.8)$$

where, I_1 represents the input current of M_1 , V_{i1} represents the input voltage of M_1 , V_{n1} is the node voltage within the memristor M_1 and m_1 denotes the total circuit gain of the memristor M_1 .

$$\begin{cases} I_2 = M_2(V_{n2})V_{i2} = -\frac{1}{R_c}(1 - m_2V_{n2}^2)V_{i2} \\ \frac{dV_{n2}}{dt} = f(V_{i2}) = -\frac{1}{R_dC_{M2}} \end{cases} \quad (3.9)$$

where, I_2 represents the input current of M_2 , V_{i2} represents the input voltage of M_2 , V_{n2} is the node voltage within the memristor M_2 and m_2 denotes the total circuit gain of the memristor M_2 .

Using Kirchhoff's laws and the current-voltage relationships for capacitors and memristors, the mathematical model of the hyperchaotic system is derived as follows:

$$\begin{cases} \frac{dV_{i1}}{dt} = -\frac{1}{R_1 C_1}(V_{i1} - V_{i2}) + \frac{1}{R_a C_1}(1 - m_1 V_{n1}^2)V_{i1} \\ \frac{dV_{i2}}{dt} = -\frac{1}{R_1 C_3}(V_{i1} - V_{i2}) + \frac{1}{R_c C_1}(1 - m_2 V_{n2}^2)V_{i2} - \frac{2R_2 + R_4}{(R_2 + R_4)R_3 C_3}V_3 \\ \frac{dV_3}{dt} = -\frac{R_2 + R_4}{R_1 R_4 C_2}(V_{i1} - V_{i2}) + \frac{\frac{R_2}{R_4} + 1}{R_c C_2}(1 - m_2 V_{n2}^2)V_{i2} - \frac{2}{R_3 C_2}V_3 \\ \frac{dV_{n1}}{dt} = -\frac{1}{R_b C_{M1}}V_{i1} \\ \frac{dV_{n2}}{dt} = -\frac{1}{R_d C_{M2}}V_{i2} \end{cases} \quad (3.10)$$

For Equation (3.10), we introduce new variables and scale the circuit parameters as follows:

$$\begin{cases} x = V_{i1}, y = V_{i2}, z = V_3, w = V_{n1}, u = V_{n2} \\ C = C_2 = C_3, R_b C_{M1} = R_d C_{M2} \\ a = \frac{C}{C_1}, b = \frac{R_1 C}{R_a C_1}, c = \frac{R_1}{R_c}, d = \frac{R_1}{R_3}, e = \frac{R_1 C}{R_b C_{M1}}, g = \frac{R_2}{R_4} \end{cases} \quad (3.11)$$

According to Equation (3.10), Equation (3.11) transforms into:

$$\begin{cases} \dot{x} = -a(x - y) + b(1 - m_1 w^2)x \\ \dot{y} = -g(x - y) + gc(1 - m_2 v^2)y - (2g + 1)/(g + 1)dz \\ \dot{z} = -(g + 1)(x - y) + (g + 1)c(1 - m_2 v^2)y - 2dz \\ \dot{w} = -ex \\ \dot{u} = -ey \end{cases} \quad (3.12)$$

By initialising the system described by Equation (3.12) with the initial conditions $(x_0, y_0, z_0, w_0, u_0) = (0.1, 0, 0, 0, 0)$, with parameters $a = 20, b = 21.43, c = 15, d = 0.15, e = 3, g = 0.05, m_1 = 0.1$ and $m_2 = 0.1$. The 2D and 3D state trajectory diagrams in Figure 3.12 show the multi-scroll chaotic attractor for various variables. The system's chaotic properties, as determined by Equation (3.12), were confirmed using the Jacobian method to calculate the Lyapunov exponents (Benkouider et al., 2024).

The computational time required to generate the hyperchaotic sequences is less than 18 seconds, which is directly related to the computational efficiency of the hyperchaotic system in practical applications. The computed Lyapunov exponents for a running time of 18 seconds are: $L_1 = 0.407598, L_2 = 0.211550, L_3 = -0.079103$, and $L_4 = -0.253775, L_5 = -2.940882$. With L_1 and L_2 being positive, L_3 equal to zero, and L_4 and L_5 being negative, the Lyapunov exponents exhibit the pattern $(+, +, -, -, -)$. For a running time of 60 seconds,

the Lyapunov exponents are: $L_1 = 0.263927$, $L_2 = 0.050132$, $L_3 = -0.056580$, and $L_4 = -0.052391$, $L_5 = -3.656629$.

For an extended running time of 10000 seconds, the Lyapunov exponents converge to: $L_1 = 0.161308$, $L_2 = 0.002066$, $L_3 = -0.000217$, and $L_4 = -0.002414$, $L_5 = -4.105235$. Thus, the system in Equation (3.12) is confirmed to be hyperchaotic for all time intervals. Figure 3.13 illustrates the Lyapunov exponents of the hyperchaotic system for running times of 18 seconds, 60 seconds, and 10,000 seconds, demonstrating that the Lyapunov exponent values meet the criteria for a five-dimensional hyperchaotic system.

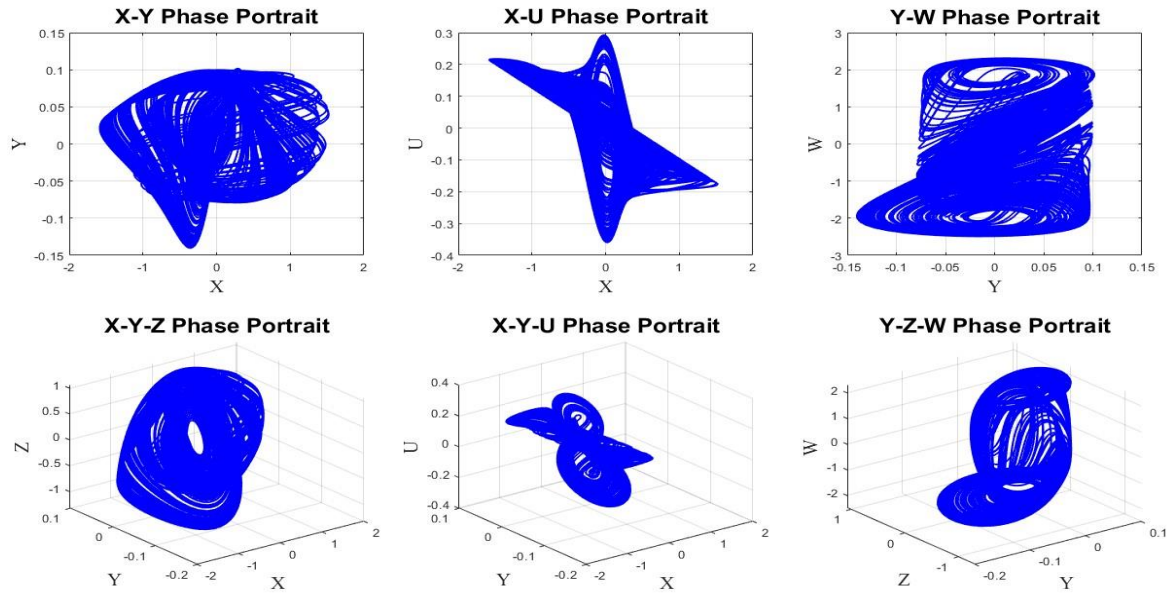


Figure 3.12: 2D and 3D Phase Portrait of the Hyperchaotic System (5)

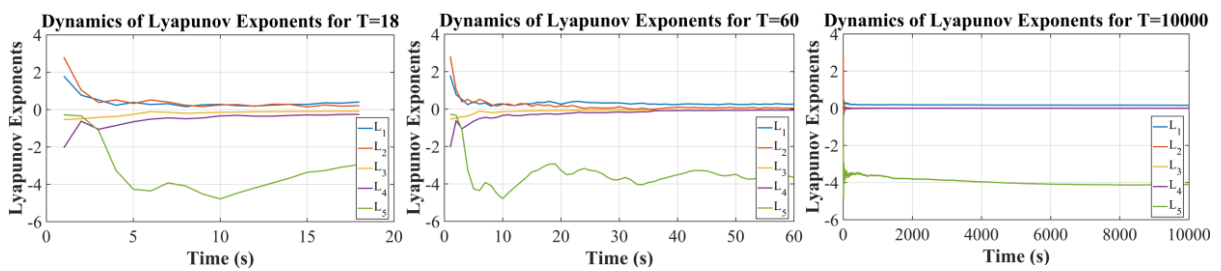


Figure 3.13: Lyapunov Exponent Diagrams of the Hyperchaotic System (3.12) for Running times $T=18$, 60, and 10000 seconds.

In the hyperchaotic system defined by Equation (3.12), the parameters a , b and c play a crucial role in influencing the system's dynamics. With these parameters set and initial conditions specified, the bifurcation diagrams for a ranging from 19.5 to 21, b ranging from 0 to 10, and c ranging from 5 to 7 are illustrated in Figure 3.14.

As parameter a increases from 19.5 to 21, d increases from 0.13 to 0.2, and m_2 increases from 0 to 0.05, the system predominantly exhibits chaotic behavior with multistability. Hence, it is evident that tuning the parameters a, d , and m_2 can induce both chaotic and hyperchaotic behaviors within the system. Specifically, when $a = 20, b = 21.43, c = 15, d = 0.15, e = 3, g = 0.05, m_1 = 0.1$ and $m_2 = 0.1$ the system demonstrates hyperchaotic behavior.

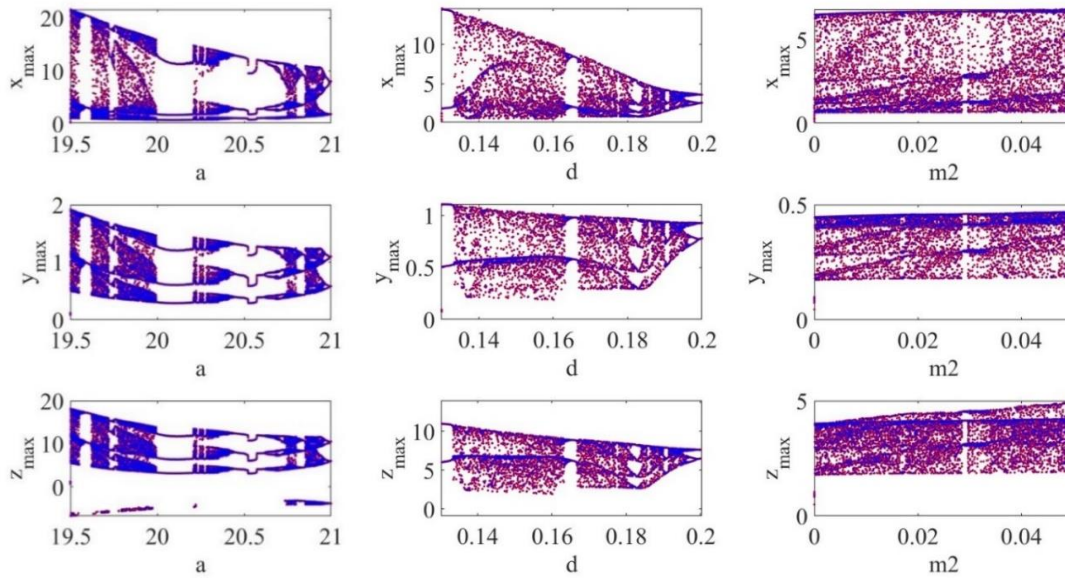


Figure 3.14: Bifurcation Diagrams, (a) $a \in (19.5, 21)$, (b) $d \in (0.13, 0.20)$, (c) $m_2 \in (0, 0.05)$

3.3.4. Memcapacitor-Based System

Firstly, this research article proposed a newly introduced memcapacitive hyperchaotic system that is controlled by charge as shown in Figure 3.15. A dynamic examination of the hyperchaotic system reveals the multistability of the system and it has an infinite equilibrium point [14, 15]. Instead of starting with the initial conditions as was the case, the extreme chaotic oscillator multistability is described with the system's parameters (Belete, Gelmecha, & Singh, 2024; Rajagopal et al., 2018).

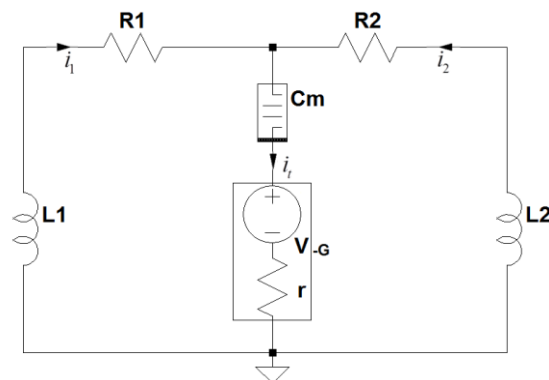


Figure 3.15: A Hyperchaotic Oscillator Based on Memcapacitors.

The memcapacitance value is given by; the relationship between the voltage $v_{CM}(t)$ and the charge $q_{CM}(t)$ of the memcapacitor is defined as the charge through the memcapacitor. A memcapacitive system with four differential equations derived as equation (3.13),

$$\begin{aligned}
\frac{dq_{CM}}{dt} &= i_1 + i_2 \\
\frac{di_1}{dt} &= \frac{1}{L_1} [-(\alpha + \beta\sigma^2)q_{CM}(t) - i_1R_1 + (i_1 + i_2)R_0] \\
\frac{di_2}{dt} &= \frac{1}{L_2} [-(\alpha + \beta\sigma^2)q_{CM}(t) - i_2R_2 + (i_1 + i_2)R_0] \\
\frac{d\sigma}{dt} &= q_{CM}(t)
\end{aligned} \tag{3.13}$$

where, R_0 , R_1 , and R_2 are the resistances, L_1 and L_2 are the inductors, G is the inverse of the resistance r , C_m is the memcapacitance, q_{CM} is the memcapacitor charge, $v_{CM}(t)$ is the memcapacitor voltage. The equation $\alpha + \beta\sigma^2$ represents the inverse memcapacitance, σ is the time integral of the charge $q_{CM}(t)$ given by $\sigma = \int_{t_0}^t q_{CM}(t) dt$.

To derive a generalised four-dimensional model of the memcapacitor system, let us define $x = q_{CM}(t)$, $y = i_1$, $z = i_2$, and $w = \sigma$. Additionally, the study expresses the constant parameters as; $a = \frac{\alpha}{L_1}$, $b = \frac{\beta}{L_1}$, $c = \frac{(R_0 - R_1)}{L_1}$, $d = \frac{R_0}{L_1}$, $e = \frac{(R_0 - R_2)}{L_2}$, $f = \frac{R_0}{L_2}$, $m = \frac{\alpha}{L_2}$, and $\sigma = \frac{\beta}{L_2}$. By using the derived equation's (3.13) proper assumptions, the four-dimensional (4D) memcapacitor-based hyperchaotic system mathematical model is expressed as,

$$\begin{aligned}
\frac{dx}{dt} &= y + z \\
\frac{dy}{dt} &= cy + dz - ax - bxw^2 \\
\frac{dz}{dt} &= fy + ez - mx - nxw^2 \\
\frac{dw}{dt} &= x
\end{aligned} \tag{3.14}$$

In equation (3.14), the selected variables exhibit hyperchaotic oscillations in the chaotic system. Dynamical properties are examined to demonstrate how applicable such a chaotic system is in the cryptography area after the parameters of initial values (X_0, Y_0, Z_0 , and H_0) are given as [0.001 0.001 0.001 0.001]. Figure 3.16 displays the system's 2D phase portraits. The chaotic behavior can be confirmed using the Lyapunov exponential spectrum and bifurcation diagram. This research employed the Jacobian approach to determine the LEs of the hyperchaotic

memcapacitor system. The memcapacitor hyperchaotic system's Lyapunov exponents are calculated numerically to be $L_1 = 0.29911$, $L_2 = 0.076340$, $L_3 = 0$, and $L_4 = -1.0740$.

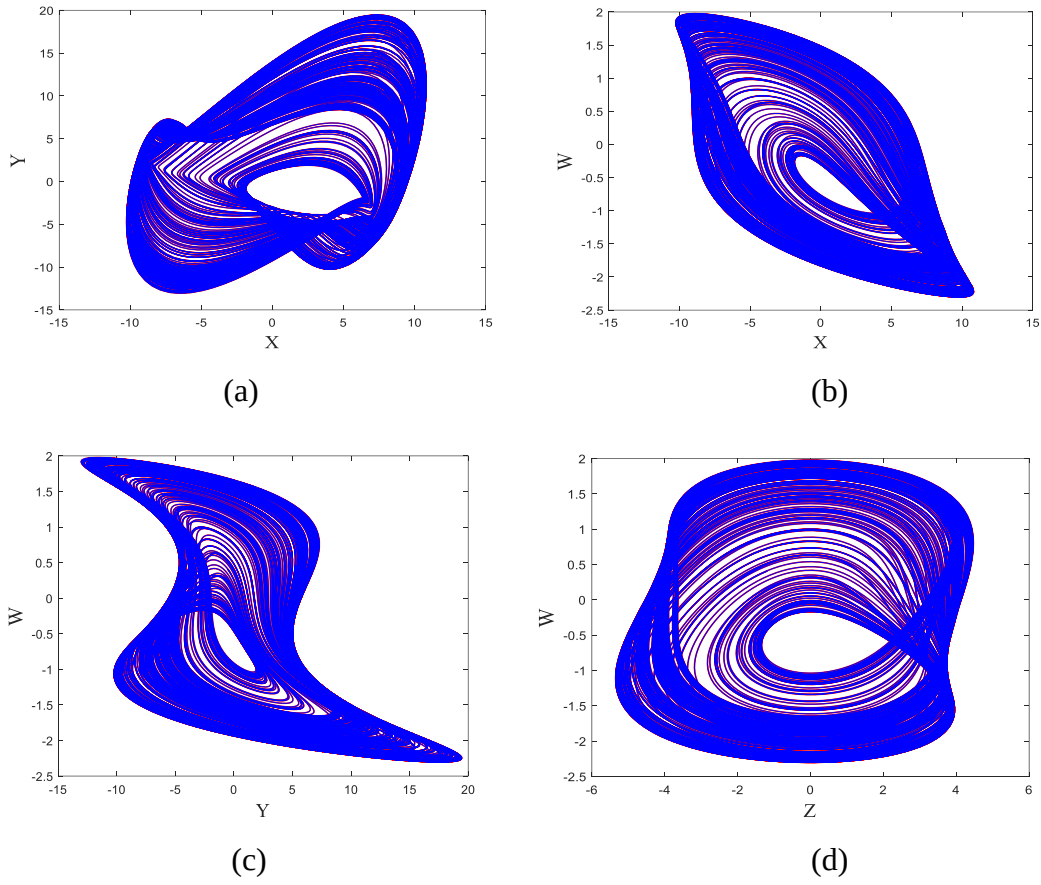


Figure 3.16: The Phase Portraits of the Proposed Memcapacitor-Based Hyperchaotic System onto Different Spaces (a) x-y, (b) x-w, (c) y-w, and (d) z-w.

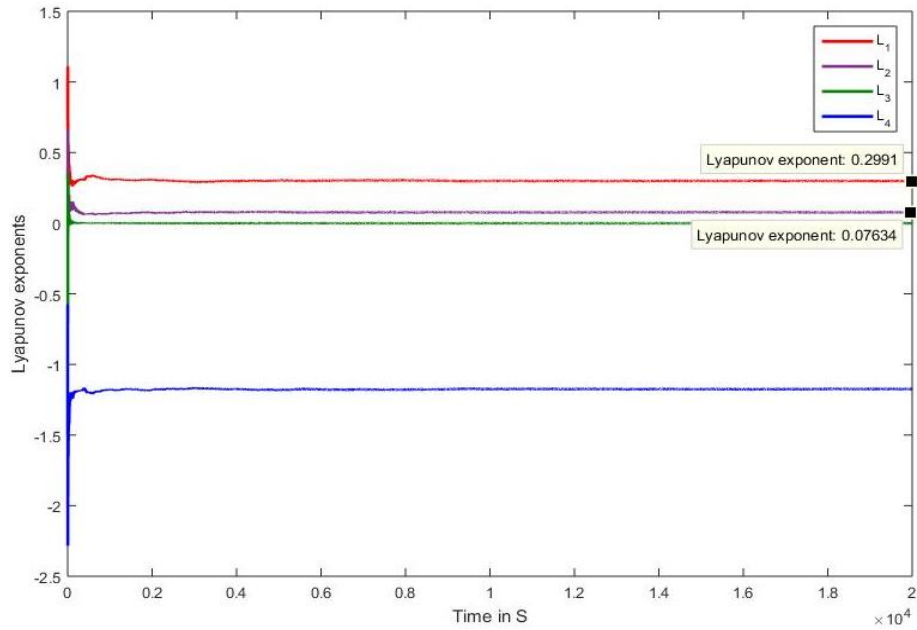


Figure 3.17: Lyapunov Exponents of the Memcapacitive Hyperchaotic System

Since the system with equation (3.14) exhibits two positive Lyapunov exponents, therefore, hyperchaos is present. The Lyapunov exponents of the memcapacitive hyperchaotic system are illustrated in Figure 3.17. The findings demonstrate the remarkably high unpredictability of the four-dimensional hyperchaotic system (Alhudhaif et al., 2021; M. Guo et al., 2021).

The bifurcation plots are generated to understand the parameter dependency of the hyperchaotic memcapacitive oscillator (HMCO) system. It also takes into account the two parameters, b and d , for bifurcation analysis. The hyperchaotic memcapacitor system bifurcation for parameter b is shown in Figure 3.18. The bifurcation of the hyperchaotic memcapacitor system for parameter d is depicted in Figure 3.19. The HMCO system exhibits severe multistability and coexisting attractors, as shown in Figures 3.18 and 3.19. We, therefore, look at the system's multistability in the parameter space. The behaviour of multistability is unpredictable and erratic (Rajagopal et al., 2018).

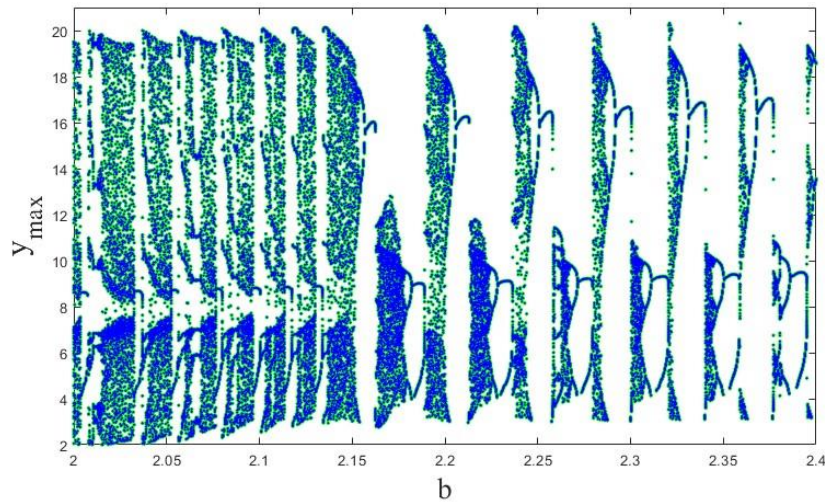


Figure 3.18: Bifurcation Plot of the Memcapacitive Hyperchaotic System for Parameter b

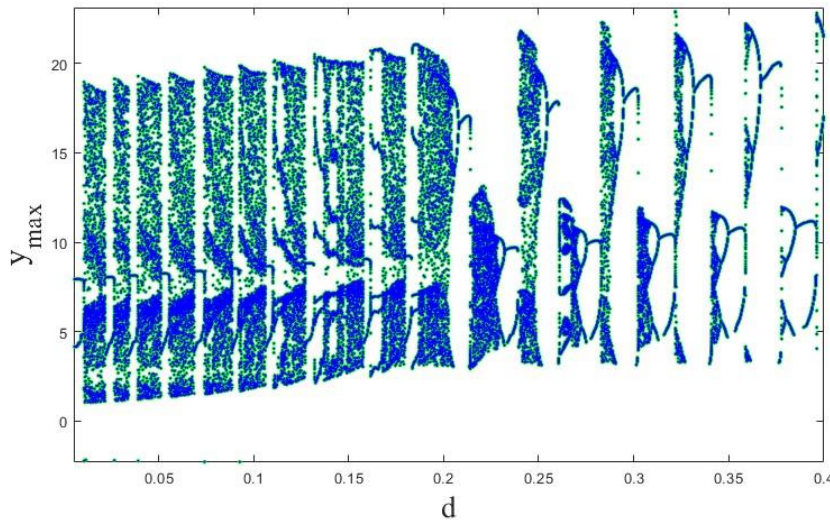


Figure 3.19: Bifurcation Plot of the Memcapacitive Hyperchaotic System for Parameter d

3.3.5. Hybrid Memcapacitor-Meminductor System

Based on the non-volatile charge-controlled memcapacitor and flux-controlled meminductor, a novel chaotic oscillating circuit is designed, as shown in Figure 3.20. The circuit consists of a meminductor (L_M), a memcapacitor (C_M), a capacitor (C_1), a linear resistor (R), and a negative resistor ($-G$). The negative resistor supplies energy to maintain the oscillatory state of the circuit, ensuring sustained chaotic behaviour.

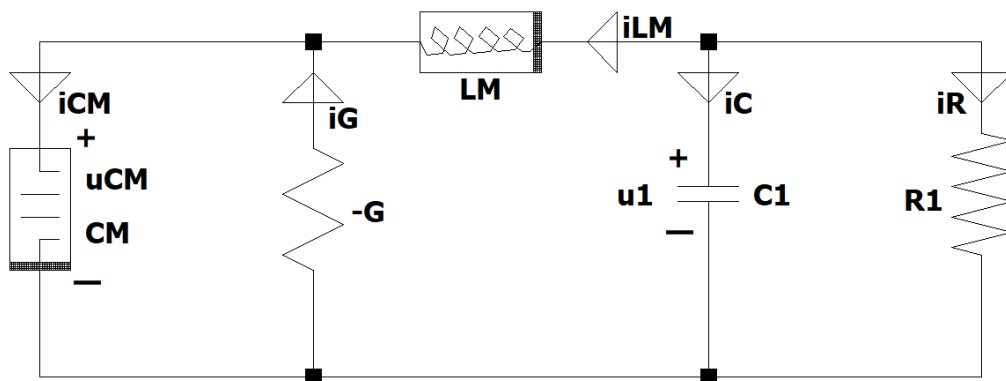


Figure 3.20: Chaotic Oscillator Circuit Using Memcapacitor and Meminductor

Using Kirchhoff's voltage and current laws, the governing equations are:

$$\begin{cases} C_1 \frac{du_1}{dt} = i_{LM} - \frac{u_1}{R} \\ \frac{dq_{CM}}{dt} = Gu_{CM} - i_{LM} \\ \frac{d\phi_{LM}}{dt} = u_{CM} - u_1 \end{cases} \quad (3.15)$$

where $u_1, u_{CM}, \phi_{LM}, q_{CM}, i_{LM}$ represent voltages, flux, charge, and currents in the circuit.

Substituting the device-specific equations, the state-space representation becomes:

$$\begin{cases} \frac{du_1}{dt} = \frac{1}{C_1} \left(\alpha' \phi_{LM} + \beta' \rho_{LM} \phi_{LM} - \frac{u_1}{R} \right) \\ \frac{dq_{CM}}{dt} = G(\alpha q_{CM} + \beta q_{CM} \sigma_{CM}^2) - (\alpha' \phi_{LM} + \beta' \rho_{LM} \phi_{LM}) \\ \frac{d\phi_{LM}}{dt} = \alpha q_{CM} + \beta q_{CM} \sigma_{CM}^2 - u_1 \\ \frac{d\rho_{LM}}{dt} = \phi_{LM} \\ \frac{d\sigma_{CM}}{dt} = q_{CM} \end{cases} \quad (3.16)$$

To simplify the analysis, the system is normalized using dimensionless variables and parameters:

$$x = \frac{u_1}{E}, y = \frac{q_{CM}}{C_0 E}, z = \frac{\phi_{LM}}{L_0 I_0}, w = \frac{\rho_{LM}}{\phi_0 t_0}, u = \frac{\sigma_{CM}}{q_0 t_0}, \quad (3.17)$$

The system equations are reduced to:

$$\begin{cases} \frac{dx}{d\tau} = az + bzu - x \\ \frac{dy}{d\tau} = dy + eyu^2 - fz - gzu \\ \frac{dz}{d\tau} = hy + jyu^2 - kx \\ \frac{dw}{d\tau} = n_1z \\ \frac{du}{d\tau} = n_2y \end{cases} \quad (3.18)$$

where, $\tau = t/(C_1R)$ and the parameters $a, b, d, e, f, g, h, j, k, n_1, n_2$ are dimensionless parameters determined by the circuit elements.

This comprehensive formulation encapsulates the hyperchaotic dynamics driven by the interplay of memory-based components and nonlinearities, enabling a more profound exploration of the system's behaviour. The system under study exhibits chaotic and hyperchaotic behaviour driven by parameters and initial conditions. The parameters are defined as: $a = 1.48$, $b = -2.04$, $d = 0.46$, $e = 0.04$, $f = 0.67$, $g = 0.19$, $h = 0.48$, $j = 0.52$, $k = 0.21$, $n_1 = 0.4$, and $n_2 = 0.4$, with initial conditions of $[0.2, 0.5, 0.45, 0.1, \text{ and } 0.5]$.

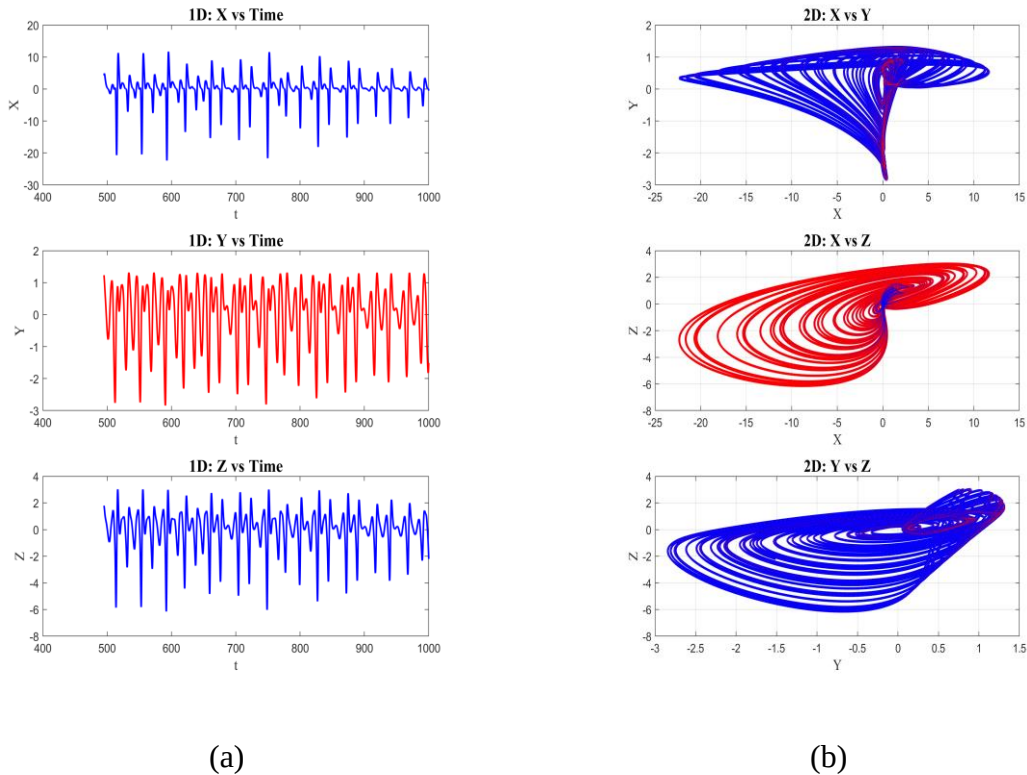


Figure 3.21: System Behavior: (a) Waveforms of State Variables (x, y, z) and (b) Chaotic Attractors of the System

Figure 3.21 illustrates the following: (a) the waveforms showing how the system evolves dynamically over time, and (b) the chaotic attractors of the system, which highlight the complex and unpredictable trajectories of the state variables in phase space.

The chaotic behaviour of the proposed system is confirmed by its positive Lyapunov exponents (Belete, Gelmecha, Singh, et al., 2024). The Lyapunov exponent results computed as, $LE1 = 0.0377$, $LE2 = 0.0348$, $LE3 = -0.0111$, $LE4 = -0.0236$, and $LE5 = -0.1766$. The first and second Lyapunov exponents show hyperchaotic behaviour. Figure 3.22 shows the Lyapunov exponents ($LE1$, $LE2$, $LE3$, $LE4$, $LE5$) of the system, which indicate the rates of divergence of nearby trajectories and reflect the system's hyperchaotic behaviour.

Figure 3.23 illustrates the forward and backward bifurcation diagrams of the hyperchaotic system, illustrating the maxima of x as the parameter ' a ' is varied. In the forward diagram, the system shifts from periodic to chaotic behaviour as ' a ' increases, while intermittent periodicity appears. The backward diagram shows different transition values, highlighting path dependence and bistability. The scattered points in chaotic regions and differences between the diagrams emphasize the circuit's nonlinear and memory-dependent characteristics.

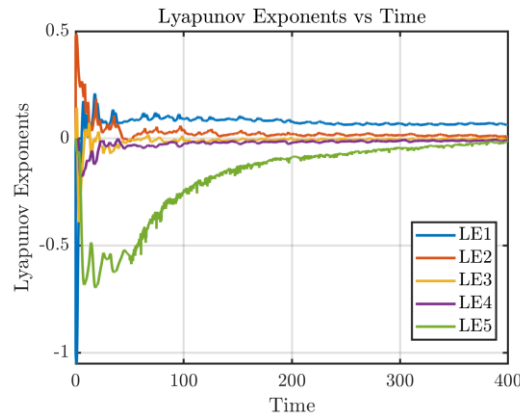


Figure 3.22: Lyapunov Exponents ($LE1$, $LE2$, $LE3$, $LE4$, $LE5$) of the System

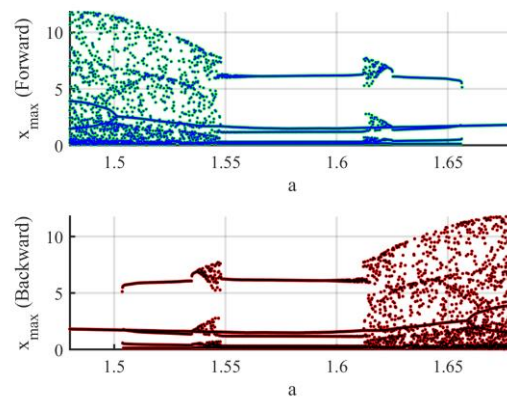


Figure 3.23: Bifurcation of the Memelements-Based Hyperchaotic system with Forward and Backward Parameter (a) Variation

3.3.6. Memcapacitor-Meminductor Enhanced System

Memelements such as Memristors, Memcapacitors, and Meminductors are nonlinear circuit elements that exhibit memory-dependent characteristics. These components have gained attention recently due to their potential applications in complex nonlinear dynamical systems, including chaos-based cryptography and neural networks. The proposed hyperchaotic system study included modelling memcapacitors and meminductors and then deriving a chaotic oscillator utilizing these elements for compressed image encryption applications. The charge-controlled Memcapacitor (X. Liu & Wang, 2022) is mathematically modelled as follows:

$$\begin{aligned} v_{MC}(t) &= C_{MC}^{-1}(\sigma)q_{MC}(t) \\ C_{MC}^{-1}(\sigma) &= \frac{d\varphi_{MC}(\sigma)}{d\sigma} \end{aligned} \quad (3.19)$$

where $v_{MC}(t)$ and $q_{MC}(t)$ are the voltage and charge across the memcapacitor, $C_{MC}^{-1}(\sigma)$ represents the inverse memcapacitance function dependent on charge history, σ denotes the integral of charge $q_{MC}(t)$, and $\varphi_{MC}(\sigma)$ is the magnetic flux across the memcapacitor. For a sinusoidal function model, the charge-controlled memcapacitor is modelled as follows:

$$v_{MC}(t) = (a + b \sin \sigma)q_{MC}(t) = \left(a + b \sin \int q_{MC}(s)ds \right) q_{MC}(t) \quad (3.20)$$

where a and b are constants.

Similarly, the flux-controlled meminductor (X. Liu & Wang, 2022) is given by:

$$\begin{cases} i_{ML}(t) = L_{ML}^{-1}(\rho)\varphi_{ML}(t) \\ L_{ML}^{-1}(\rho) = \frac{dq_{ML}(\rho)}{d\rho} \end{cases} \quad (3.21)$$

where $L_{ML}^{-1}(\rho)$ governs the inverse meminductance function based on flux history, and ρ is the integral of the magnetic flux $\varphi_{ML}(t)$, $q_{ML}(\rho)$ denotes the charge across the meminductor. The flux-controlled meminductor is defined according to the formula:

$$i_{ML}(t) = (c + d \cos \rho)\varphi_{ML}(t) = \left(c + d \cos \int \varphi_{ML}(s)ds \right) \varphi_{ML}(t) \quad (3.22)$$

where c and d are constants.

A hyperchaotic oscillator incorporating a memcapacitor and meminductor is constructed using a circuit comprising the capacitance of the memcapacitor CMC, the inductance of the meminductor LML, the resistor R, the negative conductance $-G$, and the conventional capacitor C, as shown in Figure 3.24.

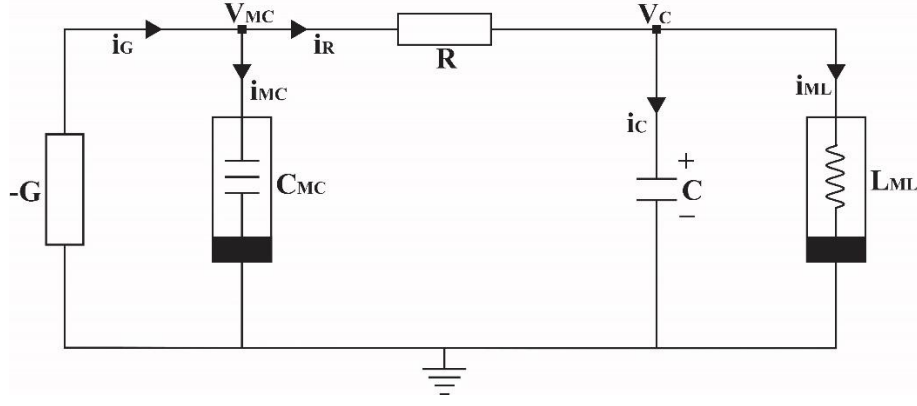


Figure 3.24: The Hyperchaotic Oscillator Circuit of the Proposed Memcapacitor and Meminductor

Applying Kirchhoff's laws, the state equations of the circuit are given by:

$$\left\{ \begin{array}{l} C \frac{dV_C(t)}{dt} = \frac{v_{MC}(t) - V_C}{R} - i_{ML}(t) \\ \frac{dq_{MC}(t)}{dt} = Gv_{MC}(t) - \left(\frac{v_{MC}(t) - V_C(t)}{R} \right) \\ \frac{d\varphi_{ML}(t)}{dt} = V_C(t) \\ \frac{d\sigma(t)}{dt} = q_{MC}(t) \\ \frac{d\rho(t)}{dt} = \varphi_{ML}(t) \end{array} \right. \quad (3.23)$$

After substituting $v_{MC}(t)$ and $i_{ML}(t)$, the equation is updated as follows:

$$\left\{ \begin{array}{l} \frac{dV_C}{dt} = \frac{(a + b \sin \sigma)q_{MC} - V_C}{RC} - \frac{(d + e \cos \rho)\varphi_{ML}}{C} \\ \frac{dq_{MC}}{dt} = \frac{V_C}{R} + \left(G - \frac{1}{R} \right) (a + b \sin \sigma)q_{MC} \\ \frac{d\varphi_{ML}}{dt} = V_C \\ \frac{d\sigma}{dt} = q_{MC} \\ \frac{d\rho}{dt} = \varphi_{ML} \end{array} \right. \quad (3.24)$$

Using state variables $x = V_C$, $y = q_{MC}$, $z = \varphi_{ML}$, $u = \sigma$, $w = \rho$, $R = e$, $C = f$, and $G = g$ the dimensionless system can be written as: eq7

$$\left\{ \begin{array}{l} \dot{x} = \frac{(a + b \sin u)y - x}{ef} - \frac{(c + d \cos w)z}{f} \\ \dot{y} = \frac{x}{e} + \left(g - \frac{1}{e} \right) (a + b \sin u)y \\ \dot{z} = x \\ \dot{u} = y \\ \dot{w} = z \end{array} \right. \quad (3.25)$$

Figure 3.25 shows the 2D and 3D phase portraits of the IOMBHCS system, highlighting its chaotic behaviour. The first two plots display 2D phase portraits in the (x, y) and (x, z) planes, illustrating the system's complex, non-repetitive trajectories. The rightmost plot reveals the 3D phase space (x, y, z) and the attractor structure, indicating chaos. These portraits reflect the system's sensitive dependence on initial conditions and the presence of a strange attractor, key features of chaotic dynamics.

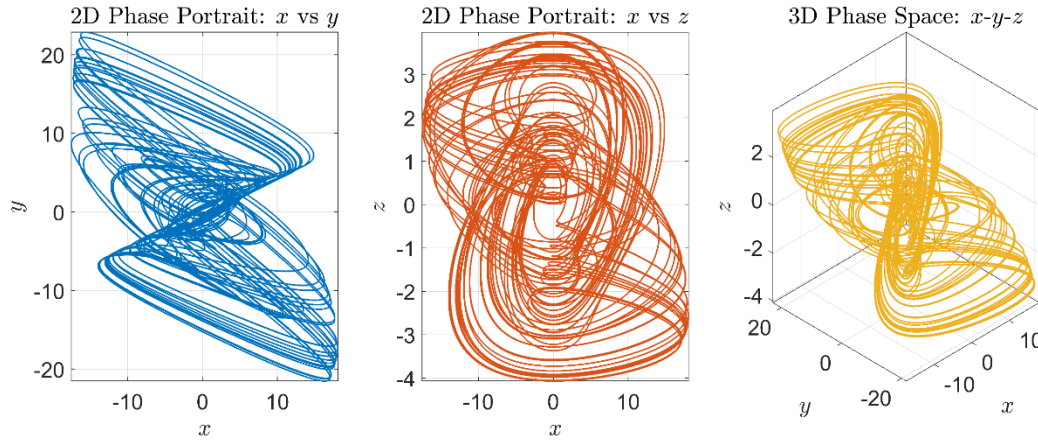


Figure 3.25: 2D and 3D Phase Portraits of IOMBHCS.

For specific parameter values (e.g., $a=0.41$, $b=2.87$, $c=10$, $d=3$, $e=0.2$, $f=1$, $g=2$), Figure 3.26 illustrates the Lyapunov Exponents of the IOMBHCS system over different operational durations. Subfigure (a) displays the Lyapunov Exponents calculated over 100 iterations, while subfigure (b) extends the computation duration to 10,000 iterations. The graphs indicate that the exponents stabilize after an initial transient phase, suggesting the system exhibits chaotic behaviour. Positive Lyapunov exponents confirm the system's hyperchaotic property, a crucial feature for its applications in image encryption and secure communications in general.

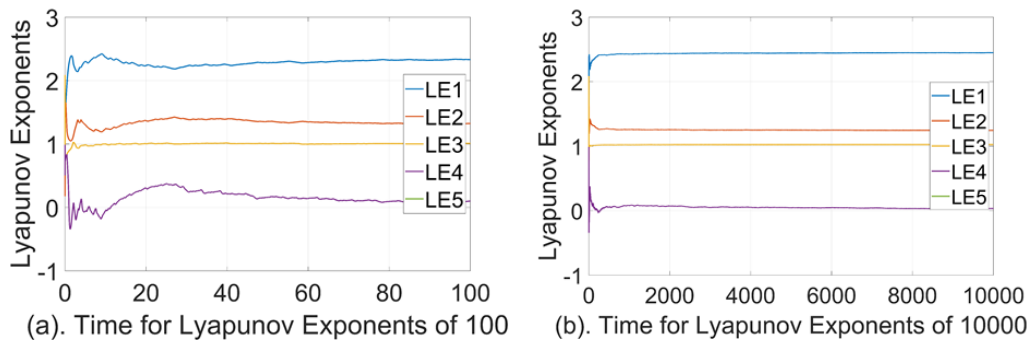


Figure 3.26: Lyapunov Exponents of IOMBHCS for Different Operation Times.

Figure 3.27 shows the forward and reverse bifurcation diagrams of the IOMBHCS system related to the parameter d . The blue diagram on the left illustrates the system's behavior as d increases, while the red diagram on the right depicts its response as d decreases. Both diagrams

demonstrate chaotic behavior at lower d values, stabilizing as d increases. The differences between the forward and reverse patterns suggest a hysteresis effect, indicating possible bistability or multistability in the system.

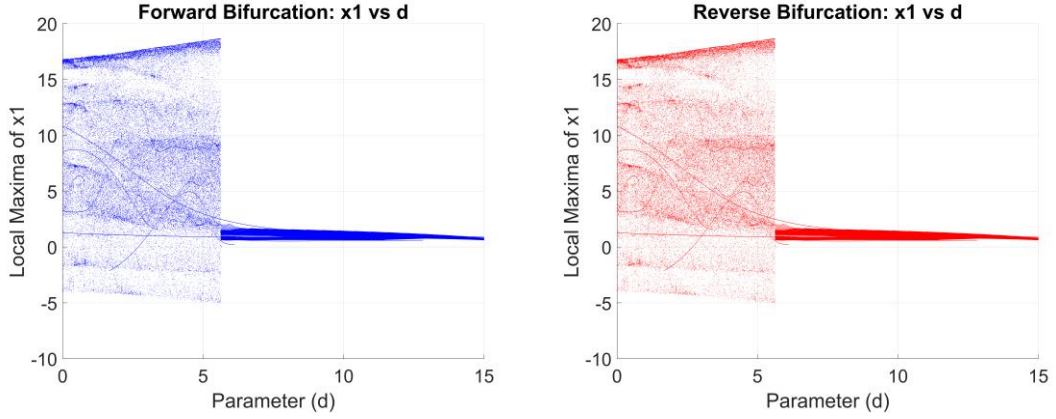


Figure 3.27: Forward and Reverse Bifurcation Diagram of the IOMBHCS Concerning Parameter d

3.4. Fractional-Order Hyperchaotic Systems with Memelements

The integer-order Memelements-based hyperchaotic system study is extended to fractional-order form using the Caputo differential definition (J. Liu et al., 2022). This extension enhances the system's chaotic behaviour, characterized by a high Lyapunov exponent, self-reproduction of attractors, and inherent instability. The system's dynamics are analyzed through system phase portraits, bifurcation diagrams, and Lyapunov exponents, providing deeper insights into its rich and complex behaviour. The Caputo fractional derivative of order q is defined as (Kilbas & Marzan, 2005):

$$*D_t^q f(t) = \begin{cases} \frac{1}{\Gamma(m-q)} \int_0^t \frac{f^{(m)}(\tau) d\tau}{(t-\tau)^{q+1-m}}, & m-1 < q < m \\ \frac{d^m}{dt^m} f(t), & q = m, q \in \mathbb{R}^+, m \in \mathbb{N} \end{cases} \quad (3.26)$$

where $*D_t^q f(t)$ represents the Caputo fractional derivative of order q of the function $f(t)$, q is the order of the fractional derivative, m is the smallest integer greater than q , $\Gamma(m-q)$ is the Gamma function, $f^{(m)}(\tau)$ is the m -th order derivative of the function f with respect to τ , $(t-\tau)^{q+1-m}$ is a weighting factor in the integral, $\frac{d^m}{dt^m} f(t)$ is the standard m -th order derivative of $f(t)$, $\mathbb{R}^+$ is the set of positive real numbers, and $\mathbb{N}$ is the set of natural numbers. The ADM can efficiently solve fractional-order nonlinear systems with high accuracy, providing approximate numerical solutions (J. Liu et al., 2022). Based on the ADM, the proposed fractional-order differential equation can be expressed as:

$$* D_{t_0}^q x(t) = Lx(t) + Nx(t) + c(t), \quad (3.27)$$

where $* D_{t_0}^q$ denotes the Caputo derivative operator of order q . L , N and c represent the linear, nonlinear and constant terms of the system, respectively. $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ is the state variable. Application of the R-L fractional integral operator $J_{t_0}^q$ representing q -order to both sides of the above equation will give:

$$x(t) = J_{t_0}^q Lx(t) + J_{t_0}^q Nx(t) + J_{t_0}^q c(t) + x, \quad (3.28)$$

where the initial value is given by $x(t_0^+)$. The integral operator $J_{t_0}^q$ has the following properties:

$$J_{t_0}^q C = \frac{C}{\Gamma(q+1)}(t-t_0)^q. \quad (3.29)$$

$$J_{t_0}^q (t-t_0)^\xi = \frac{\Gamma(\xi+1)}{\Gamma(\xi+1+q)}(t-t_0)^{q+\xi}, \quad (\xi > -1). \quad (3.30)$$

$$J_{t_0}^q J_{t_0}^\alpha x(t) = J_{t_0}^{q+\alpha} x(t), \quad (\alpha \geq 0). \quad (3.31)$$

The nonlinear term can be decomposed as (X. Wang, Cao, et al., 2023):

$$\begin{cases} B_j^i = \frac{1}{i!} \left(\frac{d^i}{d\lambda^i} N[v_j^i(\lambda)] \right)_{\lambda=0} \\ v_j^i(\lambda) = \sum_{k=1}^i (\lambda)^k x_j^k \\ i = 0, 1, 2, \dots, \infty; j = 1, 2, 3, \dots, n \end{cases} \quad (3.32)$$

The nonlinear term N is denoted as:

$$Nx = \sum_{i=0}^{\infty} B^i(x^0, x^1, \dots, x^i) \quad (3.33)$$

The numerical solution of the system is expressed as:

$$x = \sum_{i=0}^{\infty} x^i = J_{t_0}^q \sum_{i=0}^{\infty} x^i + J_{t_0}^q \sum_{i=0}^{\infty} B^i + J_{t_0}^q c + x(t_0^+), \quad (3.34)$$

where the iterative relation for x_i is calculated as:

$$\begin{cases} x^0 = J_{t_0}^q c + x(t_0) \\ x^1 = J_{t_0}^q Lx^0 + J_{t_0}^q B^0(x^0) \\ x^2 = J_{t_0}^q Lx^1 + J_{t_0}^q B^1(x^0, x^1) \\ \vdots \\ x^i = J_{t_0}^q Lx^{i-1} + J_{t_0}^q B^{i-1}(x^0, x^1, \dots, x^{i-1}) \\ \vdots \end{cases} \quad (3.35)$$

The fractional-order form of the Memelements based hyperchaotic system is obtained by replacing the integer-order derivatives with Caputo fractional derivatives:

$$\left\{ \begin{array}{l} *D_{t_0}^q x = \frac{(a + b \sin u)y - x}{ef} - \frac{(c + d \cos w)z}{f} \\ *D_{t_0}^q y = \frac{x}{e} + \left(g - \frac{1}{e}\right)(a + b \sin u)y \\ *D_{t_0}^q z = x \\ *D_{t_0}^q u = y \\ *D_{t_0}^q w = z \end{array} \right. \quad (3.36)$$

where q is the fractional order of the system. Using the ADM algorithm, the linear and nonlinear terms of the FOMMS are decomposed as:

$$\begin{bmatrix} Lx \\ Ly \\ Lz \\ Lu \\ Lw \end{bmatrix} = \begin{bmatrix} \frac{ay}{(ef)} - \frac{cz}{f} \\ \frac{x}{e} + \left(g - \frac{1}{e}\right)ay \\ x \\ y \\ z \end{bmatrix}, \begin{bmatrix} Nx \\ Ny \\ Nz \\ Nu \\ Nw \end{bmatrix} = \begin{bmatrix} \frac{-bysin(u)}{(ef)} - \frac{dz \cos(w)}{f} \\ \left(g - \frac{1}{e}\right)bysin(u) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.37)$$

As per Equation 3.32 (X. Wang, Cao, et al., 2023), the system's nonlinear terms can be decomposed as:

$$\left\{ \begin{array}{l} B_1^0 = a + b \sin(x_1^0) \\ B_1^1 = a + b x_1^1 \cos(x_1^0) \\ B_1^2 = a + b(x_1^2 \cos(x_1^0) + 0.5 x_1^1 x_1^1 \cos(x_1^0)) \\ B_1^3 = a + b \left(x_1^3 \cos(x_1^0) + x_1^2 x_1^1 \sin(x_1^0) - \frac{(x_1^1 x_1^1 x_1^1 \cos(x_1^0))}{6} \right) \\ B_1^4 = a + b \left(x_1^4 \cos(x_1^0) + x_1^3 x_1^1 \sin(x_1^0) + 0.5 x_1^2 x_1^2 \sin(x_1^0) + 0.5 x_1^1 x_1^1 x_1^2 \sin(x_1^0) + \frac{x_1^1 x_1^1 x_1^1 \cos(x_1^0)}{24} \right) \end{array} \right. \quad (3.38)$$

$$\left\{ \begin{array}{l} B_2^0 = c + d \cos(x_1^0) \\ B_2^1 = c + d(-x_1^1 \sin(x_1^0)) \\ B_2^2 = c + d(-x_1^2 \sin(x_1^0) - 0.5 x_1^1 x_1^1 \cos(x_1^0)) \\ B_2^3 = c + d \left(-x_1^3 \sin(x_1^0) - x_1^2 x_1^1 \cos(x_1^0) + \frac{(x_1^1 x_1^1 x_1^1 \sin(x_1^0))}{6} \right) \\ B_2^4 = c + d \left(-x_1^4 \sin(x_1^0) - x_1^3 x_1^1 \cos(x_1^0) - 0.5 x_1^2 x_1^2 \cos(x_1^0) + 0.5 x_1^1 x_1^1 x_1^2 \sin(x_1^0) + \frac{x_1^1 x_1^1 x_1^1 \cos(x_1^0)}{24} \right) \end{array} \right. \quad (3.39)$$

The initial condition is provided as $[x(t_0^+), y(t_0^+), z(t_0^+), u(t_0^+), w(t_0^+)]$, and the first term can be expressed as:

$$\left\{ \begin{array}{l} x^0 = x(t_0^+) \\ y^0 = y(t_0^+) \\ z^0 = z(t_0^+) \\ u^0 = u(t_0^+) \\ w^0 = w(t_0^+) \end{array} \right. \quad (3.40)$$

By defining $c_0^1 = x^0, c_0^2 = y^0, c_0^3 = z^0, c_0^4 = u^0, c_0^5 = w^0$, the initial condition can be represented as $c^0 = [c_1^0 \ c_2^0 \ c_3^0 \ c_4^0 \ c_5^0]$. Utilizing the iteration relation and integral operator properties, it follows that:

$$\left\{ \begin{array}{l} x^1 = \frac{(a + b \sin c_4^0) c_2^0 - c_1^0}{ef} - \frac{(c + d \cos c_5^0)}{f} c_3^0 \frac{(t - t_0)^q}{\Gamma(q + 1)} \\ y^1 = \frac{c_1^0}{e} + \left(g - \frac{1}{e}\right) (a + b \sin c_4^0) c_2^0 \frac{(t - t_0)^q}{\Gamma(q + 1)} \\ z^1 = c_1^0 \frac{(t - t_0)^q}{\Gamma(q + 1)} \\ u^1 = c_2^0 \frac{(t - t_0)^q}{\Gamma(q + 1)} \\ w^1 = c_3^0 \frac{(t - t_0)^q}{\Gamma(q + 1)} \end{array} \right. \quad (3.41)$$

Let

$$\left\{ \begin{array}{l} c_1^1 = \frac{(ac_2^0 + bc_2^0 \sin c_4^0 - c_1^0)}{ef} - \frac{(cc_3^0 + dc_3^0 \cos c_5^0)}{f} \\ c_2^1 = \frac{c_1^0}{e} + \left(g - \frac{1}{e}\right) (ac_2^0 + bc_2^0 \sin c_4^0) \\ c_3^1 = c_1^0 nb \\ c_4^1 = c_2^0 \\ c_5^1 = c_3^0 \end{array} \right. \quad (3.42)$$

Therefore, x^1 can be expressed as $x^1 = c^1(t - t_0)^q/\Gamma(q + 1)$. Applying the same methodology, the decomposition of the remaining term coefficients is provided in Appendix A.

Therefore, the six-term numerical approximation of the fractional order hyperchaotic system can be derived as:

$$\left\{ \begin{array}{l} \tilde{x}_j(t) = c_j^0 + c_j^1 \frac{(t - t_0)^q}{\Gamma(q + 1)} + c_j^2 \frac{(t - t_0)^q}{\Gamma(2q + 1)} + c_j^3 \frac{(t - t_0)^q}{\Gamma(3q + 1)} + c_j^4 \frac{(t - t_0)^q}{\Gamma(4q + 1)} + c_j^5 \frac{(t - t_0)^q}{\Gamma(5q + 1)}, \\ j = 1, 2, 3, 4, 5 \end{array} \right. \quad (3.43)$$

The Fractional-Order Memelements-Based Hyperchaotic System (FOMBHCS) two-dimensional phase portraits in the x - y plane for a range of fractional-order parameter values are shown in Figure 3.28. The dynamic behavior of the system as q drops from 1 to 0.35 is depicted in the figure. The trajectory displays a distinct chaotic structure at $q=1$. The attractor stays intact as q drops, but it gradually deforms, displaying more density and compression in phase space. The trajectory experiences extreme divergence at $q=0.35$, indicating either a fundamental shift in the dynamics of the system or numerical instability, and the attractor is drastically changed for $q=0.505$. This implies that, depending on the selected q -value, the

fractional-order has a substantial impact on the chaotic behavior of the system, perhaps resulting in either chaotic suppression or divergence.

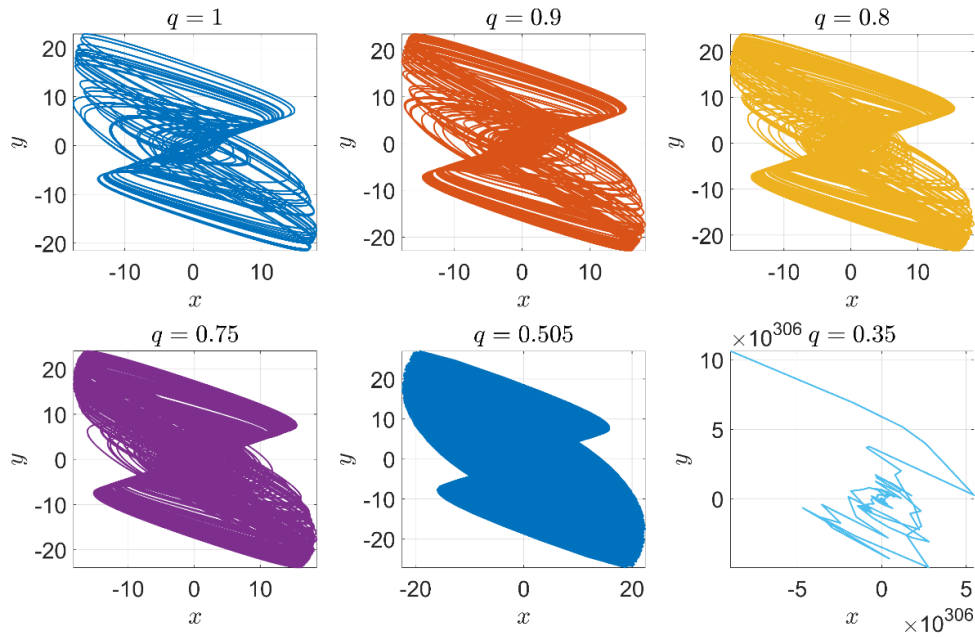


Figure 3.28: FOMBHCS 2D Phase Portrait of FOMBHCS x vs y for Different q Values

Commer

Setting the parameters as follows: $q = 0.505$, $a=0.41$, $b=2.81$, $c=10$, $d=3$, $e=0.2$, $f=1$, and $g=2$. The initial conditions are $(1, 1, 0, 0, 0)$, and the iteration step is $h = 0.001$. Under these conditions, Figure 3.29 illustrates the analysis of Lyapunov Exponents and Dimension of the FOMBHCS. In (a), the Lyapunov exponents stabilized throughout the running time for $q=0.505$, confirming the presence of hyperchaotic behaviour. In (b), the Lyapunov exponents varied with fractional order values of q , which shows transitions between different nonlinear dynamical states. In (c), the Lyapunov dimension stayed nearly constant, suggesting that the system maintains its complexity in different fractional orders.

Figure 3.30 illustrates the forward and reverse bifurcation diagrams of the FOMBHCS with respect to the parameter d . The forward bifurcation diagram (shown in blue) illustrates how the local maxima x_1 changes as d increases, highlighting chaotic transitions. In contrast, the reverse bifurcation diagram (depicted in red) shows the system's behaviour when d is decreased, revealing chaotic behaviour through different dynamical pathways. Together, these bifurcation diagrams enhance our understanding of the system's nonlinear characteristics and its bifurcation structure.

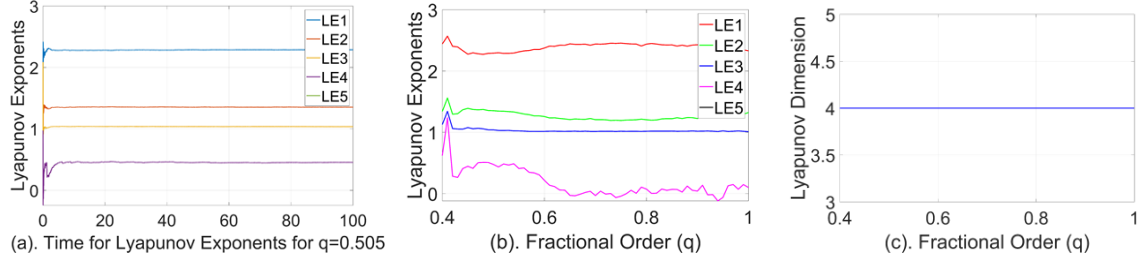


Figure 3.29: Lyapunov Exponents and Dimension of FOMBHCS: (a) Lyapunov Exponents vs Time for $q = 0.505$, (b) Lyapunov Exponents vs Fractional Order (q), (c) Lyapunov Dimension vs Fractional Order (q).

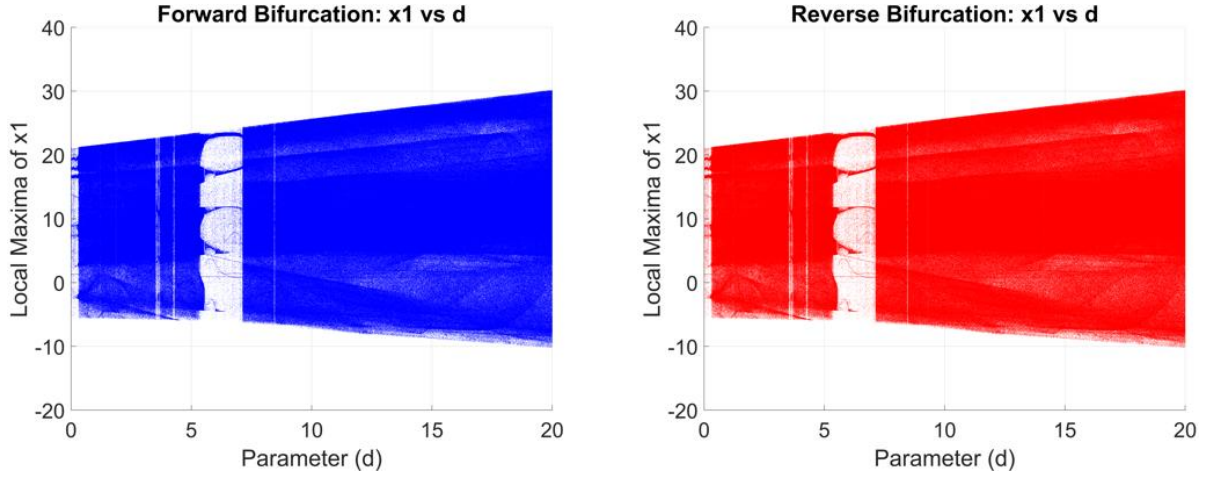


Figure 3.30: Forward and Reverse Bifurcation Diagram of the FOMBHCS for Parameter d

3.4.1. Dissipation and Stability of the System

This section analyses the dissipation and stability of the proposed Fractional-Order Memelements-Based Hyperchaotic System (FOMBHS). Dissipation is a critical property of chaotic systems, as it ensures that the system's phase space volume contracts over time, leading to the emergence of chaotic attractors. Stability analysis, on the other hand, helps us understand the system's behaviour around its equilibrium points. It is essential for ensuring the system's reliability in practical applications such as image encryption and cryptography. The chaotic properties of the system, including its high sensitivity to initial conditions and parameter variations, make it suitable for secure communication and encryption tasks.

Dissipation of the System: The dissipation of the system is determined by calculating the divergence of the system's vector field. For a dynamical system described by the state variables x, y, z, u, w , the divergence ∇V is given by:

$$\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{u}}{\partial u} + \frac{\partial \dot{w}}{\partial w} \quad (3.44)$$

For the proposed FOMBHS, the divergence is computed as:

$$\nabla V = \frac{-1}{(ef)} + \left(g - \frac{1}{e}\right)(a + b \sin(u)) \quad (3.45)$$

Substituting the system parameters $a=0.41$, $b=2.81$, $e=0.2$, $g=2$, and $f=1$, we obtain:

$$\nabla V = -5 + (2 - 5)(0.41 + 2.81 \sin(u)) \quad (3.46)$$

Since $\sin(u)$ ranges between -1 and 1, the term $0.41 + 2.81 \sin(u)$ varies between -2.40 and 3.22. Therefore, the divergence ∇V ranges between:

$$\begin{cases} \nabla V = -5 - 3(-2.4) = 2.20 & (\text{for } \sin(u) = -1) \\ \nabla V = -5 - 3(3.22) = -14.66 & (\text{for } \sin(u) = 1) \end{cases} \quad (3.47)$$

The system is dissipative when $\nabla V < 0$, which occurs when $\sin(u) > -\frac{0.41}{2.81} \approx -0.146$. This ensures that the system's phase space volume contracts over time, leading to the formation of chaotic attractors. The dissipative nature of the system is crucial for its application in image encryption, as it ensures the generation of complex and unpredictable chaotic signals that can effectively scramble image data.

Stability of the System: The system's stability is analyzed by examining its equilibrium points and the eigenvalues of the Jacobian matrix at these points. The equilibrium points are found by setting the time derivatives of the state variables to zero:

$$\begin{cases} \frac{(a + b \sin u)y - x}{ef} - \frac{(c + d \cos w)z}{f} = 0 \\ \frac{x}{e} + \left(g - \frac{1}{e}\right)(a + b \sin u)y = 0 \\ x = 0 \\ y = 0 \\ z = 0 \end{cases} \quad (3.48)$$

From the above equations, the system has a planar equilibrium point set $E(0, 0, 0, m, n)$, where m and n are arbitrary constants. The Jacobian matrix of the system at the equilibrium point $E(0, 0, 0, m, n)$ is given by:

$$J = \begin{bmatrix} \left(\frac{-1}{ef}\right) & \left(\frac{(a + b \sin m)}{ef}\right) & \left(-\frac{(c + d \cos n)}{f}\right) & 0 & 0 \\ \left(\frac{1}{e}\right) & \left(g - \frac{1}{e}\right)(a + b \sin m) & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.49)$$

Substituting the parameters $a=0.41$, $b=2.81$, $e=0.2$, $g=2$, $f=1$, $d=10$, and $d=3$, the Jacobian matrix becomes:

$$J = \begin{bmatrix} (-5) & (2.05 + 14.05 \sin m) & (-10 - 3 \cos n) & 0 & 0 \\ (5) & (-3(0.41 + 2.81 \sin m)) & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.50)$$

To find the eigenvalues, we solve the characteristic equation $\det(J - \lambda I) = 0$. The eigenvalues λ are the roots of this equation. The eigenvalues of the Jacobian matrix determine the stability of the equilibrium points. For a fractional-order system, the stability condition is given by (Matignon, 1996; X. Wang, Cao, et al., 2023):

$$|\arg(\lambda)| > \frac{\mu\pi}{2}, \quad \mu = \max(q_i), \quad \{i \in N | 1 \leq i \leq 5\} \quad (3.51)$$

where q_i are the fractional orders of the system, if all eigenvalues satisfy this condition, the system is stable.

For the equilibrium point $E(0, 0, 0, 0, 0)$, the Jacobean calculated as:

$$J_1 = \begin{bmatrix} -5 & 2.05 & -13 & 0 & 0 \\ 5 & -1.23 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.52)$$

And for the equilibrium point $E(0, 0, 0, -\pi/2, -\pi/2)$,

$$J_2 = \begin{bmatrix} -5 & -12 & -10 & 0 & 0 \\ 5 & 7.2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.53)$$

The characteristic equation is given by:

$$\det(J - \lambda I) = 0 \quad (3.54)$$

Substituting J and I (the identity matrix), we get:

$$\det_1 = \begin{bmatrix} -5 - \lambda & 2.05 & -13 & 0 & 0 \\ 5 & -1.23 - \lambda & 0 & 0 & 0 \\ 1 & 0 & -\lambda & 0 & 0 \\ 0 & 1 & 0 & -\lambda & 0 \\ 0 & 0 & 1 & 0 & -\lambda \end{bmatrix} = 0 \quad (3.55)$$

And

$$\det_2 = \begin{bmatrix} -5 - \lambda & -12 & -10 & 0 & 0 \\ 5 & 7.2 - \lambda & 0 & 0 & 0 \\ 1 & 0 & -\lambda & 0 & 0 \\ 0 & 1 & 0 & -\lambda & 0 \\ 0 & 0 & 1 & 0 & -\lambda \end{bmatrix} = 0 \quad (3.56)$$

This is a 5th-order determinant, and solving it analytically is complex. Instead, we can use numerical methods using Matlab software to compute the eigenvalues. Using a numerical solver, the eigenvalues for this Jacobian matrix are approximately:

$$\left\{ \begin{array}{ll} \lambda_1 = 0.0000 + 0.0000i(\text{zero real part}), \\ \lambda_2 = 0.0000 + 0.0000i(\text{zero real part}), \\ \lambda_3 = -5.0996 + 0.0000i(\text{negative real part}); & \text{for equilibrium point } E(0, 0, 0, 0, 0) \\ \lambda_4 = -0.5652 + 1.6781i(\text{negative real part}), \\ \lambda_5 = -0.5652 - 1.6781i(\text{negative real part}), \end{array} \right. \quad (3.57)$$

$$\left\{ \begin{array}{ll} \lambda_1 = 0.0000 + 0.0000i(\text{zero real part}), \\ \lambda_2 = 0.0000 + 0.0000i(\text{zero real part}), \\ \lambda_3 = 2.1273 + 0.0000i(\text{positive real part}); & \text{for equilibrium point } E(0, 0, 0, -\pi/2, -\pi/2) \\ \lambda_4 = 0.0363 + 5.8176i(\text{positive real part}), \\ \lambda_5 = 0.0363 - 5.8176i(\text{positive real part}), \end{array} \right.$$

Stability Analysis and Implications: The system exhibits instability at both equilibrium points, $E(0, 0, 0, 0, 0)$ and $E(0, 0, 0, -\pi/2, -\pi/2)$. This instability arises because the eigenvalues have either zero or positive real parts, which override the local stabilizing effects of the negative real parts. At $E(0, 0, 0, 0, 0)$, three eigenvalues, such as -5.0996 and $-0.5652 \pm 1.6781i$, indicate phase-space contraction. However, two eigenvalues with zero genuine parts result in a non-hyperbolic equilibrium, leading to neutral stability. Conversely, at $E(0, 0, 0, -\pi/2, -\pi/2)$, the eigenvalues display positive real parts (e.g., 2.1273 and $0.0363 \pm 5.8176i$), which drive exponential divergence. This behaviour is further complicated by eigenvalues that have zero real parts. For fractional-order systems, stability requires that all eigenvalues satisfy the condition $|\arg(\lambda)| > \mu\pi/2$. In this case, non-negative eigenvalues violate this condition, resulting in global instability (Matignon, 1996). This inherent instability reflects hyperchaotic dynamics, which can be vital for secure communication and encryption and necessitate external control for regulated applications. Thus, the interplay between stabilizing and destabilizing modes highlights the system's dual nature: it is locally contractive yet globally unstable, epitomizing hyperchaotic behaviour.

3.5. Optimization of Hyperchaotic systems

Hyperchaotic systems can be optimized by adjusting parameters to enhance Lyapunov exponents and expand chaotic bandwidth through bifurcation analysis. Traditional optimization methods like Genetic Algorithms (GA) struggle with slow convergence, while Particle Swarm Optimization (PSO) can get stuck in local optima. This study addresses these challenges using Chaotic Particle Swarm Optimization (CPSO), which adds chaotic maps to PSO's velocity equations to improve diversity. CPSO-optimized parameters include inertia weight (ω), cognitive coefficient (c_1), and social coefficient (c_2). This significantly improved

the degree of hyperchaos (h), increasing from 0.003313 to 0.005886, reflecting a 79.6% enhancement. This increase indicates greater dynamic complexity and unpredictability, which are beneficial for secure image encryption applications.

The proposed memristor-based hyperchaotic system, which is going to be optimized, is derived from the hyperchaotic circuit, as illustrated in Figure 3.31. This circuit consists of a memristor, two capacitors, a resistor, and an inductor. The mathematical model of the entire circuit is produced by using cubic memristor modelling (Sahin et al., 2020).

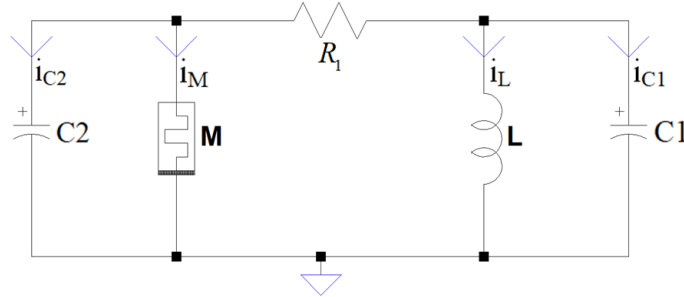


Figure 3.31: The Memristor-Based Hyperchaotic System Circuit

Using Kirchhoff's rules, the normalised set of equations for the circuit is presented as follows:

$$\begin{aligned}
 \dot{x} &= \alpha\beta y - \alpha\beta x - \alpha f z + \alpha x((K-1)f - e) \\
 \dot{y} &= \gamma\beta x - \gamma\beta y - \gamma y(n_0 + n_1 w^2) \\
 \dot{z} &= \delta f x(K-1) - \delta f z \\
 \dot{w} &= \varepsilon x
 \end{aligned} \tag{3.58}$$

Equation (3.58) is subjected to the numerical analysis using the Runge-Kutta ode23s method in MATLAB. The initial states are given as $[x, y, z, w] = [0.10, 0.0, 0.0, 0.0]$.

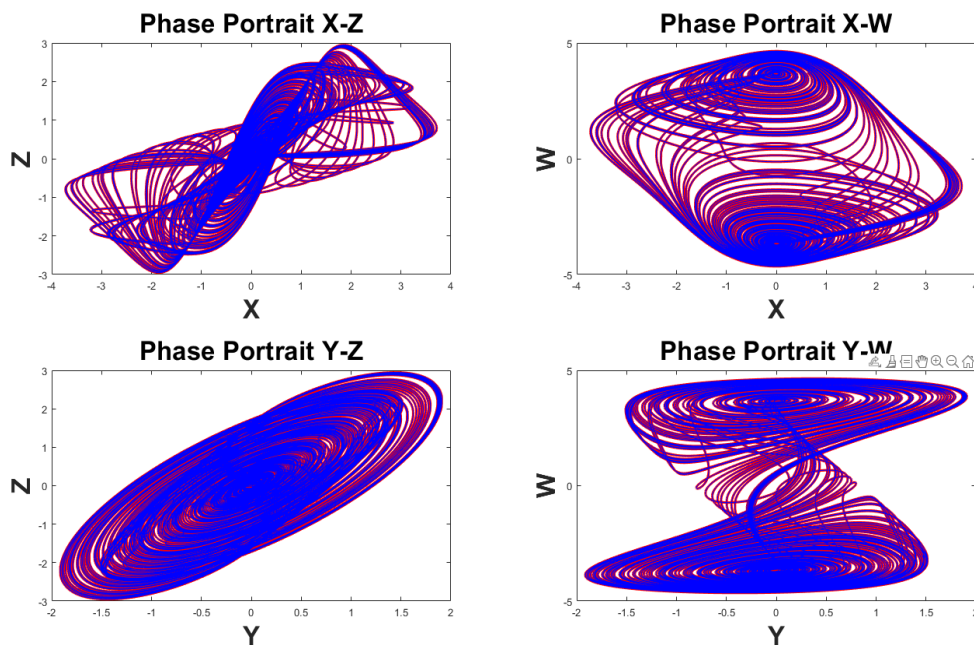


Figure 3.32: Phase Portraits of the Hyperchaotic Attractors.

Each of the parameter values which are capable of generating hyperchaotic dynamics is analyzed using bifurcation diagrams and the best parameters are selected as ($\alpha=1$, $\beta=1$, $\gamma=9.98$, $\delta=1$, $e=3$, $f=3.75$, $\varepsilon=-3$, $n_0=-1.2$, $n_1=0.1$, and $K=3.05$). Phase portraits (x_z , x_w , y_z , and y_w plane) of the hyperchaotic attractors for the improved hyperchaotic system based on memristor are displayed in Figure 3.32.

We examine several system parameters that serve as bifurcation parameters. To investigate the system's sensitivity to specific parameters, we assessed the diagrams of bifurcations by varying the parameter within a designated range while maintaining fixed values for all other parameters and the initial states. The bifurcation diagrams are displayed in Figure 3.33. The diagrams show that the modified memristor-based hyperchaotic system manifests considerable hyperchaos throughout each bifurcation interval.

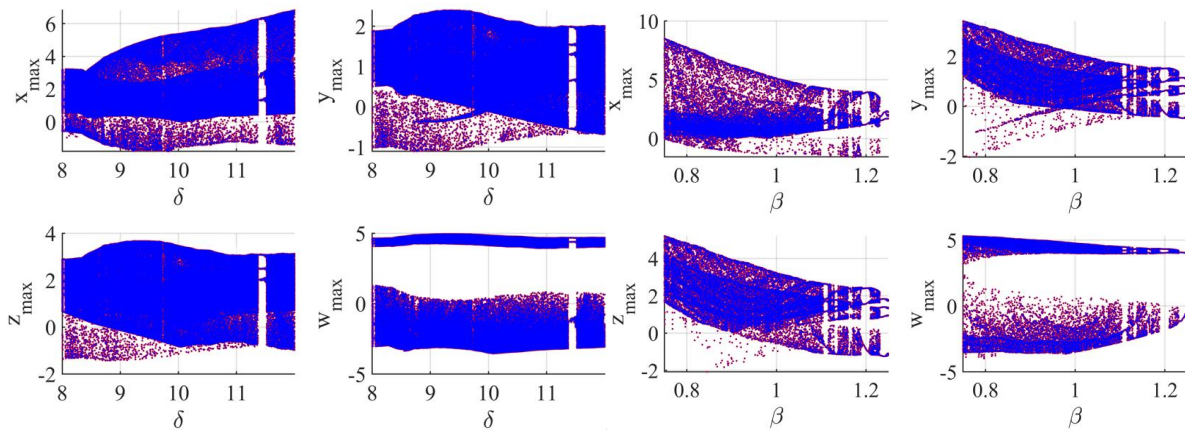


Figure 3.33: Bifurcation Diagrams of Hyperchaotic System for γ (8.0 to 12.0), β (0.75 to 1.25).

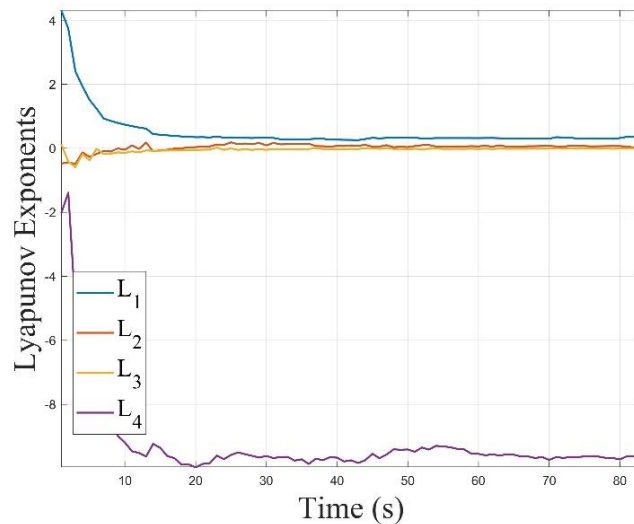


Figure 3.34: Phase Portraits of the Hyperchaotic Attractors.

The phase portraits and Lyapunov exponent diagrams indicate that the chaotic system exhibits a hyperchaotic state with two positive Lyapunov exponents ($L_1 = 0.349936$, $L_2 = 0.031855$,

$L3 = -0.010692$, and $L4 = -9.615182$), a large Lyapunov exponent, and good randomness. Figure 3.34 shows Lyapunov exponents over time, indicating a hyperchaotic system.

3.5.1. Optimizing the Hyperchaotic System Using CPSO

This study aims to optimize a four-dimensional memristive hyperchaotic system using chaotic particle swarm optimization (CPSO). The goal is to maximize the system's degree of hyperchaos. In the proposed hyperchaotic system, the degree of hyperchaos can be quantified by the ratio of the second Lyapunov exponent to the absolute value of the most negative Lyapunov exponent (C. Li & Sprott, 2014; Sahoo et al., 2024).

$$h = \frac{LE_2}{|LE_4|} \quad (3.59)$$

The system is defined by four first-order ordinary differential equations (ODEs), with 10 parameters: $p1, p2, p3, p4, p5, p6, p7, p8, p9$, and $p10$, including four initial values: $x0, y0, z0$, and $w0$. The Lyapunov exponents (LEs), LE_1, LE_2, LE_3, LE_4 , are computed using vibrational equations and the Gram-Schmidt orthonormalization method to ensure stability during integration. The system is considered hyperchaotic if $LE_1 > 0$ and $LE_2 > 0$, with the optimization objective to maximize the value of h .

The choice of hyperchaotic system parameters and CPSO tuning factors is critical for ensuring optimal system behavior. We selected the system parameters based on their ability to generate hyperchaotic dynamics, as verified through numerical simulations and bifurcation analysis. These values balance chaotic behavior and system stability, maximizing the degree of hyperchaos while preventing instability.

CPSO adjusts the particle swarm optimization (PSO) algorithm by combining a chaotic map to improve exploration and avoid premature convergence, improving global search capabilities (Miao et al., 2021). Each particle characterizes a potential solution, containing a set of system parameters, and is updated iteratively according to v_i :

$$v_i = \omega \cdot v_i + c_1 \cdot \text{rand}(\cdot) \cdot (p_i^{\text{best}} - x_i) + c_2 \cdot \text{rand}(\cdot) (g^{\text{best}} - x_i) \quad (3.60)$$

$$x_i = x_i + v_i \quad (3.61)$$

where v_i denotes the velocity, x_i is the position, and p_i^{best} and g^{best} are the personal and global best solutions, respectively. The parameters ω (inertia weight), c_1 (cognitive coefficient), and c_2 (social coefficient) regulate the balance between exploration and exploitation. The chaotic maps in CPSO improve convergence to optimal hyperchaotic solutions (El-Shorbagy et al.,

2022). The system's chaotic behavior is measured using Lyapunov exponents, calculated from the vibrational equations coupled with the system ODEs. The exponents are defined as:

$$\lambda_i = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{\|\delta X(t)\|}{\|\delta X(0)\|} \quad (3.62)$$

where λ_i represents the i -th Lyapunov exponent, and $\delta X(t)$ is perturbation vector. The Gram-Schmidt orthogonalization method ensures the orthogonality of perturbation vectors during integration. (Q. Zou et al., 2023).

The objective function maximizes the hyperchaotic degree (h), constrained by $LE2 > 0$ and $LE4 < 0$. The CPSO algorithm is initialized with specific parameter values (p_i) and initial conditions (IC_i), as represented by:

$$[p_1, p_2, \dots, p_{10}, IC_1, \dots, IC_4] = [9, 1, 1, 1, 3, 3.75, -1.2, 0.1, -3, 3.05, 0.001, 0, 0, 0] \quad (3.63)$$

This setup allows CPSO to optimize these values to achieve maximum h . Recent studies show that CPSO is highly effective for optimizing hyperchaotic systems, achieving substantial improvements in hyperchaotic behavior (Abd-El-Atty & Abd EL-Latif, 2023; Chaotic & Scheme, 2022; Hameed et al., 2023).

The CPSO tuning factors (inertia weight ω , cognitive coefficient c_1 , and social coefficient c_2) were chosen to enhance the exploration-exploitation balance. The choice of social coefficients is vital for preventing premature convergence and improving the global search capabilities of the CPSO algorithm. The CPSO optimization effectively increases the degree of hyperchaos, as reflected by the substantial improvement in the Lyapunov exponents and the hyperchaos degree (h). To calculate the percentage improvement, we use the following formula:

$$\text{Percentage Improvement} = \frac{h_{\text{opt}} - h_0}{h_0} \times 100 \% \quad (3.64)$$

The proposed system's initial Lyapunov exponents are (0.349936, 0.031855, -0.010692, -9.615182), while the optimized Lyapunov exponents are (0.266036, 0.063144, -0.051011, -10.727714), indicating a significant increase in the degree of hyperchaos. The degree of hyperchaos increases from $h_0 = 0.003313$ to $h_{\text{opt}} = 0.005886$, representing a 79.6% improvement achieved through parameter optimization, which aligns with recent findings in chaotic systems using CPSO. Further, the Lyapunov dimension at h_{opt} exceeds the original system's value. Detailed comparisons are provided in Table 3.1.

Figure 3.35 illustrates the effect of varying iterations on h . The graph shows that the optimized h increases significantly over iterations, peaking near the ninth, while the initial h remains low and constant.

Table 3.1: A Comparison Between the Original and Optimized Hyperchaotic System

Parameters	Original system	Maximized system
α	1	0.049017
β	1	0.202976
γ	9.98	8.106149
δ	1	0.484014
e	3	2.259293
f	3.75	6.124249
ε	-3	3.584593
n_0	-1.2	-1.586394
n_1	0.1	-0.833634
K	3.05	2.624039
x_0	0.001	0.954648
y_0	0	1.508127
z_0	0	0.144615
w_0	0	-1.215849
Results		
L_1	0.349936	0.266036
L_2	0.031855	0.063144
L_3	-0.010692	-0.051011
L_4	-9.615182	-10.727714
h	0.003313	0.005886

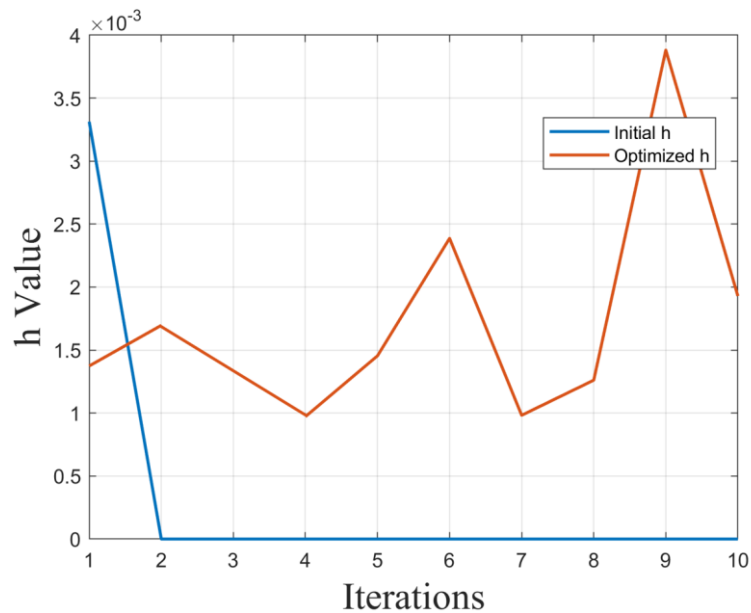


Figure 3.35: Hyperchaos Degree (h) Over Iterations

Figure 3.36 shows the time evolution of the Lyapunov exponents L_1 , L_2 , L_3 , and L_4 for the optimized system with $h = 0.0059$. The exponents stabilize over time, with L_1 and L_2 remaining positive, indicating sustained hyperchaotic behavior.

3.6. Classical One-Dimensional Chaotic Maps and Hybrid One-Dimensional Maps

This section provides an overview of one-dimensional (1D) chaotic maps and key chaos theory concepts, highlighting their simplicity and efficiency. While hybrid versions exhibit enhanced chaotic behaviour, their limitations necessitate integration with higher-dimensional systems for improved applications like secure encryption.

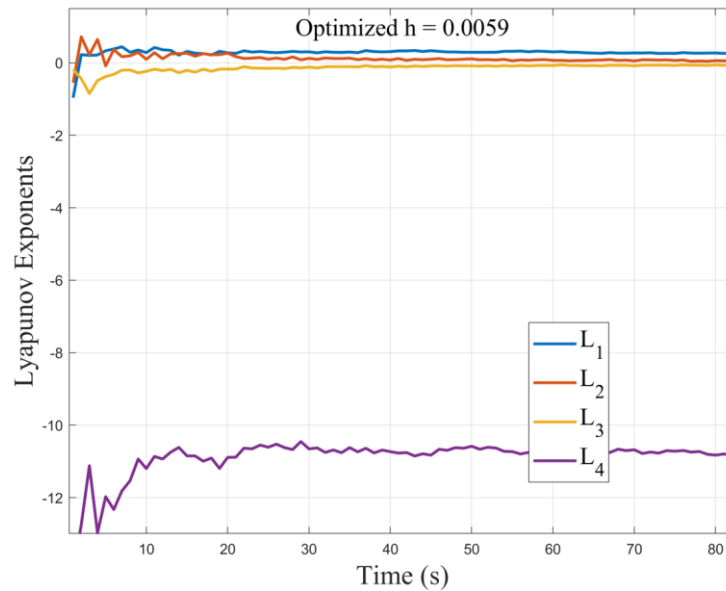


Figure 3.36: Optimized Lyapunov Exponent's Dynamics

3.6.1. Logistic Map

It is a chaotic map with one dimension, which is now a mostly utilized system in encryption systems. The logistic map is employed to increase security. The security of the communication system has been enhanced through multiple-stage encryption and decryption procedures that combine the hyperchaotic oscillator and the logistic map.

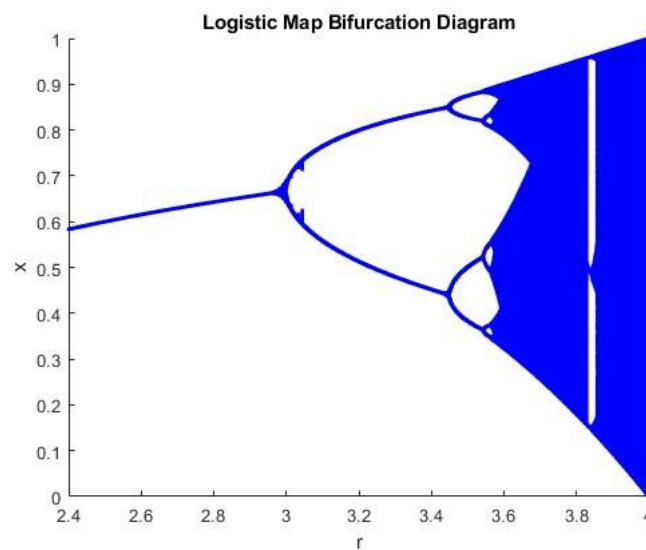


Figure 3.37: Bifurcation Diagram of the Logistic Map

A logistic map that is taken into consideration in (B. Chen et al., 2022) and calculated using Equation (3.65).

$$x_n = r * x_{n-1}(1 - x_{n-1}) \quad (3.65)$$

where, $0 \leq x_{n-1} \leq 1$ and $0 \leq r \leq 4$

Figure 3.37 shows a bifurcation diagram for the logistic system, as given by Equation (3.65). The iteration value of the mapping approaches a constant value when the value r is smaller than 3. The range of iteration values gets progressively more significant as the amount of r approaches 4. Using an initial value of $x_0 \in (0,1)$. When the value of r is between 3 and 3.569946, the produced numbers are discovered to be the periodic sequence. The period of infinity is discovered at 3.569946, and the system exhibits chaotic behaviour when the value is between 3.569946 and 4. Figure 3.37 illustrates the bifurcation of the logistic map [17, 18].

3.6.2. Tent Map

The Tent Map is a piecewise-linear, one-dimensional chaotic map defined by two distinct cases based on the current state value:

$$X_{n+1} = \begin{cases} \mu X_n & \text{if } X_n < 0.5 \\ \mu(1 - X_n) & \text{if } X_n \geq 0.5 \end{cases} \quad (\mu \in (1, 2], X_n \in [0, 1]) \quad (3.66)$$

Chaotic for $\mu=2$, generating uniform distributions. Limitations: Dynamical degradation (periodicity in digital systems) and vulnerability due to linearity. Used in basic encryption and pseudo random noise generation but requires hybridization for robust security.

3.6.3. Sine Map

The Sine map, a typical one-dimensional chaotic mapping, is used in this work as both a one-dimensional and two-dimensional sine map. This enhances the strength of the proposed image encryption algorithm and serves as an anti-clipping method.

The 1D Sine map is a one-dimensional iterative map that exhibits chaotic behaviour (Alexan, Gabr, et al., 2023). The recursive equation defining a chaotic map is expressed as:

$$X_{n+1} = r \sin(\pi X_n) \quad (3.67)$$

The scale factor r , must be greater than 1, with the initial value X_0 set within $[-r, +r]$, here $X_0 = 0.1$. Validation is done via the bifurcation diagram and Lyapunov exponent (S. Yan et al., 2023).

The 2D sine map Sine-I and Sine-II chaotic system, characterized by two input states, X_n and Y_n , and two cross output states, X_{n+1} and Y_{n+1} as defined in Equation (3.68). These outputs are used for row and column shifts to enhance the encryption algorithm and prevent clipping.

$$\begin{cases} X_{n+1} = r_1 \sin(\pi X_n) \\ Y_{n+1} = r_2 \sin(\pi Y_n) \end{cases} \quad (3.68)$$

where r_1 and r_2 are parameters that control the chaotic behaviour of the maps.

Sine maps have limitations, such as a restricted range of chaotic parameters and reduced sensitivity to initial conditions. Despite these issues, they are utilized in basic chaos-based encryption and pseudorandom number generators (PRNGs), and can be combined with higher-dimensional systems to enhance security.

3.6.4. Cubic Map

A 1D smooth chaotic map defined by:

$$X_{n+1} = \alpha X_n (1 - X_n^2), \quad (\alpha \in [2.3, 3.0], X_n \in [0, 1]) \quad (3.69)$$

The map is chaotic for $\alpha \approx 2.59$, producing complex bifurcations and non-uniform distributions. Limitations involve a narrow range of chaotic parameters, sparse chaotic regions, and degradation in finite-precision systems. While useful for basic encryption and PRNGs, these applications need to be hybridized with higher-dimensional systems or mem-elements for robust security.

3.6.5. General Limitations of 1D Chaotic Maps

- Low Dimensionality: Simple structure makes them vulnerable to phase-space reconstruction attacks.
- Limited Entropy: Small key space and repetitive patterns reduce unpredictability.
- Finite Precision Degradation: Digital implementations (e.g., FPGA, software) suffer from short periodic orbits and dynamical degradation.
- Security Weaknesses: Vulnerable to brute-force, statistical, and chosen-plaintext attacks when used standalone.

We have explored novel designs for hybrid one-dimensional chaotic maps, such as coupled or perturbed maps, aimed at enhancing complexity.

3.6.6. Logistic-Sine Adjusted Integrated Map

Traditional one-dimensional chaotic maps have limitations in terms of robustness, particularly when generating secure key sequences in encryption systems. To address these challenges, we propose a novel chaotic map, the Logistic-Sine Adjusted Integrated Map (LSAIM), which integrates the logistic and sine maps to enhance the Lyapunov exponent and generate more complex, unpredictable key sequences. This improvement strengthens image encryption and improves resilience against cryptographic attacks.

The logistic map (Rashid & Hussein, 2023), denoted as $f_{Log}(y)$, and the sine map (Gebereselassie & Roy, 2024), denoted as $f_{Sin}(y)$, are defined as follows:

$$f_{Log}(y) = r_1 y_i(1 - y_i) \quad (3.70)$$

$$f_{Sin}(y) = r_2 \sin(\pi y_i) \quad (3.71)$$

The integrated map, known as the LSAIM, is formed as:

$$f_{LSAIM}(y) = f_{Log}(f_{Sin}(y)) \quad (3.72)$$

Substituting the sine map expression into the logistic map results in the following formulation:

$$f_{LSAIM}(y) = r_1 r_2 \sin(\pi y_i) - r_1 r_2^2 \sin^2(\pi y_i) \quad (3.73)$$

Here, r_1 and r_2 are control parameters that regulate the behavior of the maps. In typical applications, both parameters are confined to the range $[0, 4]$. However, to explore the system's dynamic behavior more comprehensively, we extended the ranges to $r_1 \in [0, 4]$ and $r_2 \in [0.2, 6]$ for this study. This extended range allows for a deeper analysis of the system's chaotic behavior and potential for cryptographic applications.

Bifurcation diagram (Figure 3.38): the bifurcation diagrams offered in Figure 3.38 illustrate the system's conversion into chaotic dynamics through different values of r_2 . When $r_2 = 0.05$, the system shows linear behavior. At $r_2 = 0.09$, oscillatory dynamics begin, while for $r_2 \in [0.5, 2.5]$, the system shows multi-stable chaotic behavior. As r_2 beyond 2.5, the system enters fully chaotic dynamics, with increasing complexity as r_2 rises further.

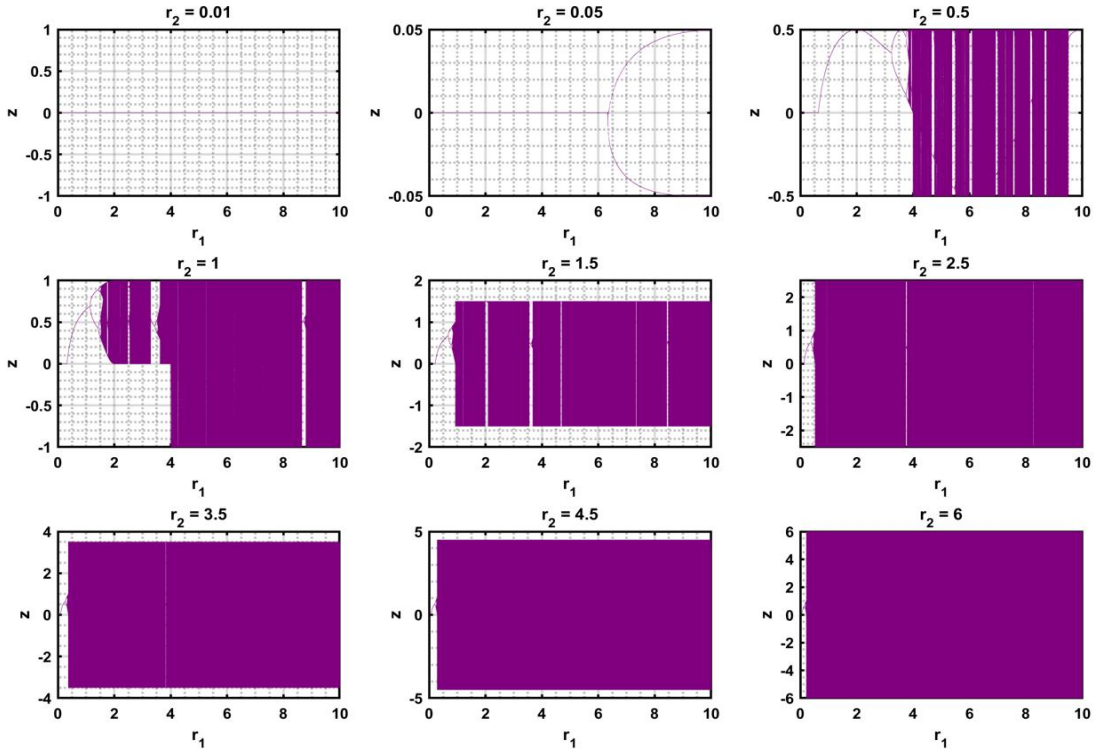


Figure 3.38: Logistic-Sine Integrated Map Bifurcation Diagrams for Different r_2 Values

Lyapunov Exponent (Figure 3.39): The Lyapunov exponent (LE) provides insight into the system's sensitivity to initial conditions. A positive LE ($LE > 0$) indicates chaotic behaviour, with larger values signifying more complex and unpredictable dynamics (M. Zhao et al., 2024). Figure 3.39 illustrates the Lyapunov exponent's steady increase as r_2 increases, confirming the system's transition to higher levels of chaos. Larger values of the Lyapunov exponent signify greater sensitivity to initial conditions, characteristic of more complex and unpredictable chaotic dynamics.

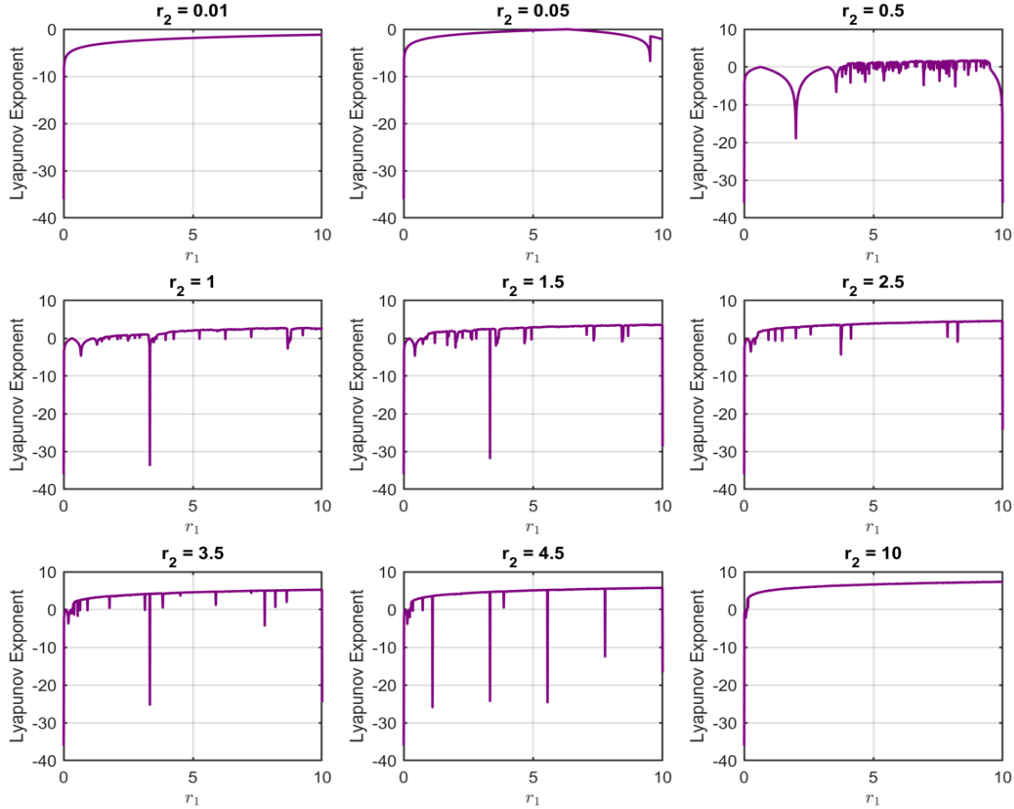


Figure 3.39: Lyapunov Exponent of LSIM for Different r_2 Values

3.6.7. Logistic-Sine-Cubic Iterative Map

The LSCIM is a hybrid one-dimensional strong chaotic system that combines the logistic, sine, and cubic maps. These maps separately give different chaotic properties that, when combined, generate a strong chaotic system with a wide range of dynamical behaviours. The mathematical model of the logistic map is represented by:

$$z_{n+1} = r_1 z_n (1 - z_n) \quad (3.74)$$

where z_n is the variable, and r_1 is a constant parameter that controls the dynamics. This map shows period-doubling bifurcations as r_1 increases. The sine map dynamics is given by:

$$z_{n+1} = r_2 \sin(\pi z_n) \quad (3.75)$$

where r_2 is the parameter that controls the dynamics. This map shows a quasi-periodic and chaotic behavior as the parameter changes. The cubic map dynamics is formulated as:

$$z_{n+1} = r_3 z_n^3 \quad (3.76)$$

where r_3 is a constant parameter that controls the degree of chaos. The cubic term increases divergence in the chaotic system and introduces new dynamic features. To get the LSCIM, start with the logistic map. Next, apply the sine map to the output of the logistic map:

$$z_{n+1} = r_2 \sin(\pi r_1 z_n (1 - z_n)) \quad (3.77)$$

Now apply the cubic map to the output of the sine map:

$$z_{n+1} = (r_2 \sin(\pi r_1 z_n (1 - z_n)))^3 \quad (3.78)$$

Finally, add the additional cubic map term. The LSCIM combines the above three maps into a single iterative equation:

$$z_{n+1} = (r_2 \sin(\pi r_1 z_n (1 - z_n)))^3 + r_3 r_2 \sin(\pi r_1 z_n (1 - z_n)) \quad (3.79)$$

Thus, this final form represents a hybrid system where the dynamics of all three maps interact, leading to various possible behaviors.

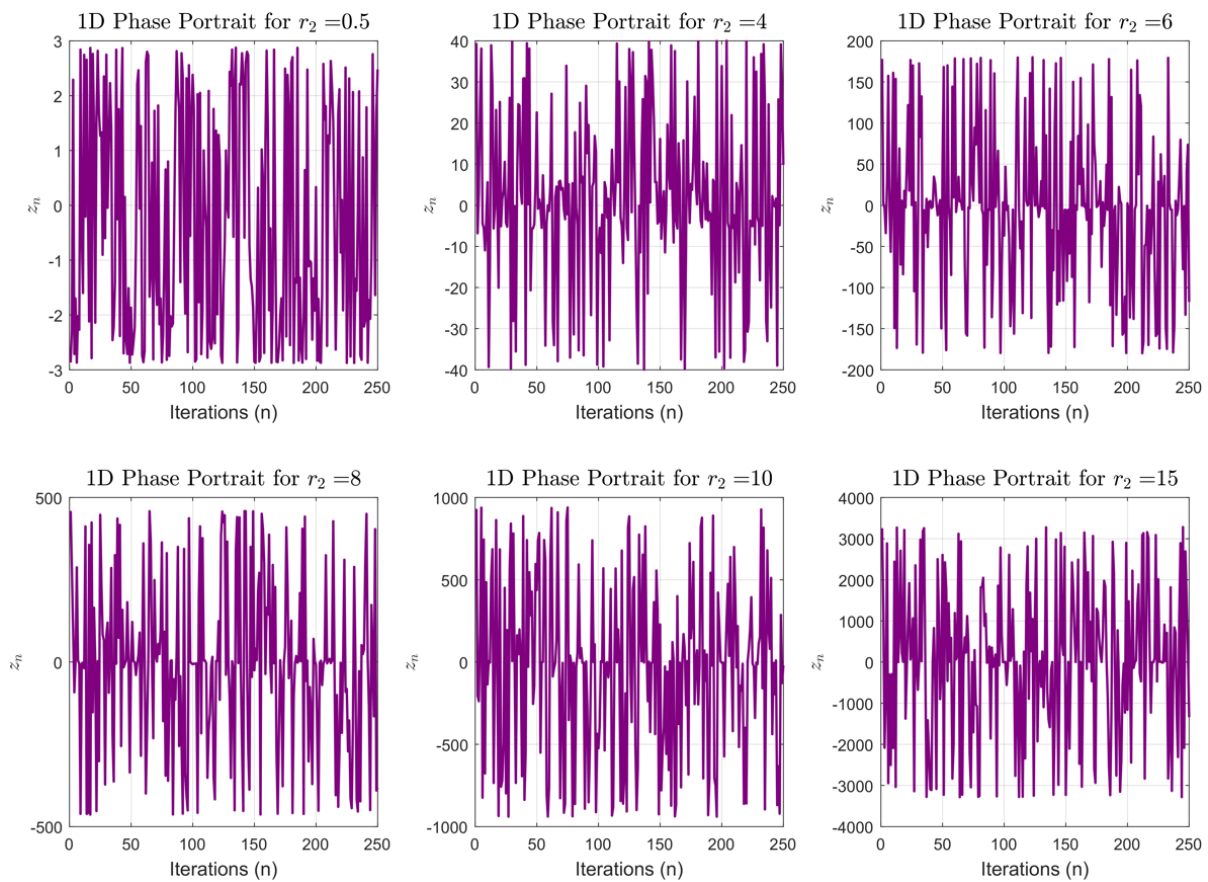


Figure 3.40: 1D Phase Portraits of a Dynamical System for Varying Parameter r_2

Figure 3.40 illustrates the 1D phase portraits of a dynamical system for varying values of the parameter r_2 , showing how the state variable z_n evolves over 250 iterations. Each subplot represents the system's behavior for a specific r_2 , highlighting its transition from periodic or stable dynamics ($r_2 = 0.5$) to increasingly chaotic and nonlinear behavior ($r_2 = 15$) as r_2 increases.

This combination of logistic, sine, and cubic terms increases the system's capacity to undergo various bifurcations as the parameters are modified, resulting in periodic, quasi-periodic, and chaotic regions. The bifurcation diagram, an essential tool in chaos theory, is used to visualize these transitions. And also, the Lyapunov exponent offers a quantitative degree of chaos, showing whether nearby trajectories diverge or converge exponentially. The Lyapunov exponent λ is given as:

$$\lambda = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=M+1}^N \ln \left| \frac{df(z_n)}{dz_n} \right| \quad (3.80)$$

where $f(z)$ is the LSCIM map function, N is the total number of iterations, and M is the number of transient iterations skipped to eliminate initial condition dependencies. Positive values of λ confirm chaotic behaviour, while negative values indicate stable or periodic orbits.

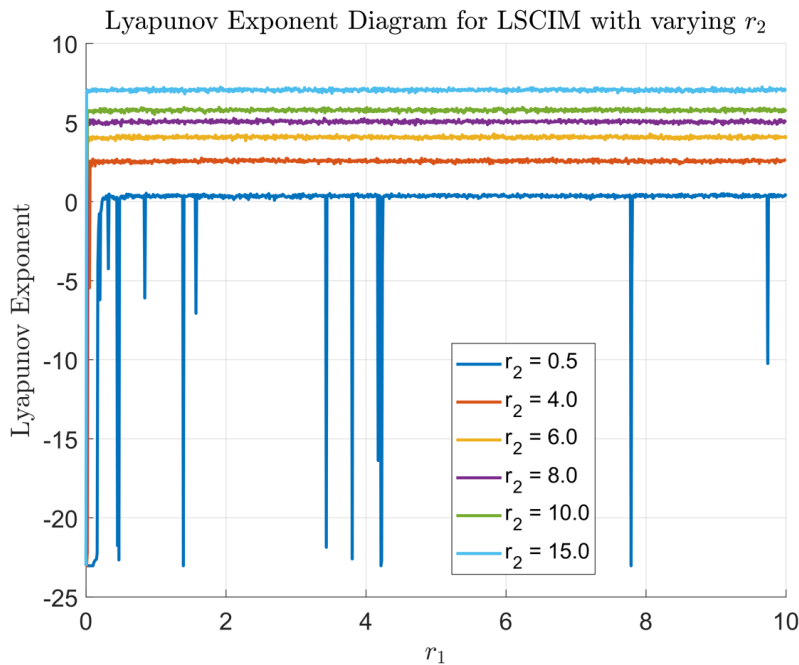


Figure 3.41: Lyapunov Exponent Diagram of LSCIM System with Varying r_1 and Fixed r_2 Values

Figure 3.41 demonstrates the Lyapunov exponents of the LSCIM chaotic map for varying r_1 values across fixed r_2 parameters ($r_2 = 0.5, 4.0, 6.0, 8.0, 10.0, \text{ and } 15.0$). The Lyapunov

exponent measures the system's sensitivity to initial conditions, where positive values indicate chaotic behavior and negative values signify stability or convergence. As r_1 increases, the system transitions between stable and chaotic regions, mainly for lower r_2 values (e.g., $r_2 = 0.5$), where fluctuations between negative and positive exponents are more frequent. For higher r_2 values (e.g., $r_2 = 15$), the exponents stabilize in the positive range, indicating a predominantly chaotic system. This suggests that the system's chaotic behavior intensifies as r_2 increases.

Figure 3.42 shows the bifurcation diagrams of the dynamical system for varying values of r_2 with r_1 as the bifurcation parameter. Each subplot illustrates the behaviour of the state variable z_n as r_1 changes for a fixed r_2 . At low r_2 values ($r_2 = 0.01, 0.05, 0.5$), the system exhibits transitions from stable fixed points to bifurcations and chaotic regions. As r_2 increases ($r_2 = 1.5, 3, 4$), the diagrams reveal broader chaotic regions with more considerable z_n variations, indicating intensified nonlinearity. The system becomes predominantly chaotic for high r_2 values ($r_2 = 6, 10, 15$), with z_n spanning significantly larger ranges and no apparent periodicity. The figure highlights how increasing r_2 amplifies the system's complexity and chaotic behaviour.

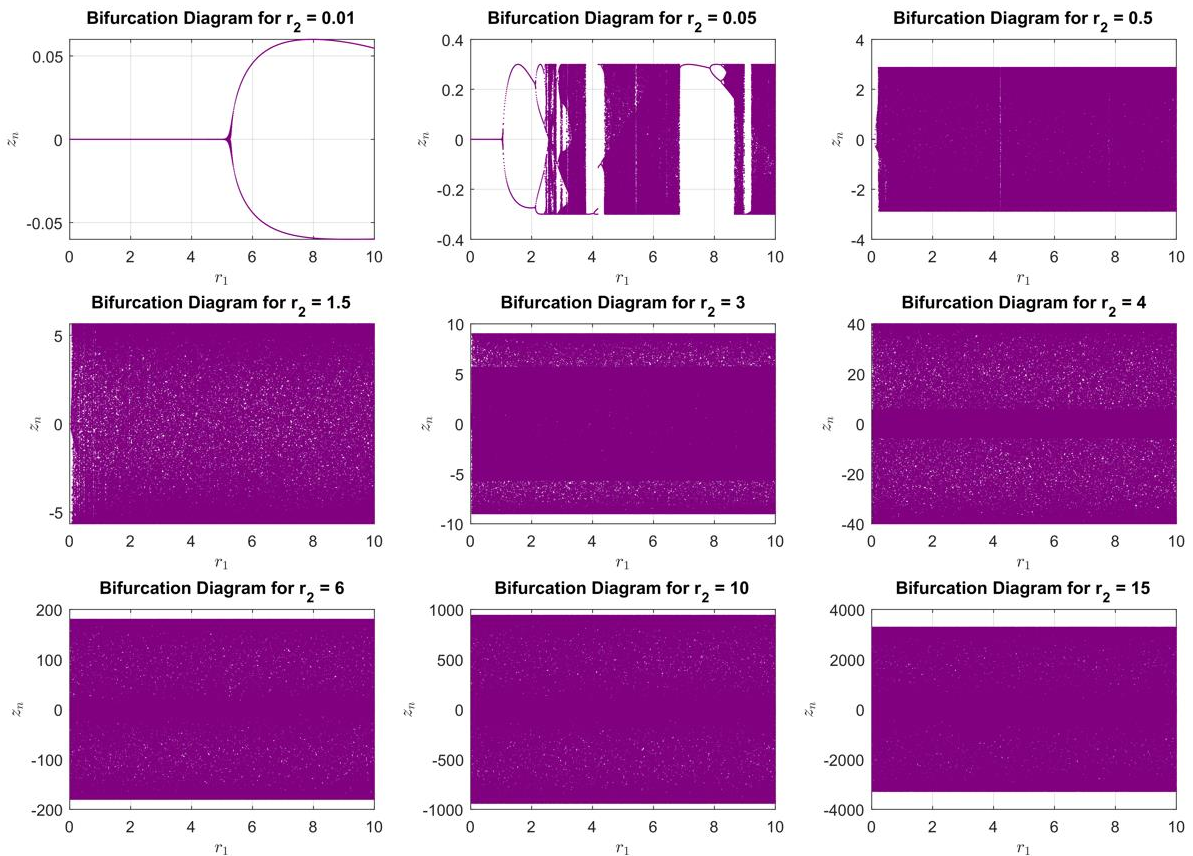


Figure 3.42: Bifurcation Diagrams of the Dynamical System for Varying r_2 and r_1

The Logistic-Sine-Coupled Iterative Map (LSCIM) is a nonlinear system exhibiting transitions from stability to chaos, driven by parameters r_1 and r_2 . Its rich dynamics, characterized by bifurcations, chaotic behaviour, and sensitivity to initial conditions, make it valuable for applications like encryption, secure communications, and random number generation.

3.6.8. Sine-Logistic Integrated Map

Chaotic maps generate random key sequences; however, traditional one-dimensional maps often lack security. We propose a hybrid chaotic map combining sine and logistic maps. This combination increases the Lyapunov exponent and enhances security, resulting in more complex key sequences. These improvements strengthen image encryption systems and improve resistance to attacks. Let $f_{Sin}(y)$ indicate the mathematical expression of the sine map (Gebereselassie & Roy, 2024) and $f_{Log}(y)$ represent the mathematical model of the logistic map (Rashid & Hussein, 2023). The integrated system, referred to as the Sine-Logistic integrated map (SLIM), is expressed as $f_{LSIM}(y)$. The mathematical derivation and formulation of the LSIM can be described as follows:

$$f_{Sin}(y) = r_2 \sin(\pi y_i) \quad 3.81$$

$$f_{Log}(y) = r_1 y_i(1 - y_i) \quad 3.82$$

$$f_{LSIM}(y) = f_{Log}(f_{Sin}(y)) \quad 3.83$$

$$f_{LSIM}(y) = r_1 r_2 \sin(\pi y_i) - r_1 r_2^2 \sin^2(\pi y_i) \quad 3.84$$

where r_1 and r_2 are constant control parameters regulating the dynamics of the maps.

The Lyapunov Exponent (LE) determines whether a system exhibits chaos. A positive LE ($LE > 0$) indicates chaotic behavior, with larger values signifying more complex and unpredictable dynamics (M. Zhao et al., 2024).

Figure 3.43 shows bifurcation diagrams of the variable z as a function of r_1 for various r_2 values, highlighting the transition from stability to chaos. The system displays stable behaviour with a fixed point at low r_2 (e.g., $r_2 = 0.05$). As r_2 increases (e.g., $r_2 = 0.5$), periodic oscillations develop, eventually leading to chaotic behavior with dense, irregular trajectories. For higher r_2 values (e.g., $r_2 = 5, 8, 12$), the system becomes highly sensitive to initial conditions, resulting in fully chaotic dynamics. The chaotic regions observed at higher r_2 values in Figure 3.44 indicate the potential for creating complex, noise-like signals for secure message transmission, making this system suitable for cryptographic applications.

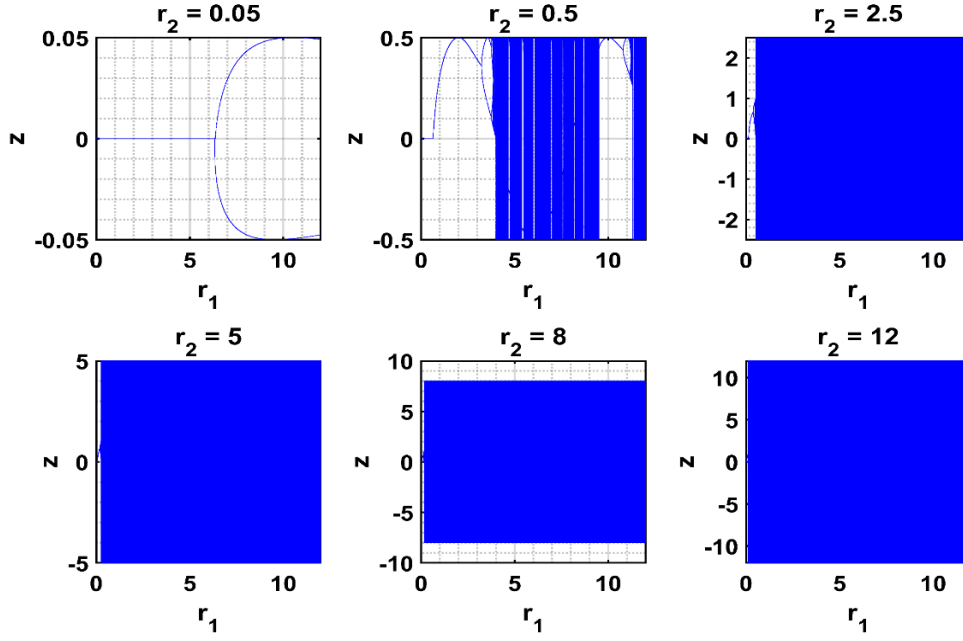


Figure 3.43: Logistic-Sine Integrated Map Bifurcation Diagrams for Different r_2 Values

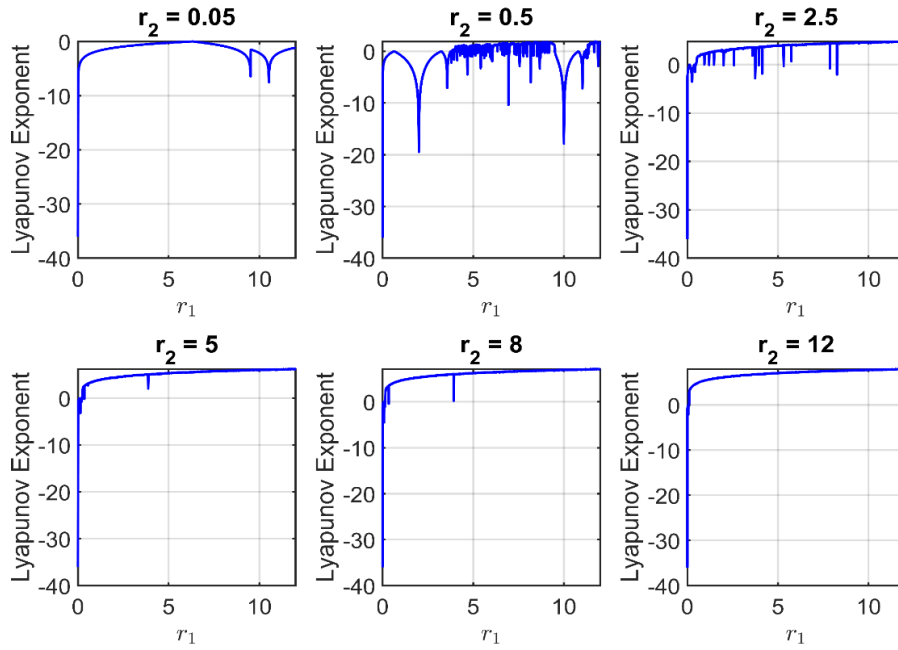


Figure 3.44: Lyapunov Exponent of LSIM for Different r_2 Values

3.7. Summary

This chapter discusses advanced chaotic and hyperchaotic systems for designing image encryption frameworks. The hyperchaotic systems developed in this study incorporate mem-elements, such as memristors, memcapacitors, and meminductors, to enhance nonlinearity, introduce memory-driven dynamics, and achieve multistability. Six integer-order mem-

element-based hyperchaotic systems are created, showing an 80% improvement in hyperchaotic behaviour through the application of Chaotic Particle Swarm Optimization (CPSO). Additionally, the study examines fractional-order extensions using Caputo derivatives to enhance chaos further. While classical one-dimensional chaotic maps—such as logistic maps, tent maps, sine maps, and cubic maps—have certain limitations, hybrid maps like LSAIM and LSCIM offer greater chaos and complexity, making them more suitable for image encryption frameworks. The mem-element-based hyperchaotic systems developed in this study aim to create robust image encryption frameworks that address the limitations of existing image encryption algorithms.

3.8. Materials

This research used various image datasets, software tools, and hardware resources to implement frameworks for image compression, encryption, and denoising. Both simulation and circuit-level modelling were employed to endure robustness.

3.8.1. Datasets

Experiments used benchmark images from the USC-SIPI Image Database and public repositories on Kaggle. Standard test images included Lena, Peppers, Baboon, Airplane, Air Base, Crown-Splash, and Mandrill, along with medical datasets for Glaucoma Detection and Breast Cancer Diagnosis (Normal, Malignant, Benign). Resolutions ranged from $256 \times 256 \times 3$ to $3185 \times 2709 \times 3$, with most experiments using 512×512 images. A total of 2,660 noisy samples were created by applying seven types of noise across 76 noise levels, normalized between 0 and 1 for model compatibility.

3.8.2. Software and Hardware Setup

Algorithmic implementations were carried out using MATLAB R2023a on a Windows system with an Intel® Core™ i7 CPU at 3.20 GHz and 8 GB of RAM. MATLAB was utilized for deep learning, modelling differential equations, performing DNA-based encryption, and assessing image quality.

3.8.3. Circuit Simulation

LT-spice was used to simulate memristive, memcapacitive and meminductive hyperchaotic circuits, verifying their dynamic behavior and nonlinear characteristics. This ensured alignment with theoretical models in MATLAB, highlighting the potential for real-world analogue hardware implementation of the encryption systems.

CHAPTER 4

IMAGE ENCRYPTION USING A MEMELEMENT-BASED HYPERCHAOTIC SYSTEM INTEGRATED WITH DNA ENCODING SCHEMES

4.1 Introduction

As discussed in the literature review of this study, traditional encryption techniques often fall short when applied to images due to their unique properties, such as high correlation and large data quantities. This chapter proposed a novel image encryption method that integrates principles from DNA rules, the logistic map, and a hyperchaotic model with memcapacitors to address these challenges. The intricate and unpredictable nature of the memcapacitor-based hyperchaotic system, coupled with deficiencies in current image encryption algorithms, catalyzes the development of this new algorithm. The proposed encryption strategy leverages the encoding properties of DNA and the complex dynamics of hyperchaotic systems to achieve enhanced image encryption. The method's effectiveness is demonstrated through simulations and evaluations using reference images. It surpasses current techniques regarding statistical robustness, resistance to brute-force attacks, and protection against differential attacks. The study concludes that combining DNA principles, logistic maps, and memcapacitor-based hyperchaotic systems significantly improves image encryption. This approach enhances data security for image transmission in open communication channels, addressing the limitations of traditional encryption techniques. As digital connectivity expands globally, the demand for robust image encryption is expected to increase, making this research pivotal in establishing standards for future advancements in image encryption algorithms.

4.2 Proposed Image Encryption Algorithm

In developing this image encryption algorithm, three two-dimensional matrices are produced from the colour image, significantly decreasing the space and time needed. Following encryption, rows and columns are again switched around; the split matrices channels are combined to generate a cyphered colour image. A two-dimensional matrix is split into blocks for DNA coding, and DNA operation rules for every block are dictated by the sequence that the hyperchaotic system produces. Then, the memcapacitive-based hyperchaotic system's initial value is linked to the original image, ensuring the "one image, one secret" effect and significantly increasing it, enhancing the system's security against attacks. Figure 4.1 shows the flow of the encryption algorithm.

4.2.1 The Encryption Algorithm's Specific Steps

The following steps outline the detailed encryption algorithm for securing a digitized image using a combination of logistic mapping, DNA encoding, and a memcapacitor-based hyperchaotic system.

Step 1: Image Splitting into RGB Channels

The digitalized image S of size $L \times M$ is split into three two-dimensional matrices S_1 , S_2 , and S_3 , corresponding to the Red (R), Green (G), and Blue (B) channels, respectively by using equation (4.1).

$$\begin{cases} S_1 = S(:, :, 1) \\ S_2 = S(:, :, 2) \\ S_3 = S(:, :, 3) \end{cases} \quad (4.1)$$

Step 2: Data Transformation for Computational Universality

To ensure the computational universality of the encryption process, the image dimensions L and M are added to be multiples of t , where t is the block size. This is done using the following conditions:

$$\begin{cases} \text{mod}(L, t) = 0 \\ \text{mod}(M, t) = 0 \end{cases} \quad (4.2)$$

The matrices S_1 , S_2 , and S_3 are then zero-padded as necessary and partitioned into blocks of size $t \times t$, as shown by equation (4.2).

Step 3: Acquire the chaotic sequence

A chaotic sequence is generated using a logistic map. The parameter r is set to 3.9999, and the initial value x_0 is computing using equation (4.3).

$$x_0 = \frac{\text{sum}(S_1(:)) + \text{sum}(S_2(:))}{255 \times L \times M \times 2} \quad (4.3)$$

where, $\text{sum}(S_1(:))$ and $\text{sum}(S_2(:))$ represent the sum of all pixel values in the R and G channels, respectively.

Step 4: Sequence Conversion for DNA Operation

The chaotic sequence n_i is converted into a $L \times M$ matrix, with values transformed to fall between 0 and 255:

$$\begin{cases} n = \text{mod}(\text{round}(n \times 10^4), 256) \\ R_s = \text{reshape}(n, M, N)' \end{cases} \quad (4.4)$$

Step 5: Memcapacitive Hyper Chaotic System Computation and Data Manipulation

Using Matlab's ODE45 function, the Hyperchaotic Memcapacitor system generates four sequences $X_i = q_{cm}(0)$, $Y_i = i_1(0)$, $Z_i = i_2(0)$, and $W_i = i_2(0)$. The initial values for these sequences are computed using the equation (4.5) as follows:

$$\begin{cases} q_{cm}(0) = \frac{\text{sum}(\text{sum}(b_i \tan d(S_1, 17)))}{(17 \times L \times M)} \\ i_1(0) = \frac{\text{sum}(\text{sum}(b_i \tan d(S_2, 34)))}{(34 \times L \times M)} \\ i_2(0) = \frac{\text{sum}(\text{sum}(b_i \tan d(S_3, 68)))}{(68 \times L \times M)} \\ \sigma(0) = \frac{\text{sum}(\text{sum}(b_i \tan d(S_4, 136)))}{(136 \times L \times M)} \end{cases} \quad (4.5)$$

The initial values are utilized as keys in the algorithm because this ensures that for each encrypted image, the four initial values will be unique.

Step 6: DNA Encoding Process for Sub-Blocks

DNA encoding is applied to the sub-blocks of $S1$, $S2$, and $S3$. The encoding process is defined by sequences X_i and Y_i .

Given that DNA can be encoded in 8 different ways, it is required to translate the values of sequences X_i and Y_i sequences must be transformed based on the equation (4.6):

$$\begin{cases} X_i = \text{mod}(\text{round}(X_i \times 10^4), 8) + 1 \\ Y_i = \text{mod}(\text{round}(Y_i \times 10^4), 8) + 1 \end{cases} \quad (4.6)$$

The translated sequence's value, X_i , is a random integer between 1 and 8. In S_1 , S_2 , and S_3 , the i -th sub-DNA block's coding method is X_i , whereas, in the chaotic matrix R , the i -th sub-DNA block's coding method is Y_i .

Step 7: Maximizing Encryption Efficiency

To enhance encryption efficiency, the same technique is applied to the corresponding blocks of $S1$, $S2$, $S3$, and R . The sequence Z_i is used to define the DNA operation (addition, subtraction, or XOR). It must be changed using equation (4.7) and converted into an integer spanning from 0 to 3.

$$Z_i = \text{mod}(\text{round}(Z_i \times 10^4), 4) \quad (4.7)$$

Specifically, it says if $Z_i = 0$, it is addition; if $Z_i = 1$, it is subtraction; if $Z_i = 2$, it is XOR.

Step 8: Enhancing Diffusion Effect with DNA Rules

The previous sub-block results of encryption and the current sub-block results are put through DNA rule once more to improve the diffusion effect; the algorithm used is selected by the sequence Z_i .

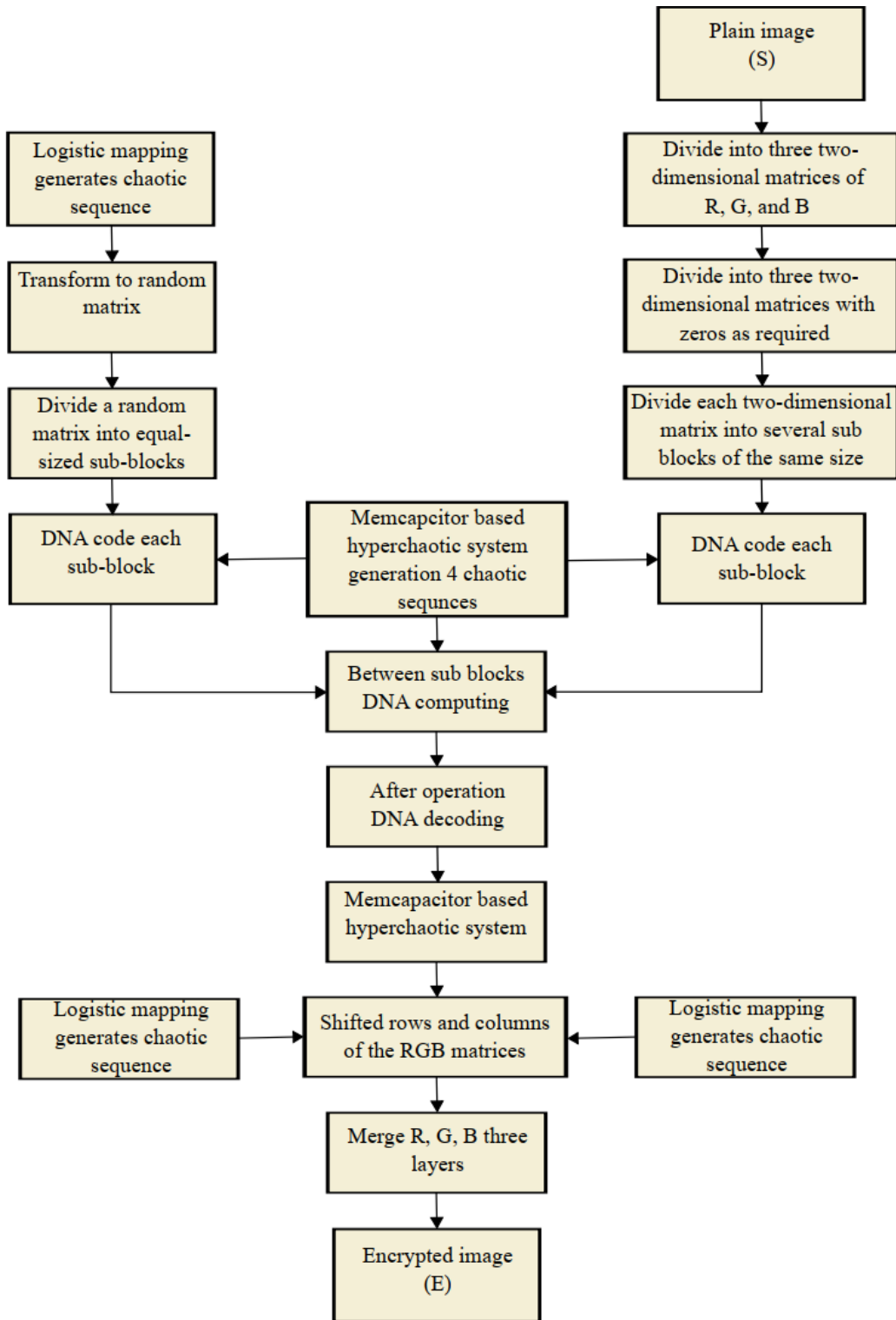


Figure 4.1: Flow of the Proposed Encryption Algorithm

Step 9: DNA Decoding Process for Matric Blocks

After DNA operation, the DNA decoding process is applied to the matric block. The DNA decoding rule for the sub-block following operation is determined by the sequence H_i . The

opposite of DNA encoding is DNA decoding. There are an additional eight decoding techniques, which translate the letters A, G, C, and T into precise numbers.

Step 10: Acquire two logistic chaotic sequences

Two chaotic sequences, k_y and k_x with lengths of M and N , respectively, are generated using a method identical to step (3) in the process of generating chaos. Among them, r is uniformly set to 3.9999, and the two initial values, x_{01} , and x_{02} , are obtained by using the equation (4.8).

$$\begin{cases} x_{01} = \frac{\text{sum}(S_1(:)) + \text{sum}(S_3(:))}{255 \times L \times M \times 2} \\ x_{02} = \frac{\text{sum}(S_2(:)) + \text{sum}(S_3(:))}{255 \times L \times M \times 2} \end{cases} \quad (4.8)$$

x_{01} and x_{02} values vary with the image and serve as keys for this encryption algorithm because x_{01} is the R channel and B channel gray average of the encrypted image and x_{02} is the G channel and the B channel gray average value.

Step 11: Descending Setup for Logistic Chaotic Sequences

According to equation (4.9), the sequences k_x and k_y are arranged in descending order to create position sequences U_x and U_y .

$$\begin{cases} [\sim, U_x] = \text{sort}(k_x, 'descend') \\ [\sim, U_y] = \text{sort}(k_y, 'descend') \end{cases} \quad (4.9)$$

After DNA decoding, row, and column movement is carried out on the matrix of the three channels using the U_x and U_y sequences values as well as their matching indices as the row and column exchange coordinates. This improves the scrambling effect and increases the resistance of cipher text to cropping attacks.

Step 12: Creating Cipher Text Image

Finally, the three-dimensional matrix is created by concatenating the shifted rows and columns of the RGB matrices, resulting in the encrypted (cipher text) image.

4.2.2 The Decryption Algorithm's Specific Steps

The following steps outline the detailed decryption algorithm for reconstructing the original image from the encrypted image using the reverse process of the encryption algorithm, which includes reversing chaotic sequences, DNA decoding, and a memcapacitor-based hyperchaotic system.

Step 1: Split the Encrypted image into RGB channels

The encrypted image E of size $L \times M$ is split into three two-dimensional matrices R , G , and B , denoted as E_1 , E_2 , and E_3 , using equation (4.10).

$$\begin{cases} E_1 = E(:, :, 1) \\ E_2 = E(:, :, 2) \\ E_3 = E(:, :, 3) \end{cases} \quad (4.10)$$

Step 2: Perform Column and Row Inverse Permutation

Apply the inverse column permutation and inverse row permutation on the three two-dimensional matrices R , G , and B using the chaotic sequences U_x , and U_y .

Step 3: Divide Each Channel into Sub-Blocks

Divide each of the two-dimensional matrices R , G , and B into equal-sized sub-blocks of size $t \times t$.

Step 4: Acquire Chaotic Sequences

Generate the same chaotic sequence used during encryption by iterating the logistic map with the same initial value and parameters:

$$x_0 = \frac{\text{sum}(E_1(:)) + \text{sum}(E_2(:))}{255 \times L \times M \times 2} \quad (4.11)$$

The initial value x_0 is computed similarly to the encryption process.

Step 5: DNA Decode Each Sub-Block

Perform DNA decoding on each sub-block of the matrices R , G , and B using the inverse of the DNA encoding technique applied during encryption.

Step 6: DNA Inverse Operation between Sub-Blocks

Apply the inverse DNA operation (addition, subtraction, XOR) between the corresponding sub-blocks of R , G , and B using the chaotic sequences Z_i generated by the memcapacitor-based hyperchaotic system.

Step 7: Memcapacitor-Based Hyperchaotic System Generation

Recompute the Hyperchaotic Memcapacitor system using the same initial values and parameters to produce the four sequences X_i , Y_i , Z_i , and W_i .

Step 8: DNA Decoding After Reverse Operation

Perform the final DNA decoding on the processed sub-blocks of the matrices R , G , and B after the reverse DNA operations.

Step 9: Remove Zero Elements Added During Encryption

Remove any zero elements that were added during the data transformation step in the encryption process.

Step 10: Merge R , G , and B Matrices

Combine the three two-dimensional matrices R , G , and B back into a single three-dimensional matrix to reconstruct the decrypted image.

Step 11: Obtain the Decrypted Image

The final output is the decrypted image which should be identical to the original image before encryption. The decryption algorithm's flowchart is shown in Figure 4.2.

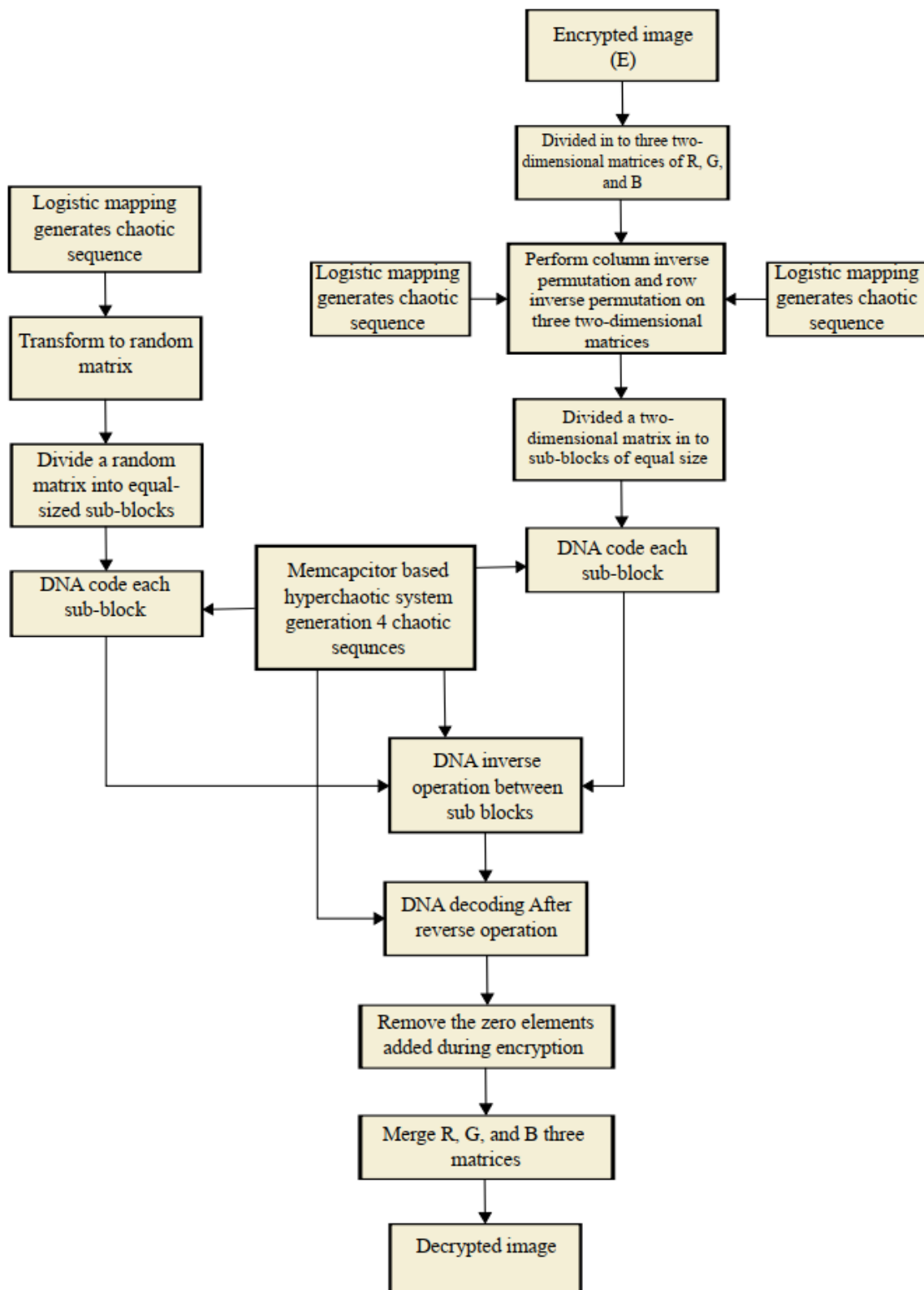


Figure 4.2: Flow of the Proposed Decryption Algorithm

4.3 Result and Discussion

All testing findings in this algorithm were simulated on a personal computer (PC) of Intel (R) Core(TM) i7 clocked at 3.20GHz with 8GHz RAM. The encryption performance test of the color images with different sizes such as Lena (256x256), Lena(512x512), Peppers (256x256), Peppers (512x512), Baboon (512x512), Airplane (1024x1024), Airbase (1024x1024), and Glaucoma Detection (3504x2336) are taken from Kaggle and USC-SIPI databases.

The performance test is carried out based on the simulation platform Matlab R2023a. The logistic map parameter (r) is set to the value 3.9999, and the memcapacitor base hyperchaotic system parameters are $a = 5.8001$, $b = 2.0013$, $c = 2.6010$, $d = 0.1002$, $e = -3.40104$, $f = 0.20017$, $m = 2.80051$, $n = 6.80031$. Figure 4.3 displays the considered images together with their original, encrypted, and decrypted versions. The proposed approach effectively hides the original image information. The encryption and decryption effects are quite good.

4.3.1 Histogram Analysis

When evaluating the quality of encrypted images, histograms serve as essential statistical tools. Within an image matrix, histograms depict the frequency distribution of various pixel levels. Utilizing the original image typically yields an unpredictable histogram plot characterized by varying spikes. For high-quality encrypted images that can withstand a statistical attack, the histogram should be consistent and free of spikes (Ravichandran et al., 2021; X. Zhang & Ye, 2021). The histogram should be consistent and free of spikes for high-quality encrypted images that can withstand a statistical attack. Figure 4.4 shows the *RGB* component histograms of the test images. These figures demonstrate that while a regular image's histogram displays irregular spikes, an encrypted image's histogram shows a more consistent pattern across various color components. This indicates the encryption technique effectively conceals the redundancy inherent in the original image (Alexan et al., 2022; Hao et al., 2021; M. Hu et al., 2023).

4.3.2 Correlation Analysis

A random selection of 5000 neighboring pixel pairs from the original image and the accompanying encrypted image is made to assess the image's resistance to attack. A strong encryption method can significantly minimize the similarity (correlation) between neighboring pixels in the original, unencrypted image, making it resistant to differential attacks such as the statistical attack of correlation analysis (X. Chen et al., 2023; Ibrahim et al., 2022). Attackers can use correlation analysis to collect cipher text information, but plaintext images possess greater adjacent pixel correlation. Moreover, test results for Lena's test images are shown in Figure 4.5.

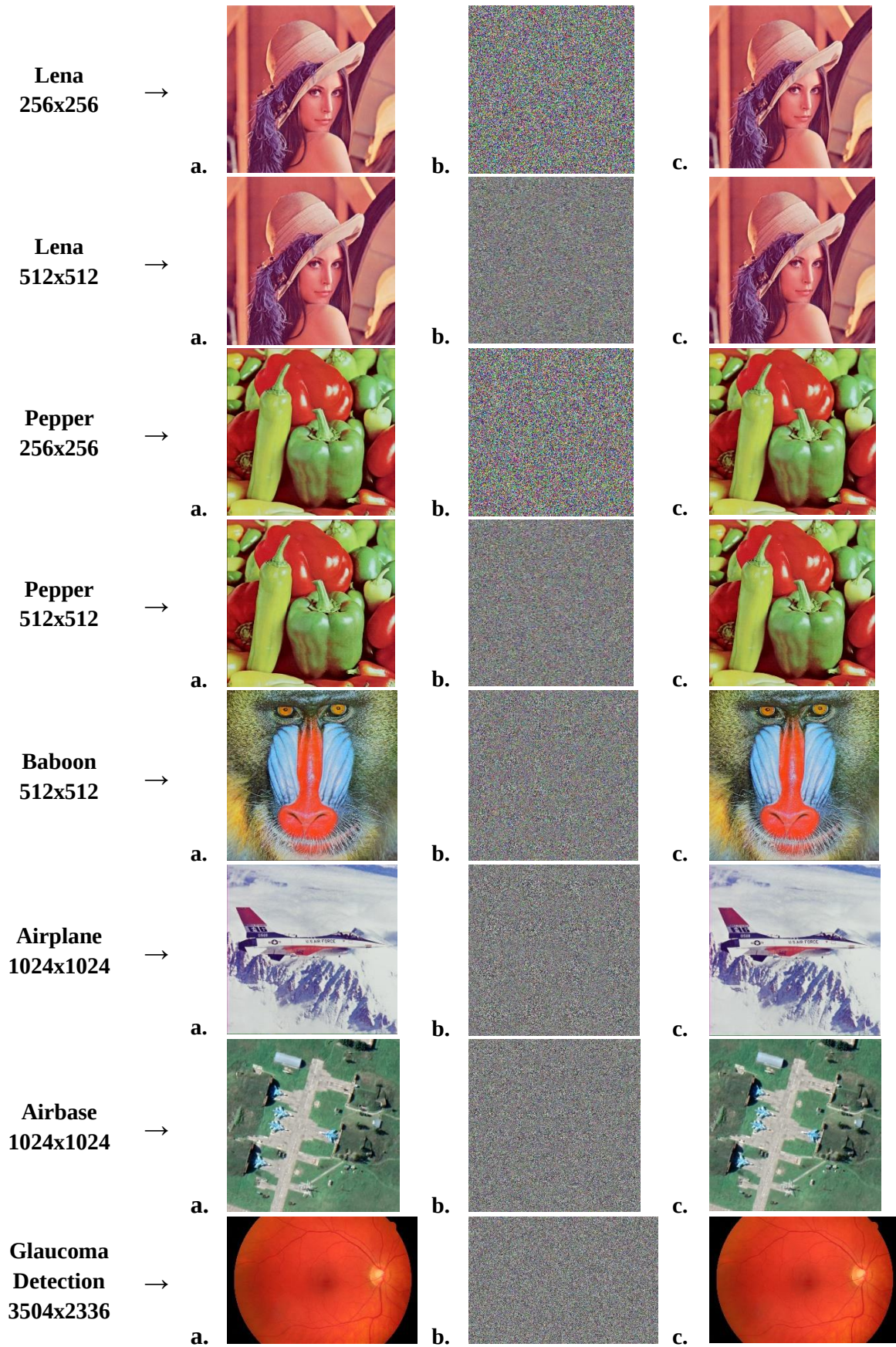


Figure 4.3: (a.) Plane Images (b.) Encrypted Images (c.) Decrypted Images (Lena, Peppers, Baboon, Airplane, Air Base, Eye for Glaucoma detection)

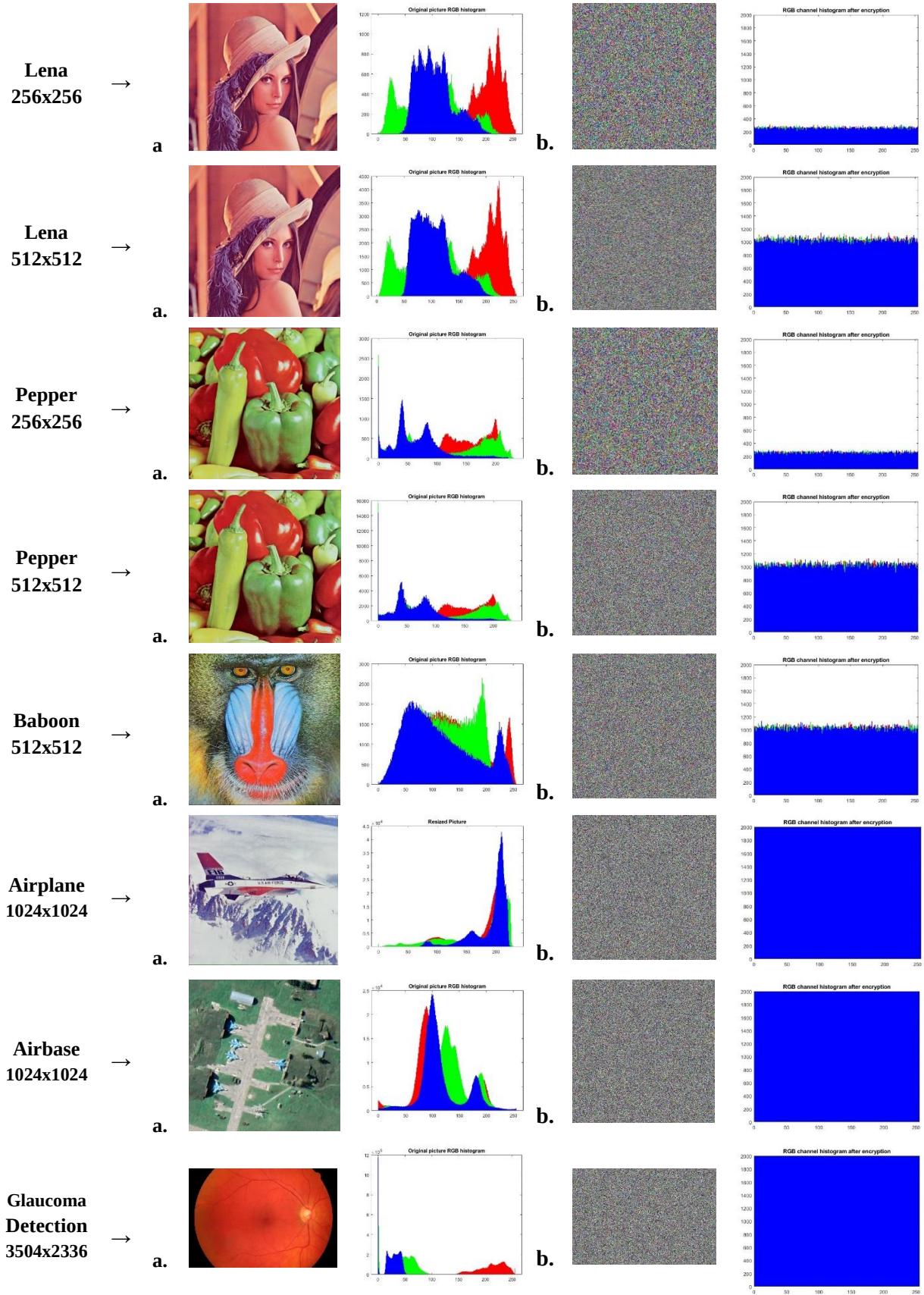
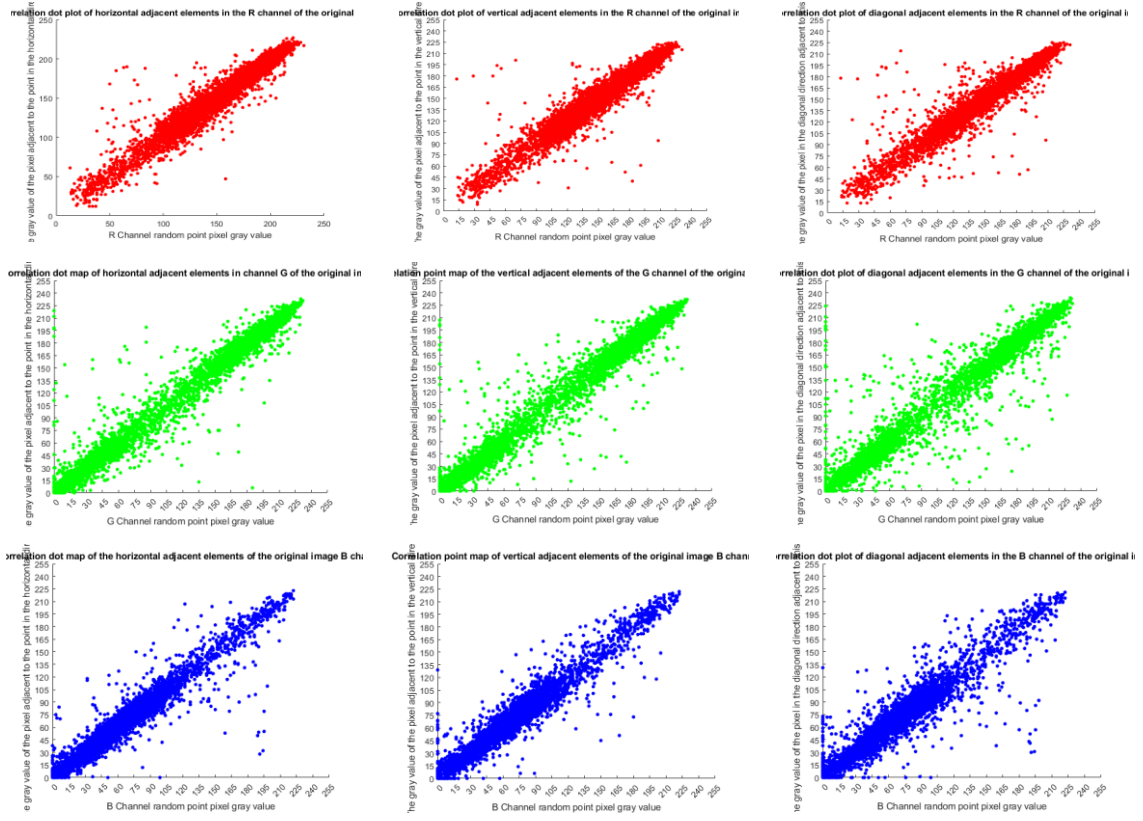


Figure 4.4: (a) Plane Images and Plane Images RGB Histogram (b) Encrypted Images and Encrypted Images RGB Histogram



(a) Horizontal (R, G, and B) (b) Vertical (R, G, and B) (c) Diagonal (R, G, and B)

Figure 4.5: Lena's Original Test Image's Horizontal, Vertical, and Diagonal Correlations

The correlation coefficient calculation formula r_{mn} is shown in equation (4.12).

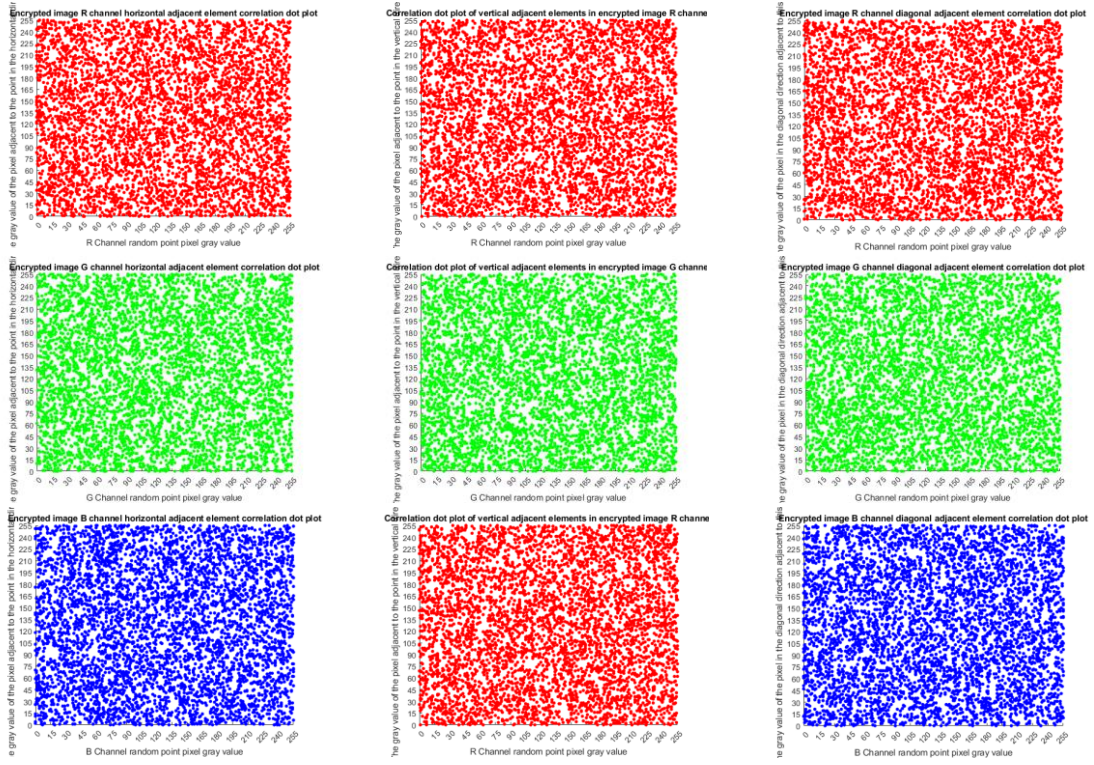
$$r_{xy} = \frac{cov(m, n)}{\sqrt{D(m)D(n)}} \quad (4.12)$$

$$cov(m, n) = \frac{1}{N} \sum_{i=1}^N (m_i - E(m))(n_i - E(n)) \quad (4.13)$$

$$E(m) = \frac{1}{N} \sum_{i=1}^N m_i \quad (4.14)$$

$$D(m) = \frac{1}{N} \sum_{i=1}^N (m_i - E(m))^2 \quad (4.15)$$

Findings of Table 4.1 and Figure 4.6 clearly show that in the cipher text image, there is no relationship between neighboring pixels, however, in the unencrypted picture, the correlation is greater and contains more information. As a result, our picture encryption system algorithm is secure, and the memcapacitive hyperchaotic system can be used for image encryption. Table 4.2 shows comparisons of correlation result values with several comparable encryption techniques, indicating how resistant to attacks.



(a) Horizontal (R, G, and B) (b) Vertical (R, G, and B) (c) Diagonal (R, G, and B)
Figure 4.6: Lena's Encrypted Test Image's Horizontal, Vertical, and Diagonal Correlations

Table 4.1: Original Test Images and Encrypted Test Images Correlation Coefficients.

Image with Size	Correlation Directions	Pixel correlation coefficients					
		Original images			Encrypted images		
		Red	Green	Blue	Red	Green	Blue
Lena 256x256	Horizontal (H)	0.94116	0.93768	0.89975	-0.0042867	-0.023386	-0.01857
	Vertical (V)	0.96989	0.96902	0.94145	0.0038526	0.011966	-0.011937
	Diagonal (D)	0.91205	0.90961	0.86143	0.016806	0.015511	0.016922
Lena 512x512	Horizontal (H)	0.97936	0.96855	0.93224	-0.005732	-0.0090023	0.0071144
	Vertical (V)	0.98930	0.98176	0.95735	0.0031900	-0.0089338	0.026196
	Diagonal (D)	0.96947	0.95408	0.91535	-0.003585	-0.0021488	-0.015438
Pepper 256x256	Horizontal (H)	0.95992	0.96459	0.94855	-0.021446	-0.010107	-0.011459
	Vertical (V)	0.96219	0.97411	0.96173	0.005143	0.0046875	-0.006566
	Diagonal (D)	0.9333	0.94183	0.92031	0.012259	-0.010107	-0.011459
Pepper 512x512	Horizontal (H)	0.96763	0.98116	0.96704	0.0055525	0.0125	0.0057408
	Vertical (V)	0.96621	0.97931	0.96673	0.0015536	-0.011885	0.0046432
	Diagonal (D)	0.95845	0.96757	0.95029	-0.0068352	-0.014089	-0.0040064
Baboon 512x512	Horizontal (H)	0.91956	0.86004	0.9042	-0.013968	-0.0084904	-0.0082242
	Vertical (V)	0.86244	0.75304	0.88162	0.004951	0.0021037	0.0015685
	Diagonal (D)	0.85148	0.72368	0.83962	-0.013968	-0.0099518	0.0014991
Airplane 1024x1024	Horizontal (H)	0.98394	0.99078	0.99191	0.0057781	-0.0039261	-0.0055248
	Vertical (V)	0.98394	0.99187	0.98639	0.01337	0.014752	0.0096844
	Diagonal (D)	0.98394	0.98375	0.97998	0.015248	-0.0066388	-0.027975
Airbase 1024x1024	Horizontal (H)	0.99634	0.99462	0.99431	-0.0033467	0.0029241	-0.015492
	Vertical (V)	0.99599	0.99457	0.98817	0.021041	0.00094788	0.0024306
	Diagonal (D)	0.99226	0.98868	0.99431	0.0038163	0.027586	-0.01735
Glaucoma Detection 3504x2336	Horizontal (H)	0.99987	0.99911	0.99763	-0.0036791	0.0043195	-0.0077271
	Vertical (V)	0.99975	0.99903	0.99669	0.039046	0.0054756	0.011023
	Diagonal (D)	0.99987	0.99855	0.99541	-0.0078015	-0.0081225	0.011524

Table 4.2: Correlation Results between Encrypted Images with Other Techniques

Channels Directions	Red			Green			Blue			Average
	H	V	D	H	V	D	H	V	D	
Original Lena image	0.94116	0.96989	0.91205	0.93768	0.96902	0.90961	0.89975	0.94145	0.86143	0.926893
Ours	-0.00428	0.00385	0.016806	-0.02338	0.01196	0.01551	-0.0186	-0.0119	0.01692	0.000765
Ref. (Ahuja & Doriya, 2023)	-0.0038	-0.0226	-0.0015	-0.0038	-0.0226	-0.0015	-0.0038	-0.0226	-0.0015	-0.00930
Ref. (X. Zhang & Ye, 2021)	-0.0085	-0.0008	0.0033	-0.0255	0.0065	-0.0035	0.0087	-0.0115	0.0030	0.00538
Ref. (V. Kumar & Girdhar, 2021)	-0.0015	0.0018	0.0019	-0.0008	0.0016	0.0069	0.0024	0.0039	0.0027	0.00303
Ref. (Ibrahim et al., 2022)	0.002	0.0054	0.0054	0.0037	0.0037	0.0022	-0.0042	-0.0049	0.002	0.00349
Ref. (M. A. Lone & Qureshi, 2023)	0.0031	0.0031	0.0031	0.0036	0.0036	0.0036	0.0008	0.0008	0.0008	0.00250
Ref. (Alexan et al., 2022)	0.002287	-0.0016	-0.00132	0.002287	-0.0016	-0.0013	0.00229	-0.0016	-0.00132	0.002287

4.3.3 Information Entropy Analysis

The level of information confusion in a system is measured by information entropy. As a result, information entropy $H(x)$ inside a picture $S(x)$ shows the level of image pixel confusion. In equation (4.16), its precise mathematical definition is displayed:

$$H(x) = - \sum_{i=0}^{2^N-1} S(x_i) \log_2 S(x_i) \quad (4.16)$$

Among them, $S(x_i)$ denotes the ratio of the image's overall gray level. When an image's gray level is M , its maximal entropy is $H_{max} = \log_2 M$ (bit/symbol). Therefore, when $M = 256 = 2^8$, $H_{max} = 8$.

Table 4.3 contrasts the information entropy data between the initial pictures and the ciphertext images, which was computed using an equation (4.16). The cipher text image information entropy in the R channel, G channel, and B channel is extremely close to the maximum

information entropy of eight in theory, which indicates that the cipher text image's level of chaotic confusion is almost at the predicted theoretical limit value, as experimentally indicated by Table 4.3. Therefore, this encryption method can fend off attacks based on the image information entropy.

Table 4.3: Information Entropy Comparison Among Original and Cyphered Images

Image with Size		Red	Green	Blue
Lena 256x256	Original images	7.2605	7.584	6.9661
	Encrypted images	7.9973	7.9972	7.9973
Lena 512x512	Original images	7.2531	7.594	6.9684
	Encrypted images	7.9993	7.9993	7.9993
Pepper 256x256	Original images	7.3011	7.5596	7.0947
	Encrypted images	7.9975	7.9969	7.997
Pepper 512x512	Original images	7.3388	7.4963	7.0583
	Encrypted images	7.9992	7.9993	7.9993
Baboon 512x512	Original images	7.7067	7.4744	7.7522
	Encrypted images	7.9992	7.9994	7.9993
Airplane 1024x1024	Original images	6.7164	6.8005	6.2095
	Encrypted images	7.9998	7.9998	7.9998
Airbase 1024x1024	Original images	7.0428	7.0971	6.9297
	Encrypted images	7.9998	7.9998	7.9998
Glaucoma 3504x2336	Original images	6.3703	5.8212	5.1795
	Encrypted images	8	8	8

The Lena test image's entropy comparison findings are displayed in Table 4.4.

Table 4.4: Entropy Results Comparison of Lena Test Image with Other Techniques

Lena encrypted image	Entropy value		
	Red	Green	Blue
Maximum	8.0000	8.0000	8.0000
Proposed	7.9993	7.9993	7.9993
Ref. (Ahuja & Doriya, 2023)	7.9971	7.9971	7.9971
Ref. (Y. Wang, Wu, et al., 2020)	7.9974	7.9970	7.9969
Ref. (V. Kumar & Girdhar, 2021)	7.9980	7.9980	7.9970
Ref. (D. Lin et al., 2022)	7.9971	7.9971	7.9969
Ref. (Ibrahim et al., 2022)	7.9992	7.9952	7.9958
Ref. (M. A. Lone & Qureshi, 2023)	7.9974	7.9974	7.9974
Ref. (Alexan et al., 2022)	7.9972	7.9973	7.9966
Ref. (H. Wang et al., 2023)	7.998187	7.998187	7.998187

4.3.4 Keyspace Analysis

An appropriate cryptographic algorithm's key space has to be large enough to resist brute-force attacks. Image encryption security will increase in proportion to the key space. The hidden keys in our proposed encryption system are made up of a total of 10 variables such as Logistic map iteration value (r), Logistic map initial value (x_0), memcapacitor hyperchaotic system initial values (X_0 , Y_0 , Z_0 , and H_0), Zero padded values of rows and columns (M_I and N_I), Average gray value of G channel (xx_0) and Average gray level value of B channel (xx_I). From (Alexan, Elkandoz, et al., 2023) the greatest machine precision that we take into account in this work is 10^{-16} , so our suggested image encryption method has a key space that is bigger than $10^{16 \times 10} = 10^{160} \approx 2^{531}$. Which is significantly more than the necessary 2^{100} (C. Chen et al., 2020; Z. Liang et al., 2023b; Xie, 2021). There is enough key space in the encryption system. The proposed encryption system algorithm can therefore resist brute force attacks.

Assume that there are three sub-keys in the generated key space to evaluate the benefit of using a memcapacitive hyperchaotic system to expand the key space. Each sub-key key is 531 bits long, so an adversary would need to perform 3×2^{531} processes to identify the key. To determine the right key using a brute force attack, the attacker will need to execute 1000×10^6 instructions per second on the computer, which will take about 6.68717×10^{143} years (Elsaid & Alotaibi, 2023).

$$\frac{3 \times 2^{531}}{1000 \times 10^6 \times 60 \times 60 \times 24 \times 365} > 6.68717 \times 10^{143} \text{ years} \quad (4.17)$$

Table 4.5: Comparative of Keyspace

Method	Keyspace value
Minimum	2^{100}
Proposed	2^{531}
Ref. (Ahuja & Doriya, 2023)	2^{398}
Ref. (Y. Wang, Wu, et al., 2020)	2^{187}
Ref. (V. Kumar & Girdhar, 2021)	2^{349}
Ref. (X. Zhang & Ye, 2021)	2^{498}
Ref. (Ibrahim et al., 2022)	2^{127}
Ref. (Alexan et al., 2022)	2^{425}

4.3.5 Key Sensitivity Analysis

Any change in a strong cryptosystem's encryption/decryption passwords will significantly impact. This implies that if the encryption key is slightly altered, the projected picture will also be altered somewhat. A different decoded image results from even the slightest alteration of the decoding key. To assess the proposed cryptosystem sensitivity of the key, the original plain

picture in Figure 4.7(a), which is the test image, is encrypted using secret keys, and then the encrypted image is displayed in Figure 4.7(b). The picture that has been correctly decrypted is shown in Figure 4.7(c). For instance, the first key is the parameter $r = 3.9999$. For example, the logistic map parameter value was modified into $r = 3.9998$. After making a single change, Figure 4.7(d) displays the decrypted picture using a single slightly different key.

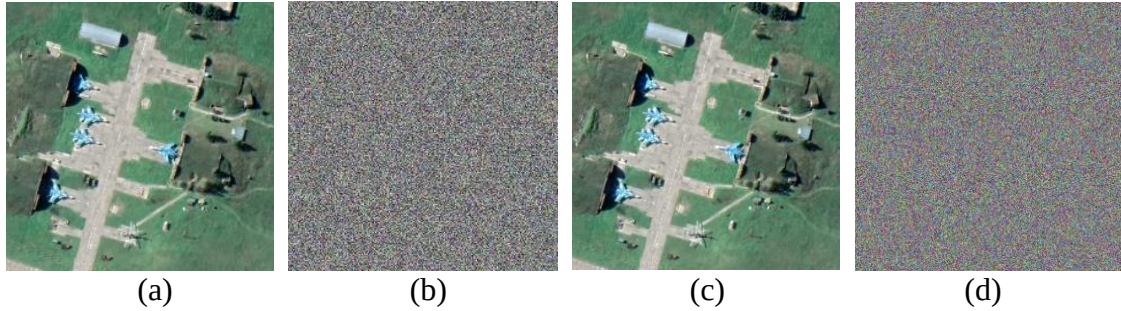


Figure 4.7: Encryption-Decryption Results with and without Slight Key Changes;
(a) Original image, (b) Ciphred image, (c) Recovered image, (d) Recovered with a slightly different key

4.3.6 Mean Squared Error

For all pixel sizes of $L \times M$ in the picture, the difference between an encrypted $E(l,m)$ image and the original test image $S(l,m)$ is squared. The total squared error values of all image pixels are then divided by all L by M pixels, as illustrated in equation (4.18). A more robust encryption scheme against statistical attacks has a higher Mean Squared Error (MSE) value.

$$MSE = \frac{1}{L \times M} \sum_{l=1}^L \sum_{m=1}^M [S(l,m) - E(l,m)]^2 \quad (4.18)$$

One of the ways to evaluate the efficiency of an encryption system is to look at The Peak Signal-to-Noise Ratio ($PSNR$), which is the ratio of the image's greatest pixel values over the mean square error (MSE). It can be expressed mathematically as:

$$PSNR = 10 \log \left[\frac{P_{max}^2}{MSE} \right] \quad (4.19)$$

where P_{max} is 255 which is the highest possible pixel value.

The Peak Signal-to-Noise Ratio ($PSNR$) value in theory should be as low as is practical because it has an inverse relationship with the MSE . The $PSNR$ values from the original and reconstructed images were found to be identical. However, the $PSNR$ values between the original and cipher images were found to be considerably low, indicating that the encrypted picture resisted attacks very well. The $SSIM$ is an additional metric employed to assess the quality of images. It's worth mentioning that the encryption method used had an infinite $PSNR$ and a value of one for Structural Similarity Index Measure ($SSIM$), indicating that the original

image was identical to the decrypted image (Ahuja & Doriya, 2023). The values of *PSNR* and *SSIM* for the test images that have been fully decrypted are displayed in Table 4.6.

Table 4.6: PSNR and SSIM Values for Original and Reconstructed Images

Image with Size	<i>PSNR</i>	Maximum	<i>SSIM</i>
Lena 256x256	∞	1	1
Lena 512x512	∞	1	1
Pepper 256x256	∞	1	1
Pepper 512x512	∞	1	1
Baboon 512x512	∞	1	1
Airplane 1024x1024	∞	1	1
Airbase 1024x1024	∞	1	1
Glaucoma Detection 3504x2336	∞	1	1

Table 4.7 shows calculated *MSE*, *PSNR*, and *MAE* values of original and encrypted images of *RGB* channels. Table 4.8 shows values of *MSE*, *PSNR*, and *MAE* for various encrypted images and compares them with results from other approaches in the literature.

4.3.7 Mean Absolute Error

To ensure that the cryptographic system is reliable, a high *MAE* is required to distinguish between encrypted and original test pictures. It is mathematically expressed as:

$$MAE = \frac{1}{L \times M} \sum_{l=1}^L \sum_{m=1}^M [S(l, m) - E(l, m)] \quad (4.20)$$

The original sample plain image pixels are referred to as $S(l, m)$, while the pixels of the cyphered picture in m rows and n columns are referred to as $E(l, m)$. M and N represent the image's dimensions. Table 4.7 demonstrates that the *MSE*, *PSNR*, and *MAE* calculated for the proposed system are either better than or on par with those of comparable schemes found in previous studies. The comparative analysis of *MSE*, *PSNR*, and *MAE* values for the Lina test image of *RGB* channels is listed in Table 4.8.

Table 4.7: MSE, PSNR, and MAE Values of Original and Encrypted Images

Image with Size	Channel	MSE (dB)	PSNR (dB)	MAE
Lena 256x256	Red	10729.45	7.83	84.58
	Green	9002.85	8.59	77.83
	Blue	7087.96	9.63	70.33
	Average	8940.09	8.683333	77.58
Lena 512x512	Red	10681.16	7.84	84.40
	Green	9025.59	8.58	77.84
	Blue	7098.02	9.62	70.36
	Average	8934.92	8.68	77.53333
Pepper 256x256	Red	7968.67	9.12	73.81
	Green	11130.13	7.67	86.22
	Blue	11010.44	7.71	85.70
	Average	10036.4	8.166667	81.91
Pepper 512x512	Red	7970.57	9.12	73.76
	Green	11242.16	7.62	86.53
	Blue	11180.86	7.65	86.40
	Average	10131.2	8.13	82.23
Baboon 512x512	Red	8643.83	8.76	76.45
	Green	7775.90	9.22	73.13
	Blue	9474.80	8.37	79.69
	Average	8631.51	8.783333	76.42333
Airplane 1024x1024	Red	9936.63	8.16	81.48
	Green	10634.57	7.86	84.18
	Blue	10432.43	7.95	83.43
	Average	10334.5	7.99	83.03
Airbase 1024x1024	Red	7948.23	9.13	73.71
	Green	6989.61	9.69	69.97
	Blue	7283.33	9.51	71.14
	Average	7407.06	9.443333	71.60667
Glaucoma Detection 3504x2336	Red	14590.04	6.49	99.66
	Green	12128.06	7.29	90.04
	Blue	15872.87	6.12	104.67
	Average	14197	6.633333	98.12333

Table 4.8: Comparative MSE, PSNR, and MAE Values of Lina Image RGB Channels

Different Methods	MSE				PSNR (dB)				MAE
	Red	Green	Blue	Average	Red	Green	Blue	Average	
Proposed	10681.16	9025.59	7098.02	8934.92	7.84	8.58	9.62	8.68	77.533
Ref. (Ahuja & Doriya, 2023)	-	-	-	-	18.66	18.66	18.66	18.66	-
Ref. (M. A. Lone & Qureshi, 2023)	9004.40	9004.40	9004.40	9004.40	8.6454	8.6454	8.6454	8.6454	-
Ref. (Alexan et al., 2022)	10,663.3	8982.02	7021.33	8888.88	7.8518	8.5970	9.6666	8.7051	77.375
Ref. (Shakir et al., 2022)	-	-	-	8914.025	-	-	-	3.891629	-
Ref. (H. Wang et al., 2023)	-	-	-	11689.814	-	-	-	9.160910	-

4.3.8 Differential Attack Analysis

Image encryption often faces a common type of attack called differential attack, where attackers change some pixels in the source image before encrypting it. This alteration can reveal a connection between different cipher images, making it necessary for encryption techniques to create completely new encrypted images with any small changes to the original picture. The *PSNR* and *SSIM* criteria are used to evaluate an encryption scheme's resistance against differential attacks. The encrypted picture would be greatly affected by even the smallest change made to the plain text image (Alexan et al., 2022). Whereas unified averaged changed intensity (*UACI*) determines the average pixel intensity changes, the numbers of changing pixel rates (*NPCR*) quantifies the proportion of pixel changes in two cipher pictures.

$$NPCR = \frac{1}{L \times M} \sum_{l=1}^L \sum_{m=1}^M D(l, m) \times 100\% \quad (4.21)$$

$$D(l, m) = \begin{cases} 1 & \text{if } S_1(l, m) \neq S_2(l, m) \\ 0 & \text{else} \end{cases} \quad (4.22)$$

$$UACI = \frac{1}{L \times M} \sum_{l=1}^L \sum_{m=1}^M \frac{|S_1(l, m) - S_2(l, m)|}{255} \times 100\% \quad (4.23)$$

whereby $S_1(l, m)$, and $S_2(l, m)$ are the pixel values for encrypted pictures both before and after the basic image's first pixel is altered. The *NPCR* and *UACI* values differ based on the characteristics of the source image. When using effective encryption techniques, the result of *NPCR* typically should be above 99.00% and the result of *UACI* is generally 25% (Z. Liu et al., 2019). Generally, for both plain and cipher images, an effective encryption method ought to have low *UACI* results and maximum *NPCR* results. Original and encrypted pictures were evaluated for *UACI* and *NPCR*, Table 4.9 displayed the results obtained.

Table 4.9: The *UACI* and *NPCR* Results of Test Images

Image with Size	<i>UACI</i> (%)				<i>NPCR</i> (%)			
	Red	Green	Blue	Average	Red	Green	Blue	Average
Lena (256x256)	33.1703	30.5218	27.5819	30.4247	99.6124	99.5712	99.5743	99.586
Lena (512x512)	33.0988	30.5243	27.5936	30.4056	99.5945	99.6101	99.6174	99.607
Pepper (256x256)	28.9456	33.8122	33.6094	32.1224	99.6323	99.6628	99.6292	99.641
Pepper (512x512)	28.9273	33.9345	33.8829	32.2482	99.6254	99.5964	99.5872	99.603
Baboon (512x512)	29.9794	28.6792	31.2504	29.9697	99.6075	99.6025	99.6193	99.6098
Airplane (1024x1024)	31.9531	33.0124	32.717	32.5608	99.6119	99.6073	99.6118	99.6103
Airbase (1024x1024)	28.904	27.4384	27.8987	28.0803	99.6202	99.6107	99.6218	99.6175
Glaucoma Detection (3504x2336)	39.0808	35.3117	41.0479	38.4801	99.6095	99.6082	99.6055	99.6077

The results in Table 4.9 demonstrate the successful performance of this proposed technique in preventing differential attacks on test images. The *NPCR* typical results in our method are between 99.634% and 99.587% which are very good results also the typical values of *UACI* values in this method are between 33.935% and 27.474% which are acceptable values. The *UACI* and *NPCR* values are nearly at optimal levels, showing that the technique is sensitive to the plaintext picture. This sensitivity leads to high-level protection against differential attack analysis.

Table 4.10: The *UACI* and *NPCR* Results of Test Images

Algorithm	<i>UACI</i>				<i>NPCR</i>			
	Red	Green	Blue	Average	Red	Green	Blue	Average
Proposed	33.0988	30.5243	27.5936	30.4056	99.5945	99.6101	99.6174	99.607
Ref. (Y. Wang, Wu, et al., 2020)	33.520	33.422	33.481	33.474	99.6077	99.608	99.605	99.6069
Ref. (V. Kumar & Girdhar, 2021)	33.477	33.552	33.468	33.499	99.615	99.639	99.641	99.6317
Ref. (Q. Zhang & Han, 2021)	33.459	33.476	33.433	33.456	99.6096	99.610	99.609	99.6095
Ref. (X. Zhang & Ye, 2021)	33.43	33.49	33.53	33.483	99.60	99.60	99.62	99.6067
Ref. (Ibrahim et al., 2022)	33.13	33.13	33.13	33.130	99.540	99.540	99.540	99.54
Ref. (Alexan et al., 2022)	33.416	30.390	33.242	32.349	99.610	99.610	99.637	99.619
Ref. (Shakir et al., 2022)	33.3656	33.3656	33.3656	33.3656	99.607	99.607	99.607	99.607

Table 4.10 shows the results of the proposed approach and compares them to data from other sources. The proposed method demonstrates better performance to subtle variations in the original image and is more resilient to attacks.

4.3.9 Running Speed Analysis

The efficiency of the encryption and decryption technique depends on variable processing speeds, which are influenced by hardware, software, and programming. Despite the time overhead associated with DNA coding, testing our algorithm on the designated platform indicated excellent efficiency. In the current environment, achieving an average encryption and decryption time of 1.4636871 seconds for a 512x512 image guarantees high-security standards. Increasing support for both hardware and software can help to further maximize operating speeds. Table 4.11 displays the ten sequential encryption decryption times, from which we can calculate the average duration.

Table 4.11: The Sample Times for Encryption and Decryption

Ten samples (in seconds)									
1	2	3	4	5	6	7	8	9	10
2.007322	1.613485	1.323120	1.411952	1.607091	1.391182	1.393017	1.229910	1.360453	1.299339
Average time = 1.4636871									

4.4 Randomness Evaluation Using the NIST Statistical Test Suite

The NIST SP800-22 Statistical Test Suite (STS) evaluates the randomness of an encrypted image's bit stream. It assesses properties such as uniformity, independence, and entropy to determine genuine randomness.

Table 4.12: NIST STS Results for the Proposed Encrypted Image Bit Stream

No	Test Name	P-Value	Proportion	Accepted/Rejected
1	Frequency (Monobit) Test	0.578211	100%	Accepted
2	Block Frequency Test	0.027181	100%	Accepted
3	Cumulative Sums (Cusum) Test	Forward Reverse 0.971815 0.519702	100%	Accepted
4	Runs Test	0.699726	100%	Accepted
5	Longest Run of Ones in a Block Test	0.607036	100%	Accepted
6	Rank Test	0.431595	100%	Accepted
7	Discrete Fourier Transform (Spectral) Test (FFT)	0.373394	100%	Accepted
8	Non-overlapping Template Matching Test Average	0.515800	100%	Accepted
9	Overlapping Template Matching Test	0.787798	100%	Accepted
10	Universal Statistical Test (Maurer's)	0.424240	100%	Accepted
11	Approximate Entropy Test	0.473661	100%	Accepted
12	Random Excursions Test Average	0.571400	100%	Accepted
13	Random Excursions Variant Test Average	0.651700	100%	Accepted
14	Serial Test	p_value1 p_value2 0.245093 0.303213	100%	Accepted
15	Linear Complexity Test	0.650030	100%	Accepted

The analysis of a single bitstream in ASCII mode displayed each bit as either '0' or '1'. The results were straightforward: the test either passed (100%) or failed (0%). As shown in Table 4.12, all tests had a p-value threshold of ≥ 0.01 , indicating that no detectable patterns were found in the binary representation of the encrypted image. This finding supports the cryptographic security of the encryption and aligns with the research of (Gedda et al., 1977) and (X. Wu et al., 2025). Overall, the results demonstrate statistical randomness in the bitstream, further reinforcing its cryptographic strength.

4.5 Comparative Evaluation of Security and Performance Metrics

The suggested encryption method, integrating a memcapacitor-based hyperchaotic system and DNA coding, demonstrates superior security performance compared to existing methods. It achieves NPCR up to 99.634% and UACI up to 33.935%, surpassing the results of Ahuja & Doriya (2023) and Wang et al. (2020). The information entropy reaches 7.9998, surpassing earlier values below 7.998, while correlation coefficients are almost zero, outperforming Ibrahim et al. (2022) and Alexan et al. (2022). Notably, the key space is 2^{531} , which surpasses the standard threshold of 2^{100} , offering significantly stronger resistance against brute-force attacks. Additionally, PSNR = ∞ and SSIM = 1 confirm excellent image recovery. Table 4.13 presents the key performance metrics achieved by the proposed encryption scheme, clearly demonstrating its high security, efficiency, and robustness.

Table 4.13: Performance Summary of the Proposed Image Encryption Scheme

Metric	Result
NPCR	99.587% – 99.634%
UACI	27.474% – 33.935%
Keyspace Size	2^{531}
Entropy (Max: 8)	Up to 7.9998
Encryption + Decryption Time	1.4637 seconds

4.6 Summary

This chapter discussed a novel and highly secure approach to image cryptography by integrating DNA code, a Memcapacitor-based hyperchaotic system, and a logistic map. The proposed method effectively conceals test image information, producing cipher images distinct from the originals. Extensive experiments demonstrate its sensitivity to encryption key changes, offering a wide range of potential key combinations and rendering it resilient against brute force and other attacks. Its security, efficiency, speed, and resistance to various attacks make it suitable for image Transmission in Wireless Sensor Networks. The research highlights the significance of this approach in enhancing data security for image transmission in open communication channels, addressing the limitations of traditional encryption techniques. The algorithm utilizes hyperchaotic sequences to enhance security as the image size increases due to their greater complexity and unpredictability. However, potential limitations include the need for real-world validation and the computational complexity of implementation. Future research should aim to optimize efficiency and explore additional applications beyond image encryption. Overall, this work makes a significant contribution to the advancement of image encryption algorithms and sets the stage for future developments.

CHAPTER 5

ENHANCING IMAGE ENCRYPTION THROUGH PARAMETERS OPTIMIZATION OF MEMRISTIVE HYPERCHAOTIC SYSTEM WITH CPSO ALGORITHM AND LSAIM

5.1 Introduction

Particle Swarm Optimization (PSO) is a simple and effective algorithm for optimization. However, it often converges too early and depends heavily on having the right parameter settings for good results, especially in complex or high-dimensional search spaces. Therefore, this algorithm is not appropriate for the parameter optimization of hyperchaotic systems and may not provide sufficient robustness and security of color images. This chapter proposed a new color image encryption based on optimizing a four-dimensional Memristor-based hyperchaotic system, in which the fourteen parameters of the hyperchaotic system are optimized by the Chaotic Particle Swarm Optimization (CPSO) method, in combination with DNA code and a new logistic sine adjusted integrated map (LSAIM).

The process of optimization using the CPSO method is discussed in Chapter 3. The simulation results show that the algorithm offers a large key space and strong protection against brute-force attacks, ensuring robust security with nearly ideal entropy values. It is resilient to statistical and differential attacks, and its high sensitivity to key changes enhances security. Thus, the algorithm is suitable for secure image transmission and storage.

5.2 Proposed Encryption Algorithm Using an Optimized Hyperchaotic System

This novel image encryption algorithm was built using an optimized memristor-based hyperchaotic system, a new logistic sine adjusted integrated map, and DNA code in a new way of multiple processing methods.

5.2.1 The Algorithm for Encryption

The suggested memristor-based cryptographic method is put into practice in the following steps. Figure 5.1 shows the flow of the encryption algorithm.

Step 1: A color image with $W \times L$ dimensions, such as channels. Three two-dimensional RGB channels are created after loading $W = 512$ and $L = 512$.

Step 2: To increase the algorithm's universality, the values of the data are placed into three two dimensional for every dimension. Consider it zero. If needed, the image is padded with

zeros to attain these dimensions, confirming uniform block processing in subsequent encryption phases.

Step 3: Setting the value of r_1 , r_2 and, initial value y_0 iterate the LSAIM continuously using predefined equation (3.73), and set the resultant sequence's length to $W \times L$. This scheme uses three keys, with r_1 and r_2 , initial set to 6, and y_0 value based on the input image's pixels, creating a unique, secure chaotic sequence.

Step 4: Convert the sequence to a $W \times L$ matrix with two dimensions, and then to carry out a DNA operation, change the matrix's value to a range between 0 and 255.

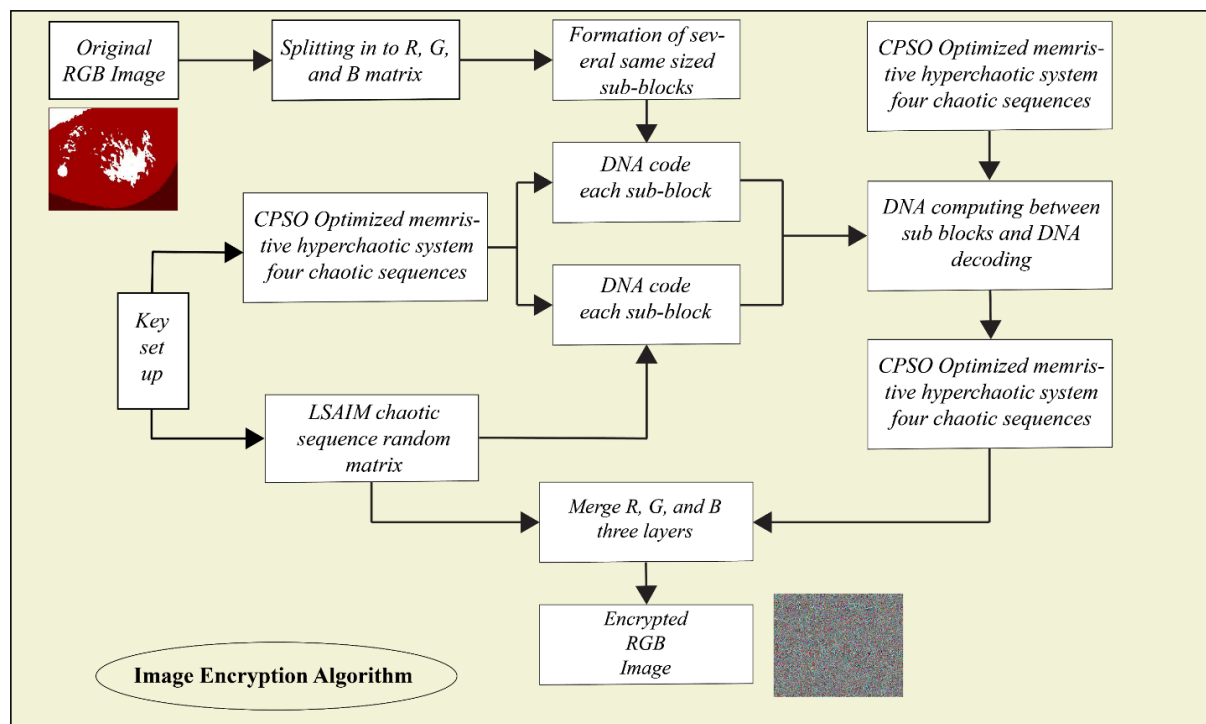


Figure 5.1: Diagram Illustrating the Suggested RGB Image Encryption

Step 5: Compute the Hyperchaotic memristor system using Matlab's built-in Runge-Kutta function ODE45 to generate the sequences of x , y , z , and w . Each of the channel data values can be represented by an 8-bit binary number because they all fall within the range of 0 to 255. And Determine the Hyper chaotic memristor system's initial condition values, $x(0)$, $y(0)$, $z(0)$, and $w(0)$.

Step 6: Using the same DNA encoding technique, which is defined by X_i and the DNA coding process of each sub-block of a channel matrix is determined by Y_i , given that DNA can be encoded in 8 different ways, it is required to translate the values of sequences X_i and Y_i sequences must be transformed into a random integer between 1 and 8.

Step 7: To maximize encryption efficiency, the same technique is used between the appropriate blocks of three channels. Then sequence is created by the Hyperchaotic memristor system. Because this technique employs four different DNA algorithms.

Step 8: The encryption results of preceding and subsequent sub blocks are reprocessed with a DNA operation once more to enhance the diffusion effect.

Step 9: After DNA operation, the matrix block is subjected to DNA decoding. There are an additional eight decoding techniques, which translate the letters A, G, C, and T into precise numbers.

Step 10: Acquire two LSAIM chaotic sequences. Two chaotic sequences, k_x and k_y with lengths of L and W , respectively, are generated in the process of generating chaos.

Step 11: The sequences k_x and k_y are set up in descending sequence, and the position sequences before the ordering of each element in the sequence are acquired. After DNA decoding, row, and column movement is carried out on the matrix of the three channels.

Step 12: To create a ciphertext image, a three-dimensional matrix is formed by concatenating three two-dimensional matrices with shifted rows and columns.

5.2.2 The Algorithm for Decryption

The decrypted image must be retrieved using the same key as the one used for encryption, ensuring no differences between the original and decrypted images. The flowchart for the decryption algorithm is shown in Figure 5.2, with the decryption outlined below.

Step 1: Split encrypted image into three two-dimensional matrices representing the R, G, and B channels.

Step 2: To obtain two keys that are consistent with the initial values used in encryption, generate them from two LSAIM chaotic sequences initialized with r_1 and r_2 , both set to 6.

Step 3: For each of the three R, G, and B channels, change the rows and columns in the matrix by replacing the rows and columns in the opposite order as the encryption.

Step 4: The rows and columns R, G, and B three channels are replaced in the opposite direction from how they were encrypted. The last column is replaced, followed by the first column's end, and the last row is replaced, followed by the first row's end. Carry out a row permutation.

Step 5: Create a 2D matrix of dimensions $L \times M$ using the generated chaotic sequences.

Step 6: Four sequences are calculated for the Hyperchaotic memristor system using Matlab's built-in Runge-Kutta function, ode45. These sequences serve as keys.

Step 7: The matrix is DNA encoded. One of the encrypted sequences is used to ascertain the DNA encoding technique for the cipher text image because the image is ultimately encrypted using the DNA decoding technique discovered by the given encrypted sequence.

Step 8: Perform block-wise DNA inverse calculation. Contrary to encryption, however, subtraction is employed in this situation if addition is used in encryption, and vice versa.

Step 9: After performing the operation, apply DNA decoding to the three generated two-dimensional matrices.

Step 10: To increase the algorithm's universality, remove the zero pixels that were created during encryption.

Step 11: To obtain the decrypted color digital image, combine the three matrices into a three-dimensional matrix after eliminating any zeros.

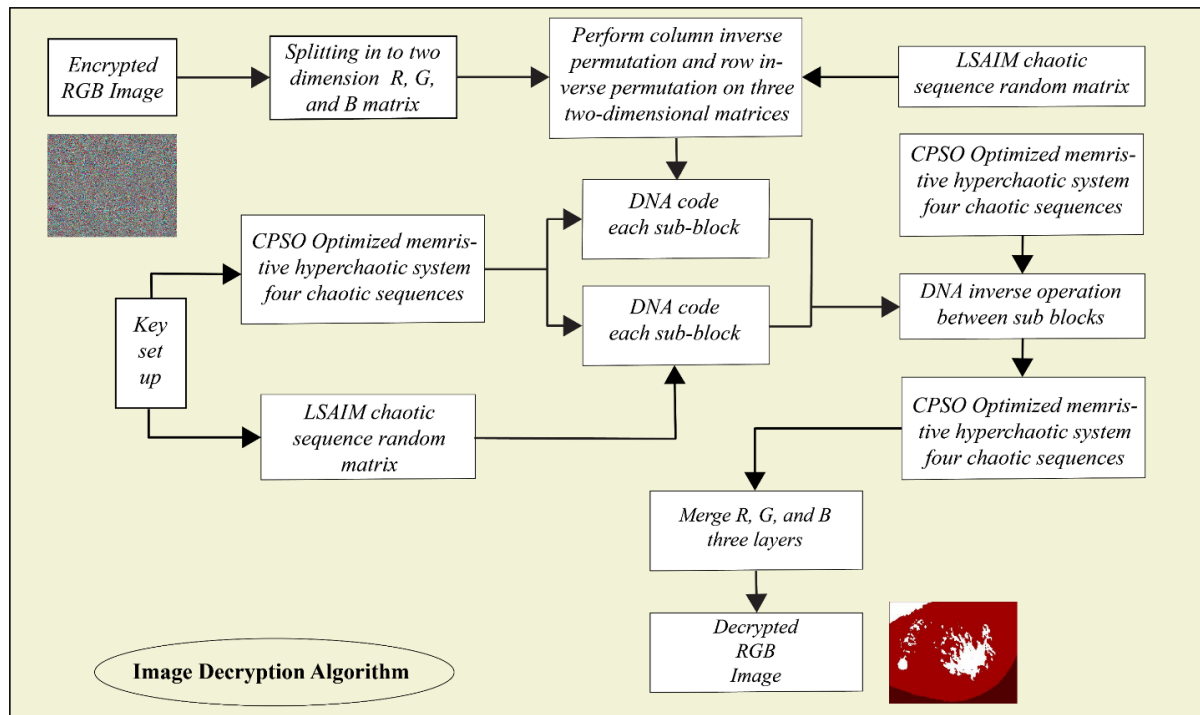


Figure 5.2: Flow Diagram for the Suggested Decryption of RGB Images

5.3 Results and Discussion

All experimental results were performed on a personal computer running Microsoft Windows 10 Pro outfitted with an Intel(R) Core(TM) i7 CPU @ 3.20GHz processor and 8G of RAM. Using the Matlab R2023a platform for simulation, the encryption efficiency evaluation of the test images taken from USC-SIPI data with a 512×512 size is carried out, and processed breast cancer medical images (Normal, Malignant tumor, and Benign tumor) are used as dataset (Wei et al., 2020). The memristor base hyperchaotic system parameters are $\alpha=1$, $\beta=1$, $\gamma=9.98$, $\delta=1$, $\epsilon=3$, $f=3.75$, $\epsilon=-3$, $n_0=-1.2$, $n_1=0.1$, and $K=3.05$. The LSAIM parameter value of $r_1 = 6$, $r_2 = 6$ and, initial value y_0 . Figure 5.3 displays the original images, the encrypted images, and the decrypted images with their histograms. The original image information is effectively hidden by the suggested method.

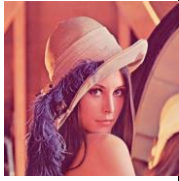
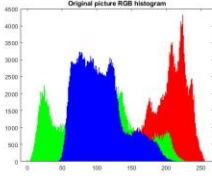
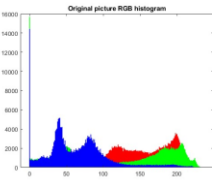
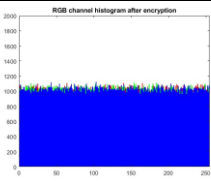
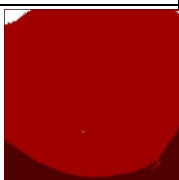
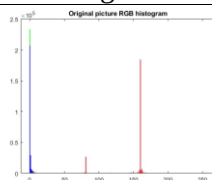
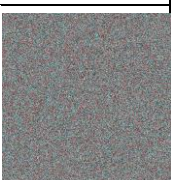
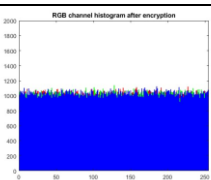
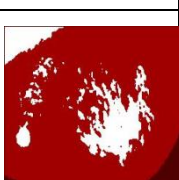
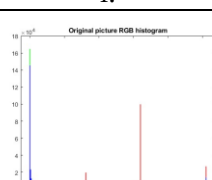
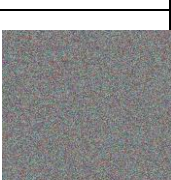
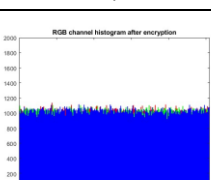
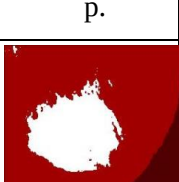
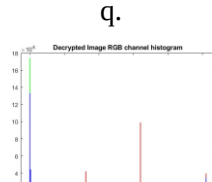
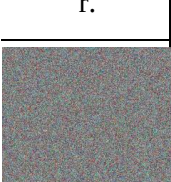
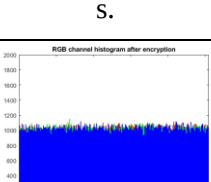
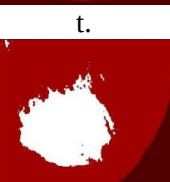
Input Images	Original images	Original image RGB Histogram	Encrypted images	Encrypted image RGB Histogram	Decrypted images
Lena					
	a.	b.	c.	d.	e.
Pepper					
	f.	g.	h.	i.	j.
Normal					
	k.	l.	m.	n.	o.
Malignant tumor					
	p.	q.	r.	s.	t.
Benign tumor					
	u.	v.	w.	x.	y.

Figure 5.3: Original, Encrypted, and Decrypted Image Results with Their Histograms

5.3.1 Analysis of Performance and Security

5.3.1.1 Keyspace

The twenty one variables constituting the hidden keys in our proposed encryption system are as follows: the three values of the LSAIM chaotic sequences generated from $r_1 = 6$, $r_2 = 6$ and, the initial value y_0 ; zero padded values for the rows and columns (L1 and W1); the average gray values of the G and B channels; the initial values of the memristive hyperchaotic system (X0, Y0, Z0, and H0); and the parameters of the memristor hyperchaotic system, which include $\alpha=1$, $\beta=1$, $\gamma=9.98$, $\Psi=1$, $e=3$, $f=3.75$, $\varepsilon=-3$, $n_0=-1.2$, $n_1=0.1$, and $K=3.05$. Our proposed image encryption method effectively achieves a key space that exceeds the capabilities of exhaustive cracking, as demonstrated in the referenced literature (Alexan, Elkandoz, et al., 2023).

We assume in our research that the maximum machine performance is considered to be 10^{-16} (Alexan, Gabr, et al., 2023), with an overall of 21 variables employed as keys. The proposed encryption of images algorithm theoretically achieved a key length of, $10^{21 \times 16} = 10^{336} \approx (2^{3.3219282})^{336} \approx 2^{1116}$, which is much greater than the minimum required value of 2^{100} . The result is compared with those of its counterparts from the literature in Table 5.1. Therefore, brute force assaults are not able to defeat this algorithm.

Table 5.1: An Analysis of Key Spaces Comparison

Algorithms	Keyspace values
Proposed	2^{1116}
Ref. (Hao et al., 2021)	2^{407}
Ref. (X. Gao et al., 2022)	2^{624}
Ref. (Yousif et al., 2022)	2^{747}
Ref. (H. System & Sequence, 2022)	$2^{13 \times 32}$
Ref. (Nazir et al., 2022)	2^{390}
Ref. (M. Liu & Ye, 2021b)	2^{186}

5.3.1.2 Key Sensitivity

The original picture cannot be adequately deciphered with a slight alteration in the key for decryption, demonstrating the high sensitivity of the algorithm (Wan et al., 2020). Adjust the secret encryption keys initially. Subsequently, one of the encryption keys is modified slightly. For instance, the LSAIM parameter $r_1 = 6$ used as a key is modified as $r_1 = 6 \pm 10^{-15}$ to enable a comparison of the outcomes, as illustrated in Figure 5.4. Even a tiny adjustment can have a significant impact. Furthermore, the decryption picture differs entirely from the source picture. As a result, it can be said that guessing the encryption key exactly is not possible to decrypt.

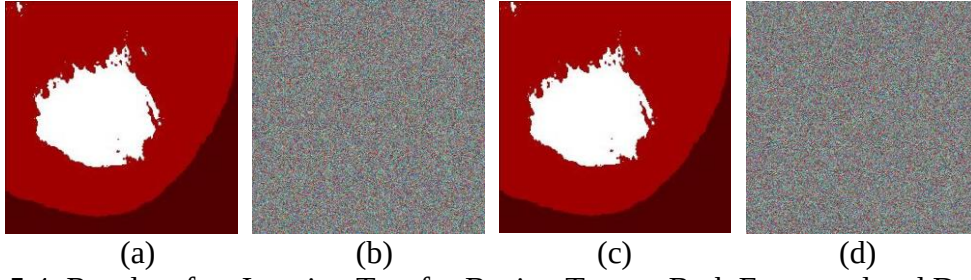


Figure 5.4: Results of an Imaging Test for Benign Tumor, Both Encrypted and Decrypted
(a) A Tumor Image, (b) Encrypted Tumor Image, (c) Decrypted Tumor Image, (d) Decrypted Using a Somewhat Different Key

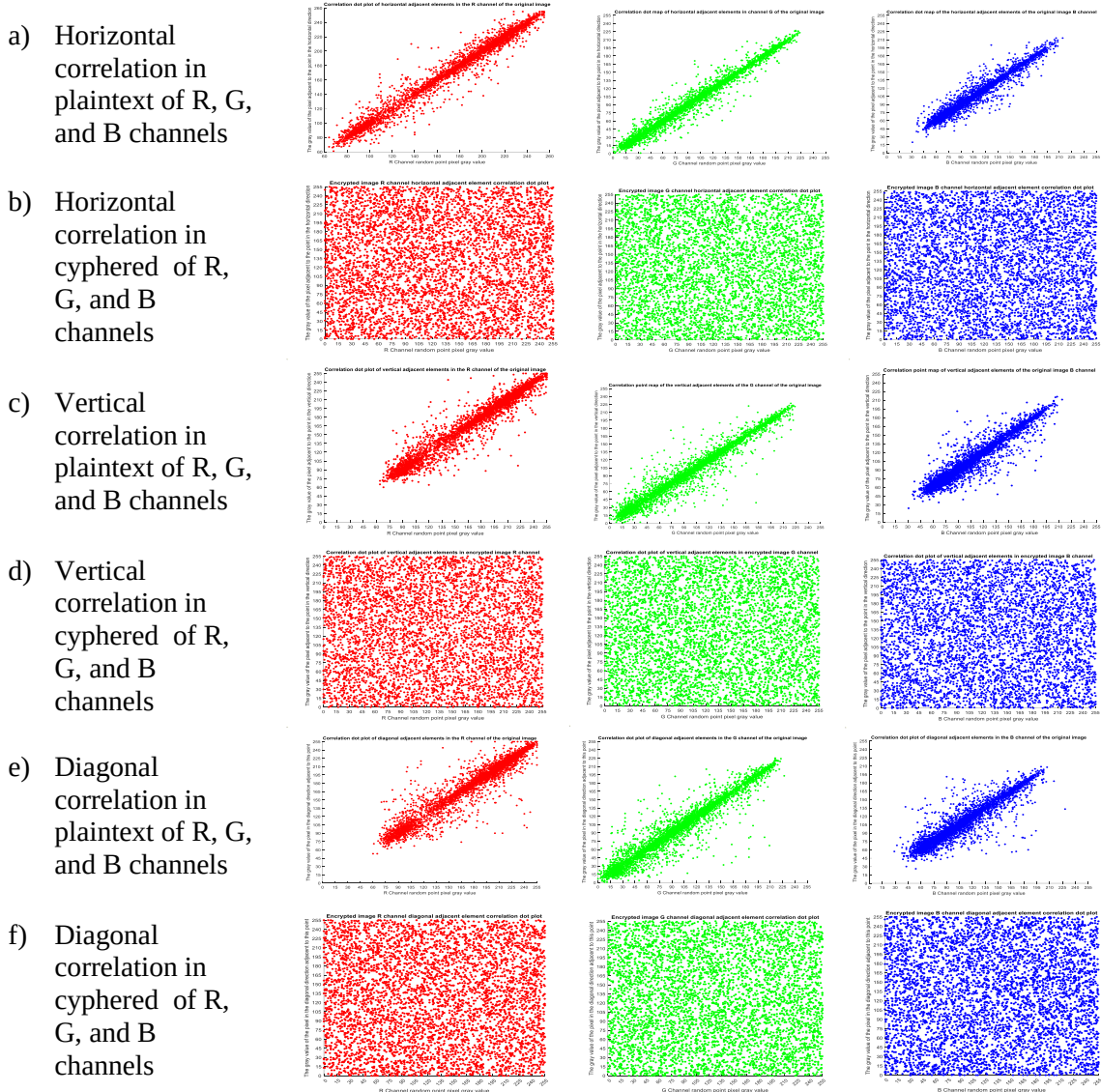


Figure 5.5: Pepper's Encrypted Image and Plaintext Image Correlation Distributions

5.3.2 Statistical Analysis

5.3.2.1 Analysis of Histogram

The spatial distribution of picture pixel values can be described using an image histogram (Y. Yang et al., 2021). The RGB component histograms of the test images are shown in Figure 5.3.

These figures show that the histogram of a normal image is irregular and filled with spikes, whereas the histogram of an encrypted image is nearly regular for various color components. It illustrates how completely the encryption method conceals the redundancy of the original image. As a result, histogram analysis is typically used to assess an image encryption system's encryption effectiveness as well as its resilience to statistical attacks.

5.3.2.2 Correlation Analysis

Attackers can use correlation analysis to collect cipher text information, but plaintext images possess more excellent adjacent pixel correlation (Jithin & Sankar, 2020; Z. Liu et al., 2019). Moreover, results for the sampled Pepper image are displayed in Figure 5.5.

Table 5.2: Pixel Correlation Values for Test Images that are Encrypted and the Source.

Images	Direction	Coefficients of correlation between pixels					
		Original images			Encrypted images		
		Red	Green	Blue	Red	Green	Blue
Lena	Horizontal (H)	0.98043	0.97131	0.93869	0.008151	-0.004226	0.016964
	Vertical (V)	0.98993	0.98356	0.96085	0.003069	0.002301	-0.004825
	Diagonal (D)	0.97101	0.95651	0.91988	0.032255	-0.001475	-0.011047
Pepper	Horizontal (H)	0.95904	0.98141	0.96649	0.007908	-0.000998	0.001213
	Vertical (V)	0.96581	0.97988	0.96655	-0.03009	0.001761	0.007952
	Diagonal (D)	0.95224	0.96682	0.94720	-0.01772	0.019682	0.004762
Normal	Horizontal (H)	0.99766	0.99944	0.99938	-0.01594	-0.005337	-0.003737
	Vertical (V)	0.98643	0.97126	0.97200	0.017763	0.011538	0.008327
	Diagonal (D)	0.98335	0.97078	0.97129	0.023080	-0.006172	-0.018226
Malignant tumors	Horizontal (H)	0.92161	0.95606	0.95708	0.002486	0.025902	0.026395
	Vertical (V)	0.94197	0.96770	0.96846	0.020938	0.004613	-0.002927
	Diagonal (D)	0.89054	0.93612	0.93825	0.002404	-0.006592	-0.015490
Benign tumors	Horizontal (H)	0.97705	0.98566	0.98603	0.031158	-0.004117	-0.010782
	Vertical (V)	0.98289	0.98890	0.98928	-0.008596	-0.018879	0.016036
	Diagonal (D)	0.97017	0.98119	0.98190	-0.010971	-0.004710	-0.005549

The calculation formula of the correlation coefficient r_{xy} is shown in equation (5.1) (T. Wang & Wang, 2020a).

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}} \quad (5.1)$$

where,

$$\text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))(y_i - E(y)) \quad (5.2)$$

$$E(x) = \frac{1}{N} \sum_{i=1}^N x_i \quad (5.3)$$

$$D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - E(x))^2 \quad (5.4)$$

where N is the overall number of image pixels chosen from the image, and x and y represent the grayscale for the two neighboring pixels in the image. Using proposed test images, Table 5.2 displays the outcomes and a comparison to various algorithms. It can be seen that while the encrypted image coefficients of correlation are roughly 0 in each of the three directions, those of the original picture are close to 1.

5.3.3 Analysis of Information Entropy

Information entropy analysis is a beneficial tool for measuring the amount of useful information in a picture. There is more ambiguity and randomness in the image when the information entropy value is higher (S. Zhang et al., 2023). The formula for calculating information entropy using a standard method is,

$$H(x) = - \sum_{i=0}^{2^N-1} P(x_i) \log_2 P(x_i) \quad (5.5)$$

The ratio of the image's overall gray level between them is shown by $P(x_i)$. $H_{\max} = 8$ is the maximal entropy of a picture with a gray level of M . Table 5.3 compares the information entropy of the unencrypted and encrypted photos.

Table 5.3: Information Entropy Comparison Between Cyphered and Original Images

Images		Red	Green	Blue
Lena	Original images	7.2531	7.594	6.9684
	Encrypted images	7.9994	7.9993	7.9993
Pepper	Original images	7.3388	7.4963	7.0583
	Encrypted images	7.9994	7.9992	7.9994
Normal	Original images	2.1962	0.84705	1.3018
	Encrypted images	7.9994	7.9994	7.9994
Malignant tumors	Original images	4.3595	2.7894	3.1367
	Encrypted images	7.9994	7.9993	7.9992
Benign tumors	Original images	3.691	2.1605	2.6703
	Encrypted images	7.9993	7.9993	7.9992

Table 5.4: Comparing Pepper Test Image Entropy Results with Alternative Methods

Lena encrypted image	Entropy value		
	Red	Green	Blue
Maximum	8.0000	8.0000	8.0000
Proposed	7.9994	7.9993	7.9994
Ref. (X. Wang & Du, 2022)	7.9967	7.9967	7.9967
Ref. (Amani & Yaghoobi, 2019)	7.9971	7.9973	7.9974
Ref. (M. A. Lone & Qureshi, 2023)	7.9974	7.9974	7.9976
Ref. (Senthilkumar et al., 2023)	7.9110	7.9110	7.9110
Ref. (M. Li et al., 2022)	7.9974	7.9965	7.9970
Ref. (Z. Liang et al., 2023b)	7.9993	7.9992	7.9992
Ref. (Valandar et al., 2019)	7.9989	7.9991	7.9989

The entropy of the picture makes this encryption technique resistant to attacks (X. Wang & Du, 2022). Table 5.4 shows the results of the entropy comparison for the Pepper test image.

5.3.4 Differential Attack Analysis

In this study, the original image's intensity values are slightly changed, and the same algorithm is used to encrypt both images. The hacker attempts to determine the relationship between the encrypted and plain images using this analysis. To appraise the efficacy of the system in mitigating differential attacks, two metrics are examined: The Unified Average Change in Intensity (UACI) and the Number of Pixel Changing Rate (NPCR) (M. A. Lone & Qureshi, 2023). These formulas are given in Equations (5.6) and (5.7), respectively:

$$NPCR = \frac{1}{L \times W} \sum_{r,c} G_{r,c} \times 100\% \quad (5.6)$$

$$UACI = \frac{1}{L \times W} \sum_{r,c} \frac{|F'_{r,c} - F''_{r,c}|}{255} \times 100\% \quad (5.7)$$

Where, $L \times W$ is the picture size, $F'_{r,c}$ and $F''_{r,c}$ represent two encrypted images made from the identical input image but varied by just one pixel, and $G_{r,c}$ is determined by equation (5.8):

$$G_{r,c} = \begin{cases} 1 & \text{if } F'_{r,c} \neq F''_{r,c} \\ 0 & \text{else} \end{cases} \quad (5.8)$$

Table 5.5: Comparative Values of UACI and NPCR for Cyphered Image

Algorithm	NPCR				UACI			
	Red	Green	Blue	Average	Red	Green	Blue	Average
Proposed	99.617%	99.605%	99.6315%	99.61783	33.8340	33.9648	28.9844	32.26107
Acceptable								
Value (Suri & Vijay, 2019)	99.000%	99.000%	99.000%	99.000%	30.0000	30.0000	30.0000	30.0000
Ref. (Cryptography & Lawnik, 2022)	99.240%	99.240%	99.250%	99.24333	32.7800	32.8500	32.7800	32.80333
Ref. (Z. Liu et al., 2019)	99.610%	99.620%	99.620%	99.61667	28.9200	33.8300	33.8400	32.19667
Ref. (Alexan, Elkandoz, et al., 2023)	99.6254	99.6254	99.6254	99.6254	33.0704	30.7620	27.8720	VVVV
Ref. (Elkandoz & Alexan, 2022)	99.571%	99.632%	99.612%	99.60500	29.0978	33.8751	33.7068	32.22657
Ref. (Ramli et al., 2020)	99.610%	99.620%	99.600%	99.61000	33.3800	33.3900	33.4800	33.41667
Ref. (Patel et al., 2020)	99.603%	99.603%	99.603%	99.60300	27.3904	27.3904	27.3904	27.39040
Ref. (Yildirim, 2020)	98.004%	98.051%	99.594%	98.54967	32.2856	35.3837	42.1522	36.60717

For NPCR and UACI, the average results are 99 and 30, respectively. It demonstrates how well the newly developed technique can withstand differential and plain-text attacks because of its remarkable sensitivity to even minute changes in the picture (Suri & Vijay, 2019). Table 5.5 shows the proposed algorithm outcomes of the UACI and NPCR, respectively. The computational results from UACI and NPCR adequately surpass the conventional values in theory. It offers proof of the precision and coherence of the NPCR and UACI value results numerically. And also the comparisons shown in Table 5.5 imply that differential attack defense is a feature of the suggested encryption technique.

5.3.5 Mean Square Error

Because of the encryption technique being applied over the original image, which undoubtedly adds a certain amount of noise to the real digital content, an image that has been encrypted shouldn't be equal to the initial digital picture. To assess the enciphering level, we calculate the mean square error (MSE) of the initial original and cyphered pictures. The definition of MSE in mathematics is:

$$MSE = \frac{\sum_{i=1}^m \sum_{j=1}^n (P_{ij} - C_{ij})^2}{m \times n} \quad (5.9)$$

Where P_{ij} and C_{ij} are the image number of pixels in the original and encrypted pictures' corresponding row (i) and column (j). An increased MSE value improves the encryption scheme's security.

5.3.6 Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) is a key metric for evaluating image cryptosystems. The measurement of the ratio of the square of the highest value of a pixel ($I_{\max}$) to the mean square error (MSE) serves as the basis for its computation (Alexan, Gabr, et al., 2023). In mathematical terms, it is represented as,

$$PSNR = 10 \log \left(\frac{I_{\max}^2}{MSE} \right) \quad (5.10)$$

Where $I_{\max}$, which has a value of 255, represents the highest grayscale intensity. Smaller PSNR values indicate better encryption since PSNR and MSE have an inverse relationship. The calculated PSNR values for the suggested cryptosystem are shown in Table 5.6, which also includes a comparison to equivalent algorithms' results from the available research. The obtained PSNR values match those reported in the literature.

Table 5.6: MSE and PSNR Between Original and Encrypted Test Images

Test Images	Pixel Difference-Based Measures							
	MSE (dB)				PSNR (dB)			
	Red	Green	Blue	Average	Red	Green	Blue	Average
Lena	10676.01	9083.17	7086.47	8948.550	7.85	8.55	9.63	8.61
Pepper	7996.210	11240.35	11158.49	10131.68	9.10	7.62	7.65	8.07
Normal	6765.016	21646.02	21599.78	16670.27	9.83	4.78	4.79	5.91
Malignant tumors	8791.44	21050.87	20892.95	16911.75	8.69	4.90	4.93	5.85
Benign tumors	9317.11	21460.10	21376.16	17384.46	8.44	4.81	4.83	5.73
Pepper Ref. (Alexan, Gabr, et al., 2023)	10049.4	10049.4	10049.4	10049.4	8.11	8.11	8.11	8.11
Pepper Ref. (Yousif et al., 2022)	7869.40	7869.40	7869.40	7869.40	9.17	9.17	9.17	9.17
Pepper Ref. (Alexan, Elkandoz, et al., 2023)	8049.574	11016.54	11032.92	10033.01	9.073	7.71	7.704	8.16
Pepper Ref. (Paul et al., 2022)	-----	-----	-----	-----	8.31	8.31	8.31	8.31
Pepper Ref. (Elkandoz & Alexan, 2022)	11179.62	11062.95	8417.52	10220.03	7.65	7.70	8.88	8.077

5.3.7 Mean Absolute Error

The Mean Absolute Error (MAE) is another statistic that is used to evaluate how resistant an encryption method is against differential attacks. The MAE calculates the average difference for each $L \times W$ pixel in an image between the original image pixel from $P(i,j)$ and its cyphered counterpart $C(i,j)$. In mathematical terms, it is represented as;

$$MAE = \frac{1}{L \times W} \sum_{i=1}^L \sum_{j=1}^W [P(i, j) - E(i, j)] \quad (5.11)$$

The encrypted picture pixels in row i and column j are referred to as $E(i,j)$, whereas the plain image pixels are denoted as $P(i,j)$. The dimensions of the image are represented by L and W . The suggested encryption scheme would be more resistant to differential assaults the higher the MAE number.

5.3.8 Structural Similarity Index Measure

The structural similarity index measure (SSIM) is used to determine how similar the processed image (E) and original image (O) (Khushwant et al., 2023). It is given as,

$$SSIM(O, E) = \frac{(2\mu_O\mu_E + \alpha)(2\sigma_{OE} + \beta)}{(\mu_O^2 + \mu_E^2 + \alpha)(\sigma_O^2 + \sigma_E^2 + \beta)} \quad (5.12)$$

The constants are represented by α and β , whereas the mean, variance, and covariance between O and E are denoted by μ , σ^2 , and σ , respectively. A comparison of the proposed algorithm computed MAE values and those of comparable systems is shown in Table 5.7. Additionally,

the table demonstrates that the suggested scheme's Structural Similarity Index Measure (SSIM) is either smaller than or comparable to similar schemes discovered in earlier research.

Table 5.7: MAE and SSIM of Original and Encrypted Test Images

Test Images	MAE (dB) between Original and Encrypted				SSIM
	R	G	B	Average	
Lena	84.41	78.16	70.30	77.62	0.0074154
Pepper	73.91	86.61	86.28	82.27	0.0061873
Normal	69.10	127.26	127.03	107.797	0.0044314
Malignant tumors	77.04	124.85	124.28	108.72	0.0059425
Benign tumors	79.03	126.54	126.19	110.587	0.0050818
Pepper Ref. (Alexan, Gabr, et al., 2023)	81.9	81.9	81.9	81.9	-----
Pepper Ref. (Yousif et al., 2022)	73.44	73.44	73.44	73.44	0.0415000
Pepper Ref. (Alexan, Elkandoz, et al., 2023)	77.99	81.84	75.46	78.43	
Pepper Ref. (Paul et al., 2022)	-----	-----	-----	-----	0.0514000
Pepper Ref. (Elkandoz & Alexan, 2022)	82.18	82.18	82.18	82.18	-----

The decrypted images closely matched the original test images, as shown by the computed metrics in Table 5.8. Each test image achieved an MSE of 0, PSNR of infinity, and SSIM of 1 across all color channels, indicating a perfect retrieval of the original images after encryption.

Table 5.8: MSE, PSNR, and SSIM of Decrypted Images

Test Images	MSE (dB)				PSNR (dB)				SSIM
	R	G	B	Average	R	G	B	Average	
Lena	0.0000	0.0000	0.0000	0.00	Inf	Inf	Inf	Inf	1
Pepper	0.0000	0.0000	0.0000	0.00	Inf	Inf	Inf	Inf	1
Normal	0.0000	0.0000	0.0000	0.00	Inf	Inf	Inf	Inf	1
Malignant tumors	0.0000	0.0000	0.0000	0.00	Inf	Inf	Inf	Inf	1
Benign tumors	0.0000	0.0000	0.0000	0.00	Inf	Inf	Inf	Inf	1

5.3.9 Computational time analysis of the proposed algorithm

The computational time for the proposed encryption algorithm exhibits slight variations across different iterations due to the dynamic behavior of the hyperchaotic system integrated into the algorithm. Table 5.9 shows the computational time recorded over six iterations across various sizes of the Lena test image, with corresponding average values. The results indicate that while computational time increases with image size, the algorithm maintains robust encryption performance. The average times are 0.11371s, 0.18241s, and 0.41648s for 256×256 , 512×512 , and 1024×1024 images, respectively, demonstrating its efficiency for secure application.

Table 5.9: Computational Time for Six Iterations and Average

Iterations	Image Size	1 st	2 nd	3 rd	4 th	5 th	6 th	Average
	$256 \times 256 \times 3$	0.11477	0.11454	0.11183	0.11686	0.11320	0.11106	0.11371
Time (s)	$512 \times 512 \times 3$	0.18546	0.18206	0.17894	0.18198	0.18656	0.17948	0.182413
	$1024 \times 1024 \times 3$	0.43070	0.40461	0.43217	0.41915	0.41071	0.40156	0.416483

5.4 Evaluation of Randomness Using the NIST Statistical Test Suite

The NIST SP800-22 Statistical Test Suite (STS) evaluates the randomness of an encrypted image's bitstream by assessing properties such as uniformity, independence, and entropy to confirm true randomness (Gedda et al., 1977).

Table 5.10: NIST STS Results for the Encrypted Image Bit Stream

No	Test Name		P-Value	Proportion	Accepted/Rejected
1	Frequency (Monobit) Test		0.980853	100%	Accepted
2	Block Frequency Test		0.509144	100%	Accepted
3	Cumulative Sums (Cusum) Test	Forward Reverse	0.436202 0.454277	100%	Accepted
4	Runs Test		0.365995	100%	Accepted
5	Longest Run of Ones in a Block Test		0.537254	100%	Accepted
6	Rank Test		0.199572	100%	Accepted
7	Discrete Fourier Transform (Spectral) Test (FFT)		0.588217	100%	Accepted
8	Non-overlapping Template Matching Test Average		0.551400	100%	Accepted
9	Overlapping Template Matching Test		0.423948	100%	Accepted
10	Universal Statistical Test (Maurer's)		0.854535	100%	Accepted
11	Approximate Entropy Test		0.768092	100%	Accepted
12	Random Excursions Test Average		0.3742	100%	Accepted
13	Random Excursions Variant Test Average		0.5753	100%	Accepted
14	Serial Test	p_value1 p_value2	0.047802 0.012885	100%	Accepted
15	Linear Complexity Test		0.949315	100%	Accepted

The analysis of the bit stream in ASCII mode, represented as '0' and '1', yielded results of either pass (100%) or fail (0%). Table 5.10 shows that all tests achieved a p-value of ≥ 0.01 , indicating no patterns in the binary representation of the encrypted image. This result confirms the encryption's security and supports the findings of (Gedda et al., 1977) and (X. Wu et al., 2025) regarding the statistical randomness of the bit stream.

5.5 Performance Validation and Comparative Security Analysis

This section comprehensively evaluates the proposed image encryption framework, integrating a memcapacitive hyperchaotic system and DNA coding. The scheme is rigorously tested across standard and medical images using multiple statistical and security benchmarks. The results confirm the robustness and superiority of the proposed system in terms of keyspace size, correlation minimization, entropy maximization, noise resistance, and image recovery accuracy. Key findings include an exceptional keyspace of 2^{1116} , correlation coefficients near zero, and information entropy values approaching the ideal maximum of 8.0000 across all channels. Additionally, the proposed method achieves average NPCR and UACI scores of 99.62% and 32.26%, respectively, outperforming multiple contemporary techniques. The decrypted images also report zero MSE and SSIM = 1, indicating perfect reconstruction without loss. Table 5.10 summarizes the performance metrics of the proposed image encryption scheme, highlighting its high security, randomness, and reversibility.

Table 5.11: Summary of Performance Evaluation Metrics

Metric	Achieved Value (Range or Avg.)	Ideal or Acceptable Standard	Performance Interpretation
Key Space	2^{1116}	2^{100}	Extremely large key space, highly resistant to brute-force attacks
Pixel Correlation (Encrypted)	~ 0 to ± 0.03	Close to 0 or negative	Strong decorrelation achieved in all RGB directions
Information Entropy	7.9992 – 7.9994	≈ 8	Near-perfect randomness of encrypted images
UACI (%)	28.98 – 33.96 (Avg: 32.26)	$> 30\%$	High sensitivity to pixel changes; excellent diffusion
NPCR (%)	99.605 – 99.631 (Avg: 99.6178)	$> 99\%$	Excellent resistance against differential attacks
MSE (Encrypted)	8948 – 17384 (Avg: ~ 12100)	High value indicates distortion	Sufficient distortion from original; good confusion
PSNR (Encrypted)	5.73 – 8.61 dB	< 10 dB (lower is better)	Poor visual similarity; strong encryption
MAE (Encrypted)	77.6 – 110.6 (Avg: ~ 91.2)	High indicates difference	Large mean difference from original; effective encryption
SSIM (Encrypted)	~ 0.0044 – 0.0074	Close to 0	Very low structural similarity; excellent transformation
MSE (Decrypted)	0	0	Perfect reconstruction of original
PSNR (Decrypted)	∞	∞	Indicates no error in decryption
SSIM (Decrypted)	1	1	Perfect structural similarity after decryption
Encryption Time for Image with Size ($512 \times 512 \times 3$)			0.182413 seconds

5.6 Summary

This chapter presented a new approach to optimizing the parameters of memristor-based hyperchaotic systems with enhanced complexity and security features seemingly for colour image encryption. Employing CPSO and integrating chaotic maps achieved precise tuning of

the system parameters, maximizing hyperchaotic behaviour and augmenting the system's unpredictability. The optimized hyperchaotic system combined with DNA sequence operations and LSAIM Map formed a robust colour image encryption method. The finding showed that the proposed approach is highly resistant to statistical, differential, and brute-force attacks, underscoring its reliability for secure communication and data protection. The proposed encryption approach can be used for high-stakes applications, including confidential imaging, medical data security, military image transfers as communications, and remote sensing. This research can be utilized for future studies in secure, hyperchaos-based image encryption, and it showed significant potential to enhance the confidentiality and integrity of sensitive information across various fields.

CHAPTER 6

DENOISING ENCRYPTED IMAGES USING ONLINE SEQUENTIAL EXTREME LEARNING MACHINE WITH A MEMELEMNTS-BASED HYPERCHAOTIC SYSTEM

6.1 Introduction

Conventional encryption techniques frequently encounter difficulties when dealing with noise and maintaining image quality during transmission. This chapter proposed an innovative approach to image encryption that integrates an Online Sequential Extreme Learning Machine (OSELM) autoencoder and chaotic systems with DNA code for enhanced image encryption and denoising. The proposed method amalgamates an OSELM autoencoder, a hyperchaotic system incorporating two mersisters, a two-dimensional sine map, and DNA coding to effectively protect images from various types of noise such as Gaussian, Salt and Pepper, Quantization, Speckle, Sensor, and Environmental noises. The OSELM-based denoising technique improves encrypted images' resistance to noise attacks while ensuring high security and quality. With a large key space and nearly ideal entropy values, it effectively counters differential and statistical attacks. Robustness is confirmed through histogram and correlation tests, and NIST SP800-22 tests validate the randomness of the encrypted bitstream. This method addresses image security and noise resilience, making it ideal for secure transmission and storage.

6.1.1 Online Sequential Extreme Learning Machine

OSELM is a real-time algorithm that incrementally updates its model, allowing rapid adaptation and learning. It achieves high speed, good generalization, and scalability by using random input weights and analytical output calculations, efficiently processing data in small chunks. OSELM's incremental learning reduces computational complexity (Al-Dhief et al., 2021; Alizamir, Othman Ahmed, et al., 2023). OSELM consists of three key layers: input, hidden, and output (Ou et al., 2021). The hidden-layer output matrix (H) is calculated as shown in Equation (6.1).

$$H = W X + B \quad (6.1)$$

where, X represents the input vector, W denotes the input weights, and B the biases of the hidden layer, both stochastically generated within $(-1, 1)$. For N unique samples (x_j, t_j) , single-layer feedforward networks (SLFNs) with n hidden neurons and an activation function $g(x)$ can be expressed mathematically as follows:

$$f(X) = \sum_{i=1}^n \beta_i g(W_i x_j + b_i) = t_j \quad (6.2)$$

where, $j = 1, 2, \dots, N$

Additionally, Equation (6.2) can be reformulated as follows:

$$H\beta = T \quad (6.3)$$

$$H = \begin{bmatrix} g(W_1x_1 + b_1) & \cdots & g(W_nx_1 + b_1) \\ \vdots & \ddots & \vdots \\ g(W_1x_N + b_1) & \cdots & g(W_nx_N + b_n) \end{bmatrix}_{N \times n} \quad (6.4)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_n^T \end{bmatrix}_{n \times m} \quad (6.5)$$

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (6.6)$$

The output weight $\hat{\beta}$ is then computed based on the following Equation (6.7) as follows:

$$\hat{\beta} = H^+T \quad (6.7)$$

where, H^+ is the inverse of the generalized Moore-Penrose pseudo-inverse for the output matrix of the hidden layer (H), and it is computed as follows:

$$H^+ = (H^TH)^{-1}H^T \quad (6.8)$$

OSELM learns progressively in two steps: initialization and sequential learning. In the initialization step, the initial output weights (β_0), hidden layer output matrix (H_0), and output weights (b_0) are calculated as follows:

$$H_{k+1} = g(W_1X_{k+1} + B_1) \quad (6.9)$$

$$P_0 = (H_0^TH_0)^{-1} \quad (6.10)$$

$$\beta = P_0H_0^TT_0 \quad (6.11)$$

In the sequential learning step, the hidden layer output matrix (H_{k+1}) and output weights matrix (β_{k+1}) are updated for new samples, as shown in Equation (6.13).

$$P_{k+1} = P_k - P_kH_{k+1}^T(1 + H_{k+1}P_kH_{k+1}^T)^{-1}H_{k+1}P_k \quad (6.12)$$

$$\beta_{k+1} = \beta_k + P_{k+1}H_{k+1}^T(T_{k+1} - H_{k+1}\beta_k) \quad (6.13)$$

After training, OSELM predicts unknown inputs by setting $k + 1$ and using Equations (6.9, 6.12, and 6.13) for subsequent samples.

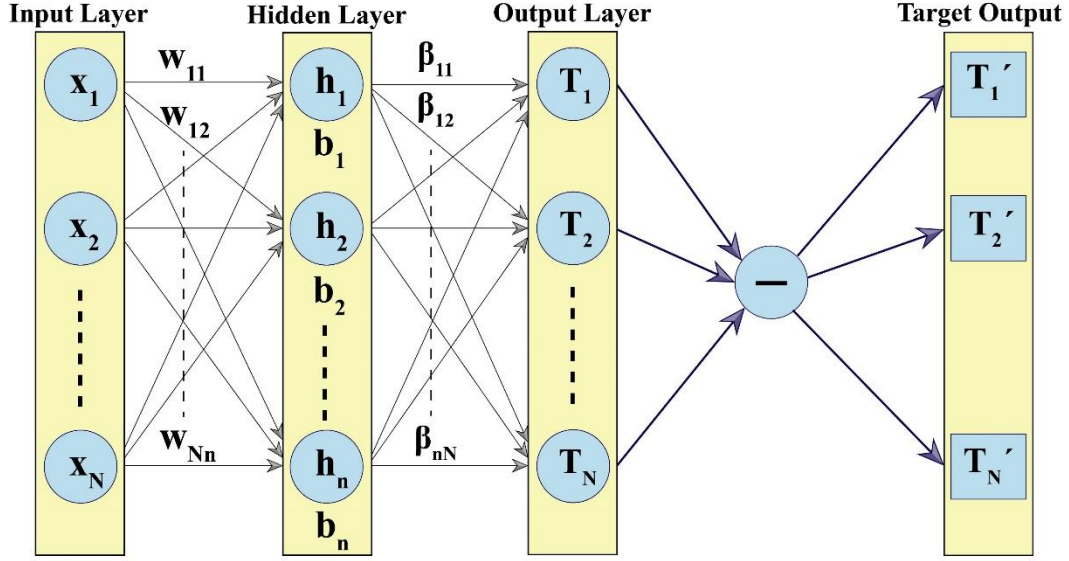


Figure 6.1: OSELM algorithm structure (Ou et al., 2021)

Figure 6.1 shows the OSELM algorithm, where noisy images are the input and denoised images are the output. It represents a neural network designed for image processing, comprising an input layer, hidden layer, output layer, and target output nodes. The input layer consists of nodes $x_1, x_2, \dots, x_N$, each corresponding to distinct pixel values of the input image. In the hidden layer, neurons $h_1, h_2, \dots, h_n$ compute weighted sums of inputs, add biases, and apply non-linear activation functions. The output layer, with nodes $T_1, T_2, \dots, T_N$ generates predictions based on the hidden layer's outputs. Target output nodes $T'_1, T'_2, \dots, T'_N$, represent the actual values the model aims to predict, serving as benchmarks for loss calculation during training. Dashed arrows illustrate the flow of information, while the subtraction symbol (-) denotes the comparison between final outputs and target outputs, essential for error computation and model refinement.

6.1.2 OSELM Auto-Encoder

An autoencoder is an unsupervised neural network with an encoder that compresses data into a latent representation and a decoder that reconstructs the input from this representation (Berahmand et al., 2024). The autoencoder can be represented as:

$$h = (W_{enc}X + b_{enc}) \quad (6.14)$$

$$\hat{X} = g(W_{dec}h + b_{dec}) \quad (6.15)$$

where, $f(\cdot)$ and $g(\cdot)$ are the activation functions of the encoder and decoder, respectively, W_{enc} and W_{dec} are the weights, and b_{enc} and b_{dec} are the biases.

The OSELM Auto-Encoder combines OSELM with autoencoder principles for unsupervised learning, updating parameters sequentially without retraining. It uses random input weights and fast learning, making it ideal for large datasets and real-time tasks like anomaly detection and data compression (Alizamir, Othman Ahmed, et al., 2023). Figure 6.2 shows a block diagram for denoising applications.

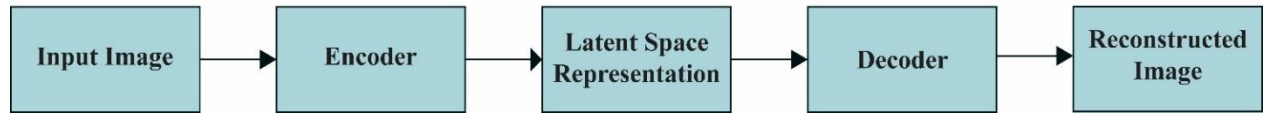


Figure 6.2: Block diagram of image denoising Autoencoder

6.1.3 Major Image Noises

Image noise consists of random pixel variations that degrade image quality. Common varieties comprise Gaussian, Speckle, Salt and Pepper, Sensor, Quantization, Environmental, and Banding noises (Bakour et al., 2024; Banerjee & Shanmugam, 2022; Kirtas et al., 2023; Shen et al., 2023; Youn et al., 2023). Reducing noise is crucial for precise image analysis.

6.2 Proposed Methodology

The study in this chapter integrates a novel method that combines image encryption and denoising to enhance security and quality; the basic block diagram is shown in Figure 6.3. It employs a hyperchaotic system with memristors, bandpass filter, and 2D sine maps for anti-clipping, along with DNA coding. For denoising, an OSELM autoencoder is trained to effectively manage multiple noise types, including Gaussian, Speckle, Salt and Pepper, Sensor, Quantization, and Environmental noises.

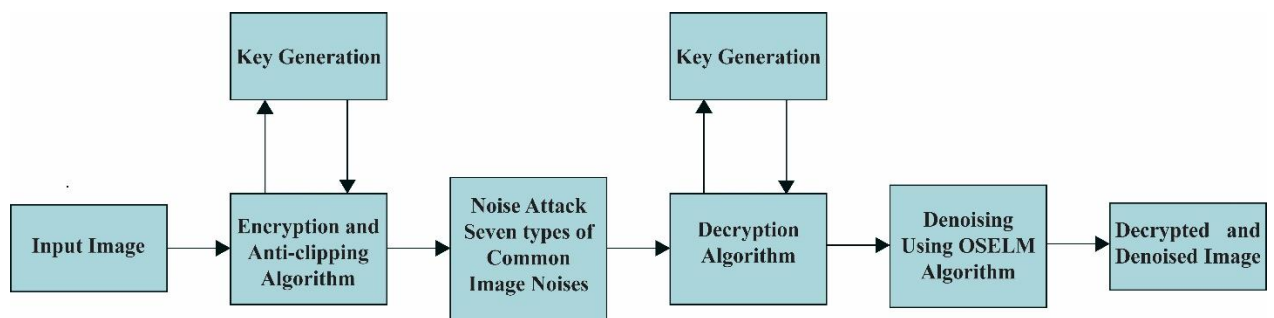


Figure 6.3: Encryption and Denoising Algorithm Blocks

The autoencoder is trained on one noisy image of size 512x512 color images from the different USC-SIPI and Kaggle databases for each noise type, performing 100 iterations for each. The complete noisy image serves as the training data, with the model executing 100 feedforward passes per noise type to facilitate weight updates. The OSELM algorithm employs 100 sigmoid

hidden neurons and a regularization parameter of 0.1 to 0.5, effectively tackling secure, high-quality image processing challenges. Figure 6.4 depicts the processes of encryption, decryption, and denoising.

6.2.1 Proposed Image Encryption and Anti-clipping Algorithm

The proposed image encryption algorithm utilizes a hyperchaotic system featuring dual memristors and a band pass filter to create secure encryption keys. It integrates DNA coding to enhance security and complexity while applying anti-clipping techniques with 2D sine maps to maintain image quality and avoid data loss. This algorithm uses 1D and 2D sine maps as chaotic sequence generators to enhance the security of image encryption and decryption. The 1D sine map produces pseudo-random sequences for DNA encoding, while the 2D sine map generates chaotic sequences for permuting rows and columns. Together, these maps provide high sensitivity to initial conditions, a large key space, and strong resistance to cryptographic attacks, ensuring a secure encryption process.

6.2.2 The Encryption and Anti-clipping Algorithm's Specific Steps

The encryption algorithm starts with preprocessing the image and dividing it into R, G, and B channels. Following zero-padding for a block size of $t=4$, a chaotic sequence is generated using the Sine map. Smaller block sizes, such as $t = 4$ (blocks of 4×4 pixels), ensure finer encryption granularity and improve attack resistance. The initial values from the hyperchaotic system are employed to solve the memristive circuit, producing chaotic sequences. The image blocks undergo DNA encoding, followed by processing through DNA operations and diffusion, after which they are DNA-decoded. An anti-clipping step permutes rows and columns using additional chaotic sequences before the encrypted channels are combined and saved.

Step 1: Read and Preprocess Image reconstruct encrypted images

The algorithm processes the input image by segmenting it into R, G, and B channels. It displays the original image along with histograms for each channel to illustrate pixel intensity distribution. This preparation facilitates the subsequent encryption and anti-clipping processes for data protection.

Step 2: Zero Padding

The process adjusts image dimensions with zero padding to ensure they are divisible by block size $t=4$, allowing uniform encryption and preventing data loss or boundary issues.

Step 3: To generate the Sine Map chaotic sequence, compute the initial value from the image's pixel sum, iteratively apply the formula, discard the first 1000 points for randomness, and convert the sequence into integers (0–255). Reshape it into a matrix for encryption, ensuring chaos for secure transformations.

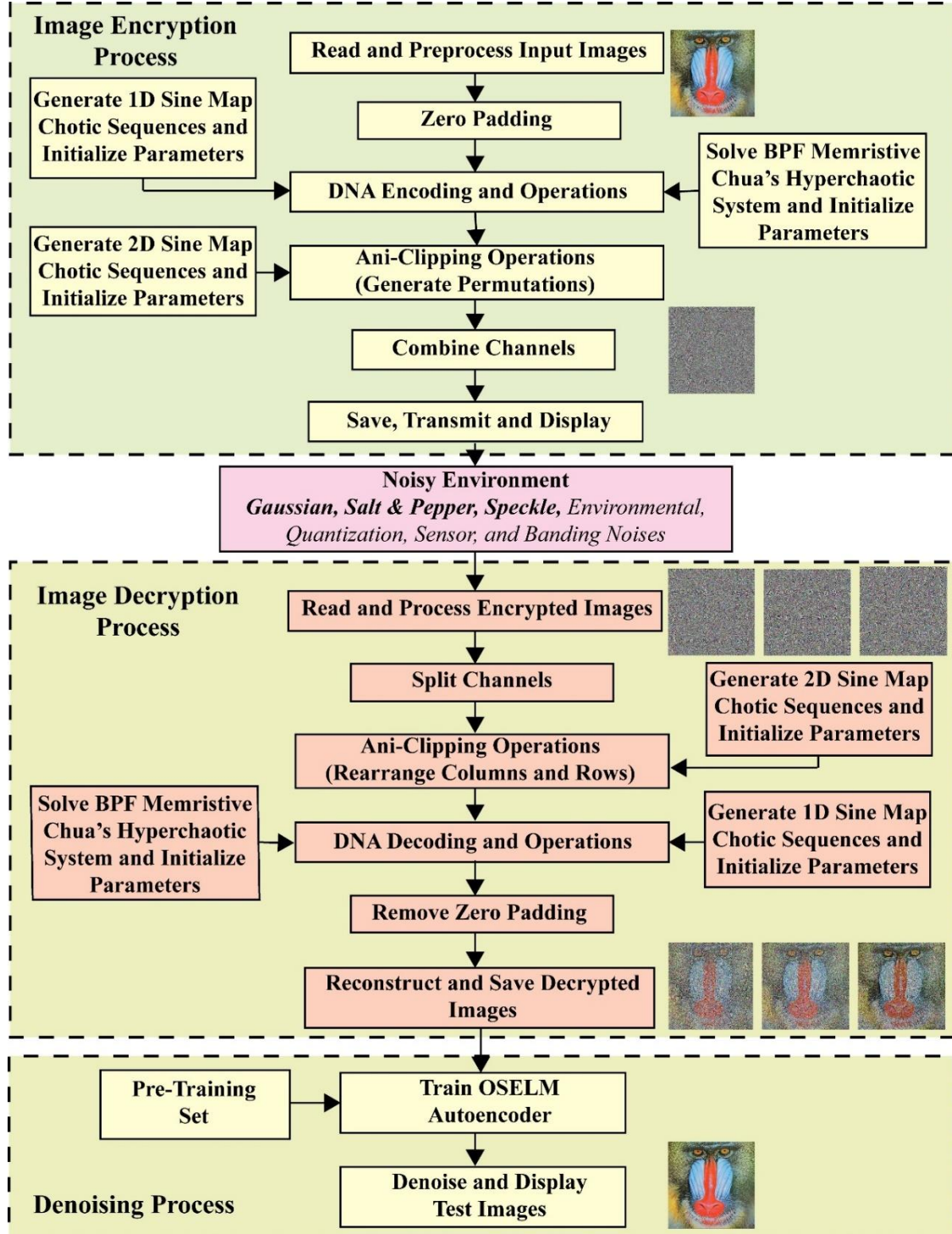


Figure 6.4: The Proposed Encryption, Decryption, and Denoising Algorithm Flow Diagram

Step 4: To compute initial values (X_0 , Y_0 , Z_0 , H_0 , U_0), bitwise operations are applied to the pixel values of the R, G, and B image channels.

$$\begin{cases} X_0 = \frac{\sum \text{bitand}(I_1, 17)}{17 \times SUM} \\ Y_0 = \frac{\sum \text{bitand}(I_2, 34)}{34 \times SUM} \\ Z_0 = \frac{\sum \text{bitand}(I_3, 68)}{68 \times SUM} \\ H_0 = \frac{\sum \text{bitand}(I_1, 136)}{136 \times SUM} \\ U_0 = \frac{\sum \text{bitand}(I_2, 272)}{272 \times SUM} \end{cases} \quad (6.16)$$

where, I_1 , I_2 , and I_3 represent the Red, Green, and Blue channels of the image, respectively, $\text{itand}(I, M)$ performs a bitwise AND operation between image matrix I and integer M , and SUM denotes the total number of pixels in the image. These initial values are vital for generating chaotic sequences and starting encryption, ensuring the process begins with predefined states from the image content.

Step 5: Solve the Hyperchaotic System

Chaotic sequences $X(t)$, $Y(t)$, $Z(t)$, $H(t)$, and $U(t)$ are derived from solving the BPF memristive circuit with initial values from pixel bitwise operations. The first 3001 points are discarded for randomness. X and Y control DNA coding and random matrix R , Z specifies DNA operations, and H determines decoding, ensuring secure transformations for each image block.

Step 6: DNA Encoding and Operations

Image blocks (R, G, B) are DNA-encoded with sequences X and Y , undergo DNA operations from Z , and are diffused for randomness. Decoding with sequence H and merging creates encrypted channels, ensuring robust security. This process ensures secure and randomized encryption.

$$Q_i = \text{DNA_op}(\text{DNA_encode}(I_i, X), \text{DNA_encode}(R, Y), Z) \quad (6.17)$$

Decode using H and merge blocks to form encrypted channels:

$$Q_{final} = \text{DNA_decode}(Q, H) \quad (6.18)$$

Step 7: Anti-Clipping Operation

The Anti-Clipping operation calculates average gray values of G and B channels as keys, generates and sorts chaotic sequences with the Sine map for permutation indices, and permutes the rows and columns of the encrypted R , G , and B channels accordingly. Mathematically, the keys are:

$$\begin{cases} Key_G = \frac{1}{n} \sum_{i=1}^n G_i \\ Key_B = \frac{1}{n} \sum_{i=1}^n B_i \end{cases} \quad (6.19)$$

Sorted chaotic sequences can be generated as:

$$\begin{cases} S_i = \sin(\pi(Key_G + Key_B + i)) \\ Perm_i = sorted_indices(S_i) \end{cases} \quad (6.20)$$

Permuted channels are computed as:

$$\begin{cases} R_{perm} = R(Perm_i, Perm_j)_i \\ G_{perm} = G(Perm_i, Perm_j)_i \\ B_{perm} = B(Perm_i, Perm_j)_i \end{cases} \quad (6.21)$$

Step 8: Combine Channels

Concatenating the shifted RGB matrices creates a three-dimensional matrix, which results in the encrypted image.

6.2.3 The decryption and Denoising algorithms

The method improves image security and quality by preserving fidelity through chaotic encryption, DNA coding, and anti-clipping. An OSELM autoencoder enhances denoising for better visual clarity.

6.2.4 The Proposed Image Decryption and Anti-clipping Algorithm's Steps

The decryption algorithm uses chaotic systems and DNA coding, involving image preprocessing, chaotic sequence generation, block-wise DNA coding, and final reconstruction. The process is summarized as follows:

Step 1: Image Preprocessing

First, to separate the encrypted image into RGB channels, load the encrypted image I from a file. Then, split the image into its three color channels: Red (I_R), Green (I_G), and Blue (I_B). Extract each channel's pixel values for further analysis or processing.

$$\begin{cases} I_R = I(:, :, 1), \\ I_G = I(:, :, 2), \\ I_B = I(:, :, 3), \end{cases} \quad (6.22)$$

Step 2: Permutation Using Chaotic Sequences

Sort and permute the columns and rows of rows of I_R , I_G , and I_B using the chaotic sequences p_x and p_y . Swap column i with column $U_y(i)$ and row i with row $U_x(i)$, where U_y and U_x are permutations obtained by sorting ppy and ppx .

Step 3: Initial Setup and Chaotic Sequence Generation

Start with an encrypted image I consisting of RGB channels I_R, I_G, I_B . Using the SineMap method, generate chaotic sequences $ppx(i)$ and $ppy(i)$ to ensure randomness and security.

$$\begin{cases} ppx(i+1) = r_1 \cdot \sin(\pi \cdot ppx(i)) \\ ppy(i+1) = r_1 \cdot \sin(\pi \cdot ppy(i)) \end{cases} \quad (6.23)$$

where, r_1 controls the chaotic behavior, and $ppx(1) = xx0$, $ppy(1) = yy1$, are initial values.

Step 4: Hyperchaotic System Initialization

Initialize the parameters X_0, Y_0, Z_0, H_0, U_0 for the BPF Memristive Chua's circuit hyperchaotic system, as described in section 2, equation (6.9)

$$A = BPF_memristive_output(X_0, Y_0, Z_0, H_0, U_0, r) \quad (6.24)$$

The sequences X, Y, Z, H , and U are derived from A based on the underlying equation (6.24).

Step 5: DNA Coding and Block Processing

DNA coding, which utilizes DNA coding to process image blocks,

$$\begin{cases} X(i) = \text{mod}(X(i) \cdot 10^4, 8) + 1 \\ Y(i) = \text{mod}(Y(i) \cdot 10^4, 8) + 1 \\ Z(i) = \text{mod}(Z(i) \cdot 10^4, 4) \end{cases} \quad (6.25)$$

Block Encryption involves encrypting blocks I_1, I_2 , and I_R using DNA operations.

$$\begin{cases} Q1_R = DNA_{bian}(\text{fenkuai}(t, I_1, i), H(i)) \\ Q2_R = DNA_{yunsuan}(Q1_R, Q1_last_R, Z(i)) \end{cases} \quad (6.26)$$

where, $\text{fenkuai}(t, I_1, i)$ represents a block of size $t \times t$.

Step 6: Anti-Clipping Mechanism

Zero Padding Removal adjusts the dimensions to eliminate the zero padding introduced during encryption.

$$Q_{jiemi} = \text{Remove zero padding from } Q_{jiemi} \quad (6.27)$$

Step 7: Final Image Reconstruction

Reconstruct and save the decrypted image Q_{jiemi} to a specified location, then display the decrypted image.

This decryption algorithm effectively utilizes chaotic sequences and DNA coding techniques to decrypt encrypted images while mitigating clipping artifacts through careful block processing and diffusion mechanisms. Integrating these methods ensures both security and fidelity in image decryption applications.

Table 6.1: Pseudo-code of the Encryption and Anti-clipping Steps

Algorithm: Specific steps of the encryption and anti-clipping algorithm

```

Input: Digital Color Image (S)
Output: Encrypted Image (I)
# Initialization
Read image I and extract RGB channels I1, I2, I3
Display original image and RGB histograms
# Zero Padding
Compute M, N (dimensions of I1)
Set block size t
Zero-pad I1, I2, I3 to make dimensions divisible by t
# Generate Sine Map Chaotic Sequence
Set sine map parameter r1
Compute initial value x0 from I1, I2
Generate chaotic sequence p using sine map iteration
Discard the first 1000 values of p for randomness
# Transform Chaotic Sequence into Random Matrix
Transform p into integers in the range [0, 255]
Reshape p into matrix R of dimensions M x N
# Solve BPF Memristive Circuit Hyper Chaotic System
Compute the number of blocks r
Compute initial values X0, Y0, Z0, H0, U0 from I1, I2, I3
Solve memristive hyperchaotic system to obtain sequences X, Y, Z, H, U
Discard the first 3001 values of each sequence for randomness
# DNA Coding and Block Encryption
Compute DNA coding methods X, Y, operations Z, decoding methods H
Divide the image into blocks and apply DNA encoding and operations
Merge encoded blocks into complete image channels Q_R, Q_G, Q_B
Convert Q_R, Q_G, Q_B to uint8
# Anti-Clipping
Compute average gray values xx0, xx1 from I2, I3
Generate chaotic sequences ppx, ppy using sine map
Sort ppx and ppy to obtain permutation indices Ux, Uy
Permute rows and columns of Q_R, Q_G, Q_B using Ux, Uy
Display the encrypted image and RGB histograms
Encrypted image Q_jiami
END

```

6.2.5 Denoising using OSELM Auto-encoder

The methodology trains an OSELM Autoencoder to denoise images with various noise types. It involves loading images, adding noise, training, and evaluating performance. The OSELM's autoencoder compresses and reconstructs the image for effective noise reduction.

The OSELM autoencoder is trained on 512×512 noisy images from USC-SIPI and Kaggle, covering seven noise types and 76 noise levels across five image datasets (total of 2,660 normalized samples). It uses 100 sigmoid neurons, regularization (0.1–0.5), 100 iterations, and an 80/20 training-testing split to effectively denoise images under diverse noise conditions.

6.3 Result and Discussion

Simulations were conducted on a Windows 10 PC with an Intel® Core™ i7 CPU @ 3.20, 8GB RAM, using MATLAB 2023a. Parameters included the sine map ($r = 1.8$), two-dimensional parameters ($r_1 = 2.8, r_2 = 3.8$), and the five-dimensional memristive hyperchaotic system parameters ($a = 20, b = 21.43, c = 15, d = 0.15, e = 3, g = 0.05, m_1 = 0.1$, and $m_2 = 0.1$). DNA coding and operations were applied to 512x512 color images from the USC-SIPI and Kaggle databases. The OSELM Autoencoder effectively denoised images during decryption, demonstrating the method's effectiveness.

6.3.1 Simulation Results

The simulations were performed under predefined conditions. The initial parameters for the hyperchaotic systems were set as $(x_0, y_0, z_0, w_0, u_0) = (0.1, 0, 0, 0, 0)$ and for sine maps as $(X_0, X_{01}, X_{02}) = (0.1, 0.1, 0.1)$. Figures 6.5–6.11 illustrate the progression of the image processing stages, including original, encrypted, noised, decrypted, and denoised images.

- **Encryption:** The encrypted images appear completely unrecognizable, ensuring confidentiality.
- **Noise Addition and Removal:** Noise was applied across different types, and noise removal was shown to be effective.
- **Decryption:** Using reverse encryption, anti-clipping, and the OSELM Autoencoder, the decrypted images closely resembled the original images.

The simulations were conducted on a Windows 10 PC with an Intel® Core™ i7 CPU @ 3.20 GHz, 8 GB RAM, using MATLAB 2023a. The images used were sourced from the USC-SIPI image database and the Kaggle dataset, with a resolution of 512x512 pixels.

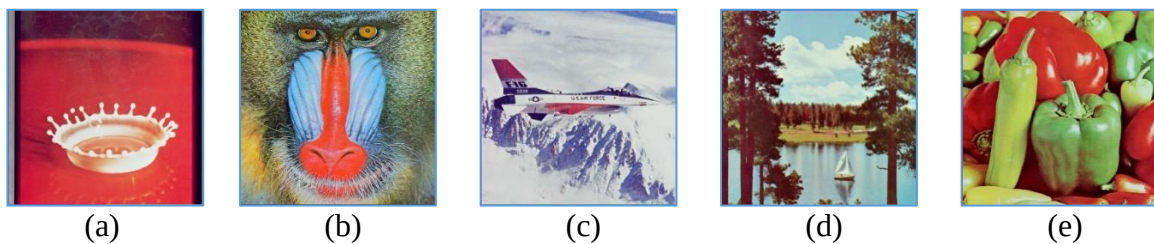


Figure 6.5: Original Input Images. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper

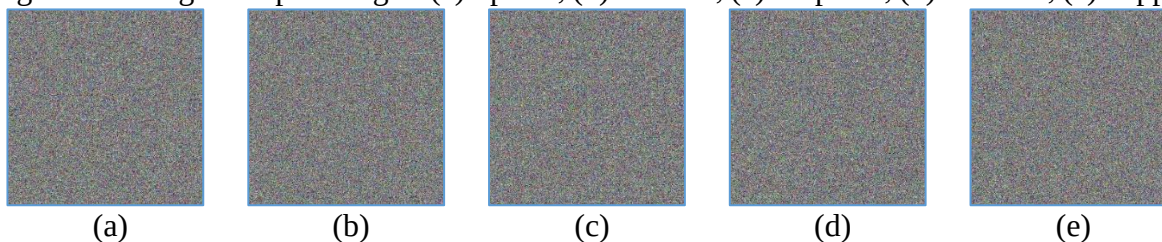
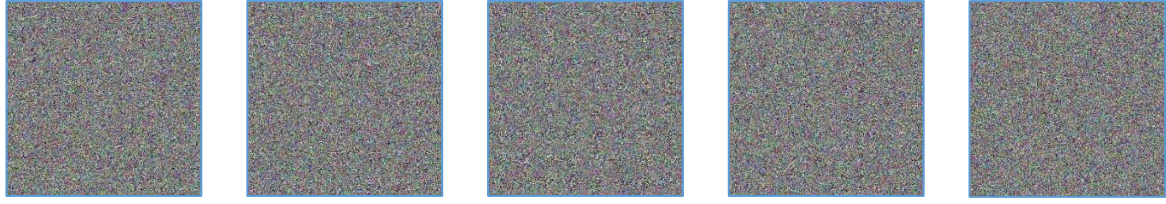


Figure 6.6: Encrypted Images. (a) Encrypted Image of Splash, (b) Encrypted Image of Baboon, (c) Encrypted Image of Airplane, (d) Encrypted Image of Sailboat, (e) Encrypted Image of Pepper



(a) (b) (c) (d) (e)

Figure 6.7: Encrypted Images Corrupted by Gaussian Noise with a Variance of $\sigma^2 = 0.5$. (a) Encrypted-Noised of Splash, (b) Encrypted-Noised of Baboon, (c) Encrypted-Noised of Airplane, (d) Encrypted-Noised of Sailboat, (e) Encrypted-Noised of Pepper



(a) (b) (c) (d) (e)

Figure 6.8: Decrypted Images Corrupted by **Gaussian** Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Noised of Splash, (b) Decrypted-Noised of Baboon, (c) Decrypted-Noised of Airplane, (d) Decrypted-Noised of Sailboat, (e) Decrypted-Noised of Pepper



(a) (b) (c) (d) (e)

Figure 6.9: Decrypted Images Corrupted by **Sensor** Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Noised of Splash, (b) Decrypted-Noised of Baboon, (c) Decrypted-Noised of Airplane, (d) Decrypted-Noised of Sailboat, (e) Decrypted-Noised of Pepper



(a) (b) (c) (d) (e)

Figure 6.10: Decrypted Images Corrupted by **Environmental** Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Noised of Splash, (b) Decrypted-Noised of Baboon, (c) Decrypted-Noised of Airplane, (d) Decrypted-Noised of Sailboat, (e) Decrypted-Noised of Pepper



(a) (b) (c) (d) (e)

Figure 6.11: Decrypted-Nenoised from Gaussian Noise with a Variance of $\sigma^2 = 0.5$. (a) Decrypted-Denoised of Splash, (b) Decrypted-Denoised of Baboon, (c) Decrypted-Denoised of Airplane, (d) Decrypted-Denoised of Sailboat, (e) Decrypted-Denoised of Pepper

6.3.2 Image Security Analysis

6.3.2.1 Keyspace Analysis

The key space must be large enough to resist brute-force attacks, with a size greater than 2^{100} generally considered secure (S. Yan et al., 2023). As per (S. Zhang & Liu, 2021), with a computer precision of 10^{-16} , the key components in our encryption system include the 1D sine map iteration value (r) and its initial value (X_0), the 2D sine map iteration values (r_1 and r_2) and their initial values (X_{01} and X_{02}), the initial values of the Memristive hyperchaotic system (x_0, y_0, z_0, w_0, u_0), zero-padded values (m_1 and n_1), and the average gray values of the G and B channels (xx_0, xx_1). The 15 variables create a key space larger than $10^{16 \times 15} = 10^{240} \approx 2^{797}$, far exceeding the 2^{100} threshold. This ensures the encryption method is highly resistant to brute-force attacks. Table 6.2 compares key spaces, showing the proposed method's superior strength.

Table 6.2: Comparative of Keyspace

Method	Keyspace value
Minimum	2^{100}
Proposed	2^{797}
Ref. (S. Zhang & Liu, 2021)	2^{279}
Ref. (Ahuja & Doriya, 2023)	2^{398}
Ref. (Ibrahim et al., 2022)	2^{127}
Ref. (Alexan et al., 2022)	2^{425}

With three 797-bit sub-keys, an attacker would need 3×2^{797} attempts (Elsaid & Alotaibi, 2023). At 3×2^{797} instructions per second, brute-forcing the key would take about 7.929×10^{223} years.

$$\frac{3 \times 2^{797}}{1000 \times 10^6 \times 60 \times 60 \times 24 \times 365} > 7.929 \times 10^{223} \text{ years} \quad (6.28)$$

6.3.2.2 Key Sensitivity Analysis

Encryption algorithms must be sensitive to small key changes. Figure 6.12 shows that altering r by 1.8 to 1.8×10^{-16} during decryption significantly changed the output, highlighting the algorithm's security and robustness against minor key variations.

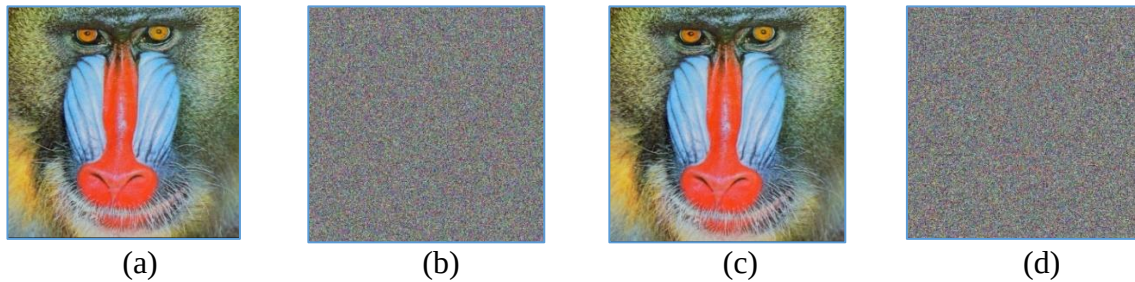


Figure 6.12: Encryption-Decryption Results with and without Slight Key Changes;
(a) Original Image, (b) Encrypted Image, (c) Recovered Image, (d) Recovered with a Slightly Different Key

6.3.2.3 Information entropy analysis

Information entropy serves as a criterion for assessing the randomness present in an image, calculated by

$$H(x) = - \sum_{i=0}^{2^N-1} S(x_i) \log_2 S(x_i) \quad (6.29)$$

$H(x)$ represents image entropy, where N is the number of bits and x_i are pixel values (0–255). $P(x_i)$ is the probability of each x_i . For 8-bit images, entropy ranges from 0 to 8, with values near 8 indicating uniform pixel distribution. Table 6.3 compares entropy for plain and encrypted images.

Table 6.3: Entropy of Original and Cyphered Images

Images		Red	Green	Blue
Splash	Original images	6.9481	6.8845	6.1265
	Encrypted images	7.9993	7.9994	7.9994
Baboon	Original images	7.7067	7.4744	7.7522
	Encrypted images	7.9994	7.9994	7.9994
Airplane	Original images	6.7178	6.799	6.2138
	Encrypted images	7.9993	7.9994	7.9993
Sailboat	Original images	7.3124	7.6429	7.2136
	Encrypted images	7.9994	7.9994	7.9993
Pepper	Original images	7.3388	7.4963	7.0583
	Encrypted images	7.9994	7.9993	7.9993

Table 6.4 compares the entropy values of our approach and similar research, demonstrating the effectiveness of our proposed cryptosystem.

Table 6.4: Comparative Entropy Results of Test Images Using Different Techniques

Methods	Entropy value of Encrypted Baboon		
	Red	Green	Blue
Maximum	8.0000	8.0000	8.0000
Proposed	7.9994	7.9994	7.9994
Ref. (T. Zhang & Wang, 2023)	7.99914	7.99926	7.99929
Ref. (Z. Liang et al., 2023a)	7.9993	7.9992	7.9992
Ref. (Mfungo & Fu, 2023)	7.9964	7.9984	7.9961

6.3.2.4 Correlation Analysis

Five thousand neighboring pixel pairs from original and encrypted images were analyzed to assess vulnerability. Effective encryption reduces pixel correlation to resist differential attacks (X. Chen et al., 2023; Ibrahim et al., 2022). Figure 6.13 shows the RGB correlation test results for the original Baboon image, with the correlation coefficient r_{mn} computed using equation (6.15).

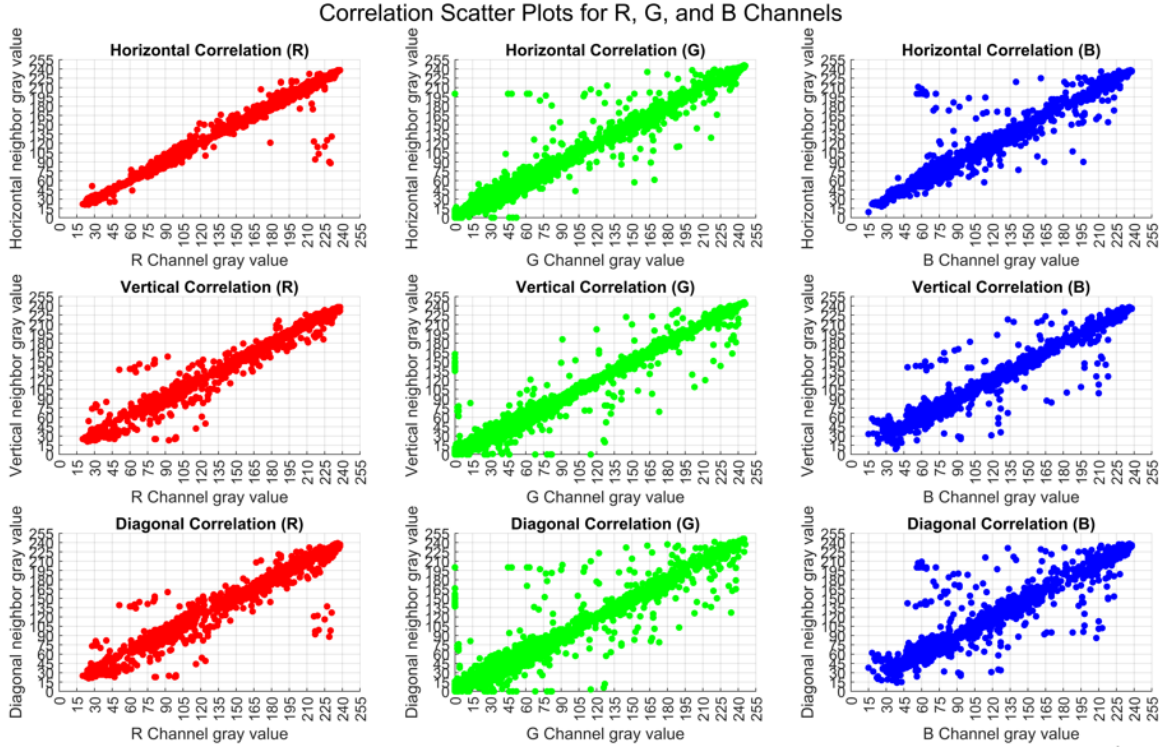


Figure 6.13: Baboon's Original Test Image's Horizontal, Vertical, and Diagonal Correlations

Table 6.5: Correlation Coefficients Between Original and Encrypted Version

Image	Correlation Directions	Pixel correlation coefficients					
		Original images			Encrypted images		
		Red	Green	Blue	Red	Green	Blue
Splash	Horizontal (H)	0.99341	0.97958	0.98128	0.00058	0.02647	0.00563
	Vertical (V)	0.99451	0.98788	0.97917	0.01227	0.00290	0.02093
	Diagonal (D)	0.98876	0.97052	0.96522	0.02404	-0.0162	-0.0042
Baboon	Horizontal (H)	0.92586	0.868	0.91053	-0.0364	-0.0005	0.00730
	Vertical (V)	0.87145	0.77742	0.88889	-0.01302	-0.0074	0.01463
	Diagonal (D)	0.86042	0.74827	0.85018	0.00664	0.00282	-0.0246
Airplane	Horizontal (H)	0.9702	0.95435	0.95958	0.00790	0.00445	-0.0332
	Vertical (V)	0.95293	0.96382	0.92647	-0.0206	-0.0046	0.00879
	Diagonal (D)	0.92752	0.92373	0.90269	-0.0197	0.01554	0.00040
Sailboat	Horizontal (H)	0.95517	0.97018	0.9705	0.01215	0.00025	-0.0126
	Vertical (V)	0.95061	0.96422	0.96763	0.00070	0.00238	0.03111
	Diagonal (D)	0.93671	0.94932	0.94990	-0.03175	-0.0193	-0.0004
Pepper	Horizontal (H)	0.96763	0.98116	0.96704	0.00375	0.02556	0.00689
	Vertical (V)	0.96621	0.97931	0.96673	0.01625	0.00269	-0.0106
	Diagonal (D)	0.95845	0.96757	0.95029	0.00235	0.01044	-0.0107

$$r_{xy} = \frac{cov(m, n)}{\sqrt{D(m)D(n)}} \quad (6.30)$$

$$cov(m, n) = \frac{1}{N} \sum_{i=1}^N (m_i - E(m))(n_i - E(n)) \quad (6.31)$$

$$zE(m) = \frac{1}{N} \sum_{i=1}^N m_i \quad (6.32)$$

$$D(m) = \frac{1}{N} \sum_{i=1}^N (m_i - E(m))^2 \quad (6.33)$$

where, $cov(m, n)$ denotes the covariance between m and n , $D(m)$ and $D(n)$ represent the variances of m and n , respectively.

Table 6.5 and Figure 6.12 show that neighboring pixels in the encrypted image lack discernible correlation, unlike the higher correlation in the unencrypted image. This indicates effective encryption and security provided by the hyperchaotic system. Table 6.6 compares correlation results with various encryption techniques, highlighting the robustness against attacks.

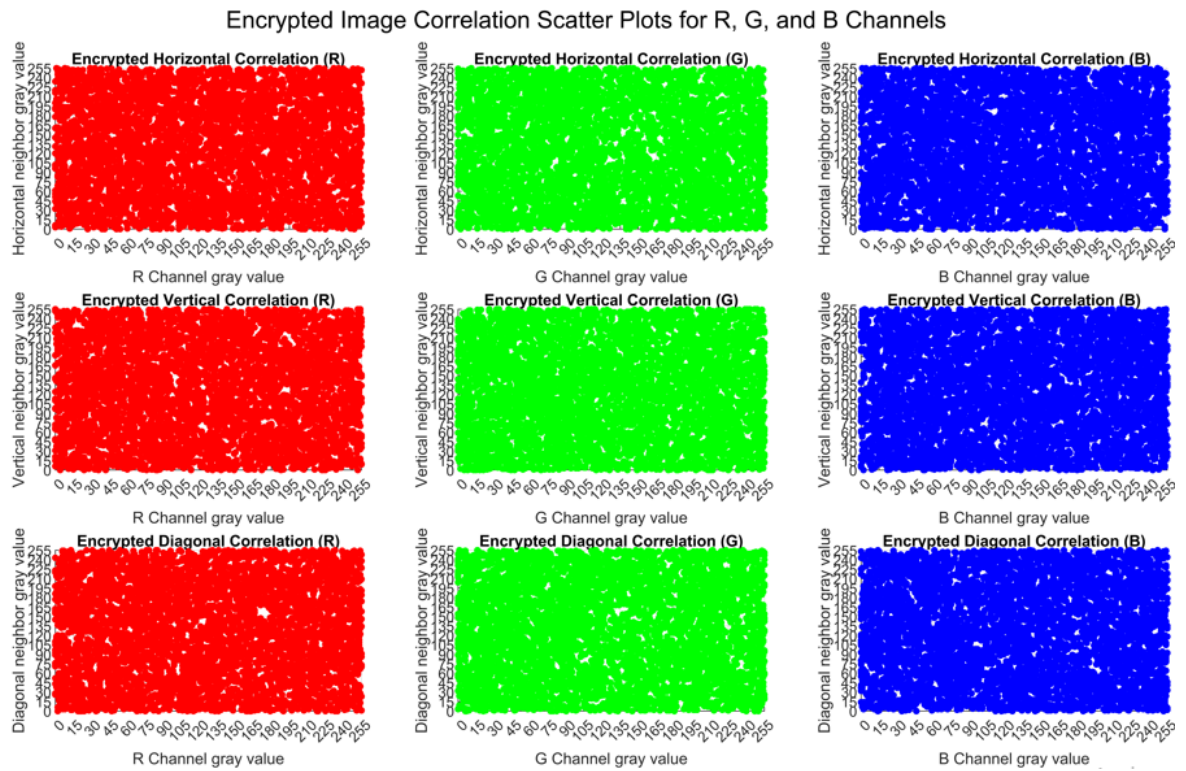


Figure 6.14: Baboon's Encrypted Test Image's Horizontal, Vertical, and Diagonal Correlations

Table 6.6: Correlation Coefficients for Encrypted Images Using Different Techniques

Channels Directions	Red			Green			Blue			Averag e
	H	V	D	H	V	D	H	V	D	
Original Baboon image	0.92586	0.87145	0.86042	0.868	0.7774	0.7483	0.9105	0.8889	0.85018	0.85567
Encrypted Ours	-0.03645	0.01302	0.00664	-0.00047	-0.0074	0.00282	0.00730	0.01463	-0.02458	-0.00561
Ref. (Ahuja & Doriya, 2023)	-0.0038	-0.0226	-0.0015	-0.0038	-0.0226	-0.0015	-0.0038	-0.0226	-0.0015	-0.00930
Ref. (Ibrahim et al., 2022)	0.002	0.0054	0.0054	0.0037	0.0037	0.0022	-0.0042	-0.0049	0.002	0.00349
Ref. (M. A. Lone & Qureshi, 2023)	0.0031	0.0031	0.0031	0.0036	0.0036	0.0036	0.0008	0.0008	0.0008	0.00250
Ref. (Alexan et al., 2022)	0.00228 7	-0.0016	-0.00132	0.002287	-0.0016	-0.0013	0.00229	-0.0016	-0.00132	0.00228 7

6.3.2.5 Histogram analysis

Histograms are essential for evaluating encrypted image quality and showing pixel intensity distribution. Original images often have predictable patterns with spikes, while encrypted images should display a uniform distribution, indicating resilience against statistical attacks (Alexan, Gabr, et al., 2023). Figures 6.15 and 6.16 show the original images and their RGB histograms, while Figures 6.17 and 6.18 display the encrypted images and their RGB histograms. The histograms reveal that encrypted images have a uniform distribution, unlike the irregular spikes in original images, demonstrating the effectiveness of the encryption method in concealing image redundancy (Alexan et al., 2022).

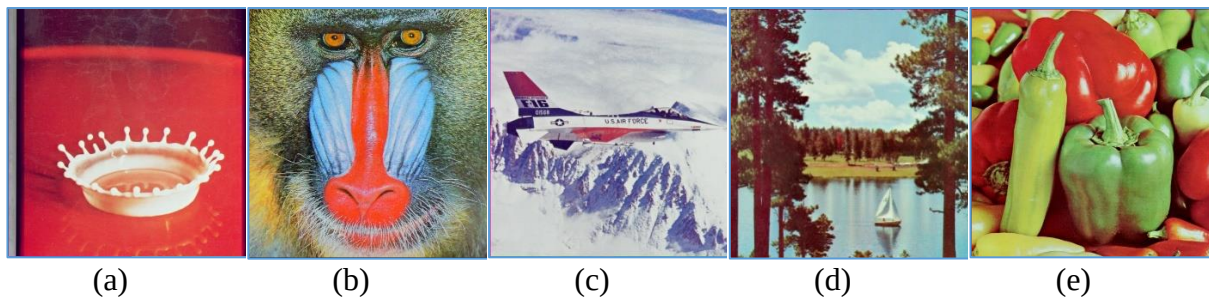


Figure 6.15: Original Input Images. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper

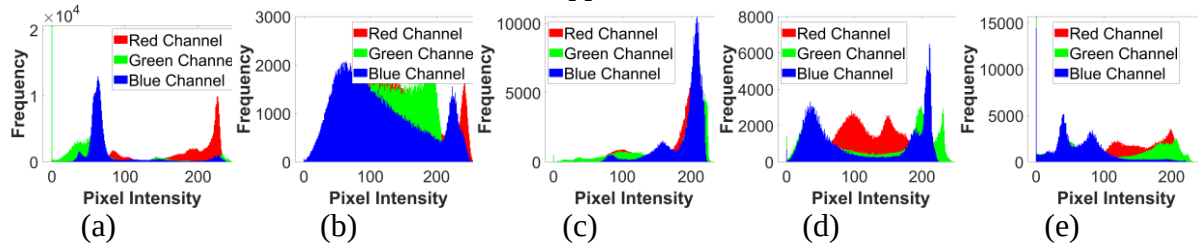


Figure 6.16: Plane Images RGB Histogram. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper

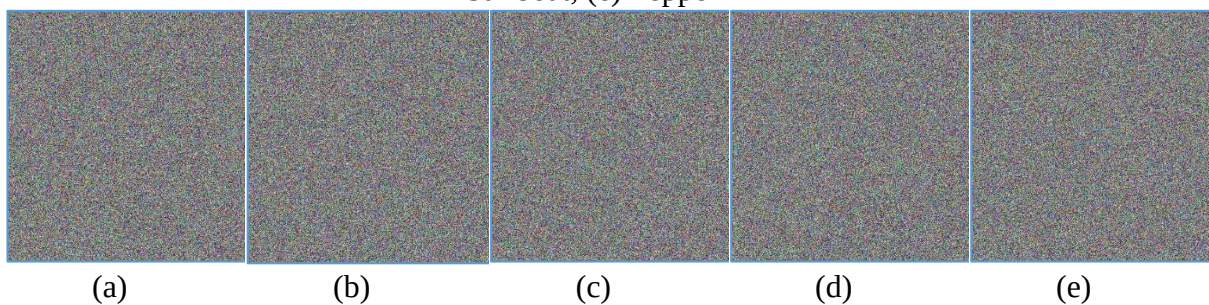


Figure 6.17: Encrypted Images. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper

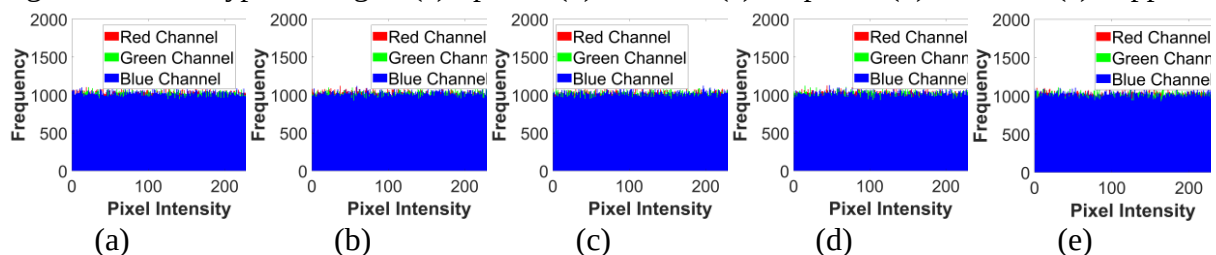


Figure 6.18: Encrypted Images RGB Histogram. (a) Splash, (b) Baboon, (c) Airplane, (d) Sailboat, (e) Pepper

6.3.2.6 Differential Attack Analysis

Differential attack analysis assesses how minor changes in the input image affect the encrypted output (Z. Liang et al., 2023a). Even a single-pixel change can cause significant variation in the encrypted image. NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) are commonly used metrics for this analysis (Z. Liang et al., 2023a). These metrics are defined as follows.

$$NPCR = \frac{1}{L \times M} \sum_{l=1}^L \sum_{m=1}^M D(l, m) \times 100\% \quad (6.34)$$

$$D(l, m) = \begin{cases} 1 & \text{if } S_1(l, m) \neq S_2(l, m) \\ 0 & \text{else} \end{cases} \quad (6.35)$$

$$UACI = \frac{1}{L \times M} \sum_{l=1}^L \sum_{m=1}^M \frac{|S_1(l, m) - S_2(l, m)|}{255} \times 100\% \quad (6.36)$$

where, $S_1(l, m)$, and $S_2(l, m)$ represent the pixel values of the encrypted images before and after a single pixel in the original image has been altered.

Table 6.7: The NPCR and UACI Results of Input Images

Pictures	NPCR (%)				UACI (%)			
	Red	Green	Blue	Average	Red	Green	Blue	Average
Splash	99.625	99.6044	99.6044	99.6113	34.1926	35.6666	31.8916	33.917
Baboon	99.625	99.5975	99.6014	99.608	30.195	31.659	32.253	31.369
Airplane	99.6063	99.6258	99.5991	99.6104	32.0117	33.1105	32.6373	32.587
Sailboat	99.5983	99.6101	99.6002	99.6029	27.951	34.3675	34.3997	32.2394
Pepper	99.6323	99.6262	99.6159	99.6248	29.0509	33.9288	33.8063	32.262

NPCR and UACI values depend on the source image properties. Effective encryption typically achieves NPCR values over 99.00% and UACI around 33% (Alexan, Gabr, et al., 2023). Table 6.7 shows the NPCR and UACI results for the original and encrypted images. It demonstrates the proposed technique's effectiveness against differential attacks, with NPCR values between 99.587% and 99.634% and UACI values ranging from 31.369% to 33.917%. These results indicate strong performance and robust protection against differential attack analysis.

Table 6.8: NPCR and UACI Results for Ciphered Baboon Test Image

Algorithm	NPCR				UACI			
	Red	Green	Blue	Average	Red	Green	Blue	Average
Proposed	99.595	99.610	99.617	99.607	30.195	31.659	32.253	31.369
Ref. (Ibrahim et al., 2022)	99.475	99.685	99.485	99.548	32.895	33.102	32.805	32.934
Ref. (Shakir et al., 2022)	99.6124	99.6124	99.6124	99.6124	29.2827	29.2827	29.2827	29.2827
Ref. (H. Wang et al., 2023)	99.6090	99.5987	99.5972	99.6016	33.3996	33.5134	33.4727	33.4619

Table 6.8 compares our method with other approaches, highlighting superior performance in managing subtle image variations and enhanced attack resilience.

6.3.2.7 Statistical Attack Analysis

Statistical analysis, including histogram and correlation tests, shows the algorithm's resilience. Uniform pixel distribution in ciphertext hides the original image, while near-zero correlation coefficients indicate strong security (Mfungo & Fu, 2023). A higher Mean Squared Error (MSE) reflects better encryption performance and resilience against statistical attacks (Alexan, Gabr, et al., 2023). MSE is calculated by squaring the pixel-wise differences between the encrypted $E(l,m)$ and original images $S(l,m)$, averaging over all pixel dimensions $L \times M$. MSE is illustrated in equation (6.22).

$$MSE = \frac{1}{L \times M} \sum_{l=1}^L \sum_{m=1}^M [S(l, m) - E(l, m)]^2 \quad (6.37)$$

6.3.2.8 Peak Signal-to-Noise Ratio and Structural Similarity Index Measure

The peak signal-to-noise ratio (PSNR) is used to assess encryption effectiveness and is defined as the MSE. The formula is:

$$PSNR = 10 \log \left[\frac{P_{max}^2}{MSE} \right] \quad (6.38)$$

Table 6.9: MSE and PSNR Values of Original and Encrypted Images without Noise

Image	Channel	MSE (dB)	PSNR (dB)
Splash	Red	11402.1501	7.56
	Green	12364.2575	7.21
	Blue	9899.5653	8.17
	Average	11221.99	7.63
Baboon	Red	8672.6646	8.75
	Green	7718.1028	9.26
	Blue	9465.5451	8.37
	Average	8618.77	8.78
Airplane	Red	9964.2553	8.15
	Green	10691.0189	7.84
	Blue	10396.3156	7.96
	Average	10350.53	7.98
Sailboat	Red	7318.9014	9.49
	Green	11509.2684	7.52
	Blue	11533.1821	7.51
	Average	10120.45	8.08
Pepper	Red	11402.1501	7.56
	Green	12364.2575	7.21
	Blue	9899.5653	8.17
	Average	11221.99	7.63

The PSNR is calculated from the maximum pixel value (255) and the MSE. Lower PSNR values between original and cipher images, compared to those between original and reconstructed images, indicate strong encryption resistance. Table 6.9 presents the MSE and PSNR values for RGB channels of original and encrypted images. Table 6.10 shows the MSE and PSNR values for various encrypted images, comparing the results with other methods in the literature. The Structural Similarity Index Measure (SSIM) evaluates image quality, with an infinite PSNR and SSIM value of one indicating perfect fidelity between the original and decrypted images (Ahuja & Doriya, 2023). Table 6.11 presents the PSNR and SSIM values for the fully decrypted test images.

Table 6.10: Comparative MSE and PSNR Values of RGB Channels

Different Methods	MSE				PSNR (dB)			
	Red	Green	Blue	Average	Red	Green	Blue	Average
Proposed	10681.16	9025.59	7098.02	8934.92	7.84	8.58	9.62	8.68
Ref. (Ahuja & Doriya, 2023)	-	-	-	-	18.66	18.66	18.66	18.66
Ref. (M. A. Lone & Qureshi, 2023)	9004.40	9004.40	9004.40	9004.40	8.6454	8.6454	8.6454	8.6454
Ref. (Alexan et al., 2022)	10,663.3	8982.02	7021.33	8888.88	7.8518	8.5970	9.6666	8.7051
Ref. (Shakir et al., 2022)	-	-	-	8914.025	-	-	-	3.891629

Table 6.11: PSNR and SSIM Values for Original and Reconstructed Images

Image	PSNR	Maximum	SSIM
Splash	∞	1	1
Baboon	∞	1	1
Airplane	∞	1	1
Sailboat	∞	1	1
Pepper	∞	1	1

6.3.3 Image Denoising Analysis

This study introduces an image encryption algorithm using an OSELM autoencoder, DNA coding, and a memristor-based hyperchaotic system, effectively removing Gaussian, salt-and-pepper, and speckle noise. Without the OSELM autoencoder, noise (0.01 to 0.20 variance) was added, and the image core remained recognizable up to 0.04 variance. Figures 6.19-6.21 show decrypted images with 0.01-0.04 noise, while Tables 6.13 - 6.15 assess noise in decrypted images. Using the OSELM autoencoder after decryption enhances noise resistance. Performance is evaluated using PSNR, SSIM, MSE, and MAE metrics.

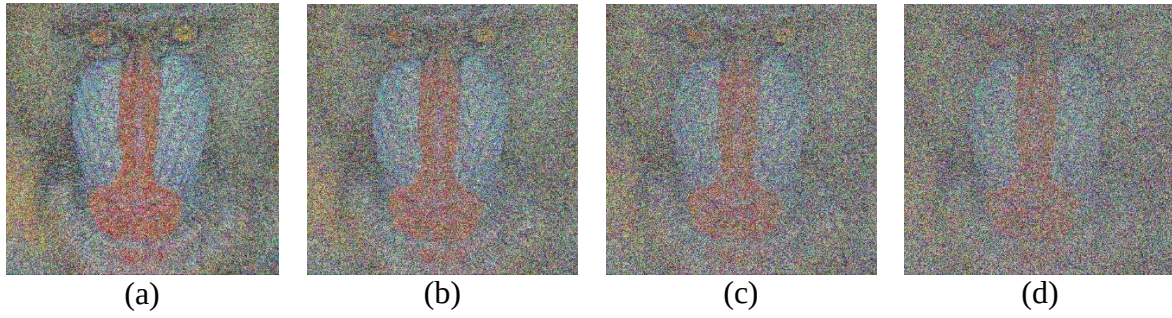


Figure 6.19: Decrypted Images with Gaussian Noise Attacks: (a) 0.01 Noise Level, (b) 0.02 Noise Level, (c) 0.03 Noise Level, (d) 0.04 Noise Level.

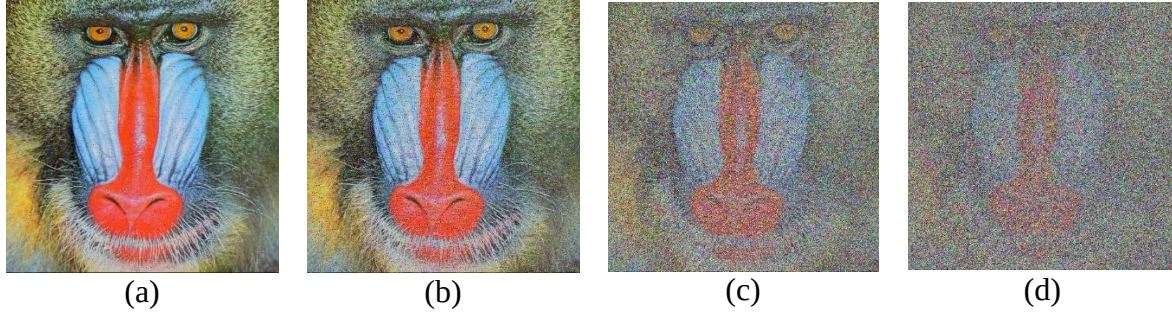


Figure 6.20: Decrypted Images with Salt-and-Pepper Noise Attacks: (a) 0.01 Noise Level, (b) 0.04 Noise Level, (c) 0.1 Noise Level, (d) 0.14 Noise Level.

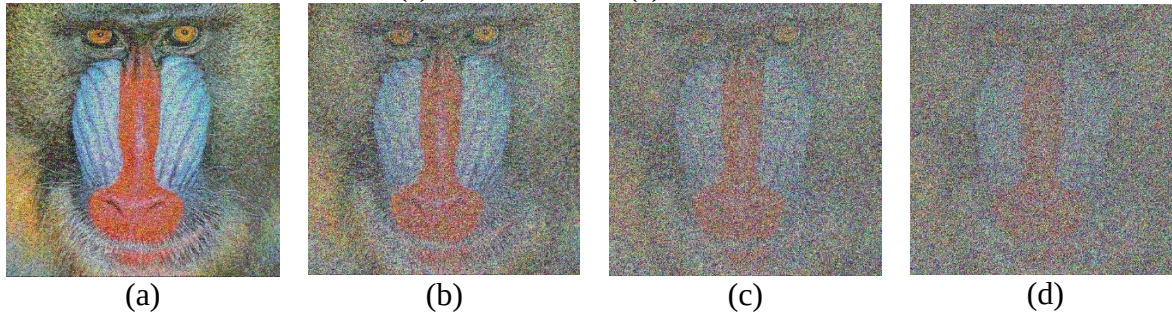
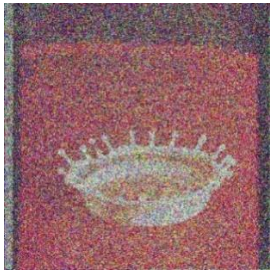


Figure 6.21: Decrypted Images with Speckle Noise Attacks: (a) 0.01 Noise Level, (b) 0.03 Noise Level, (c) 0.05 Noise Level, (d) 0.07 Noise Level.

Table 6.12: Assessment of Gaussian Noise Qualification in Decrypted Images

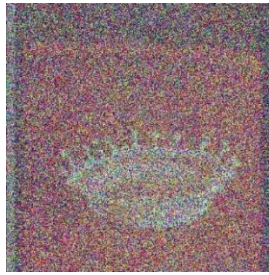
Gaussian Noise Variance	MSE			PSNR		
	MSE_R	MSE_G	MSE_B	PSNR_R	PSNR_G	PSNR_B
0	4.8804e+03	4.4276e+03	5.3824e+03	11.2463	11.6691	10.8210
0.0200	6.7204e+03	6.1028e+03	7.3838e+03	9.8568	10.2755	9.4480
0.0400	7.5530e+03	6.7998e+03	8.2467e+03	9.3496	9.8058	8.9680
0.0600	8.0899e+03	7.2729e+03	8.8066e+03	9.0514	9.5137	8.6827
0.0800	8.3812e+03	7.5361e+03	9.2138e+03	8.8977	9.3593	8.4864
0.1000	8.5288e+03	7.6668e+03	9.3866e+03	8.8219	9.2847	8.4057
0.1200	8.6078e+03	7.7002e+03	9.4322e+03	8.7819	9.2658	8.3847
0.1400	8.6106e+03	7.7276e+03	9.4481e+03	8.7805	9.2504	8.3773
0.1600	8.6084e+03	7.7230e+03	9.4575e+03	8.7816	9.2529	8.3731
0.1800	8.6105e+03	7.7199e+03	9.4584e+03	8.7805	9.2547	8.3726
0.2000	8.6128e+03	7.7197e+03	9.4577e+03	8.7793	9.2548	8.3730

Gaussian Noise (0.75)



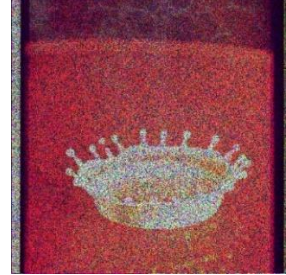
(a) Splash

Salt-and-Pepper Noise (0.75)

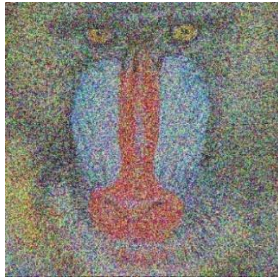


(b) Splash

Speckle Noise (0.75)



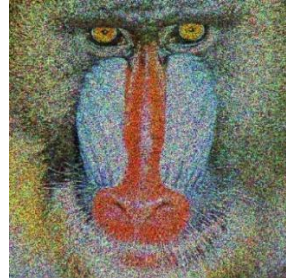
(c) Splash



(d) Baboon



(e) Baboon



(f) Baboon



(g) Airplane



(h) Airplane



(i) Airplane



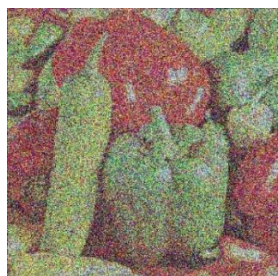
(j) Sailboat



(k) Sailboat



(l) Sailboat



(m) Pepper



(n) Pepper



(o) Pepper

Figure 6.22: Decrypted Noised Images

To enhance anti-noise capability, we integrate an OSELM autoencoder after decryption. Seven noise types were added during transmission, and noisy images were paired with clean versions for training. Using 40 hidden neurons and 10,000 iterations, the network effectively denoised images, as seen in Figure 6.23, compared to the noisy images in Figure 6.22. Both figures display Gaussian, salt-and-pepper, and speckle noise results across five test images.

Table 6.13: Assessment of Salt-and-Pepper Noise in Decrypted Images

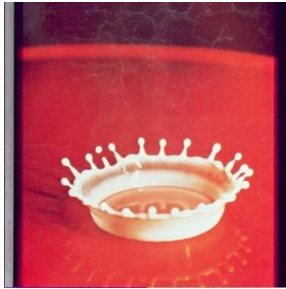
Salt-and-pepper Noise Variance	MSE			PSNR		
	MSE_R	MSE_G	MSE_B	PSNR_R	PSNR_G	PSNR_B
0	0	0	0	65535	65535	65535
0.0200	516.9315	470.0154	542.7389	20.9965	21.4097	20.7849
0.0400	1.5981e+03	1.4286e+03	1.7428e+03	16.0947	16.5818	15.7183
0.0600	3.0509e+03	2.6998e+03	3.3104e+03	13.2865	13.8175	12.9320
0.0800	4.5382e+03	4.0444e+03	4.9541e+03	11.5619	12.0623	11.1812
0.1000	5.9002e+03	5.2761e+03	6.4295e+03	10.4221	10.9077	10.0490
0.1200	6.9476e+03	6.2127e+03	7.5968e+03	9.7125	10.1980	9.3245
0.1400	7.6741e+03	6.8727e+03	8.4000e+03	9.2805	9.7596	8.8880
0.1600	8.1633e+03	7.2934e+03	8.9283e+03	9.0121	9.5015	8.6231
0.1800	8.4136e+03	7.5286e+03	9.1879e+03	8.8810	9.3637	8.4986
0.2000	8.5465e+03	7.6704e+03	9.3498e+03	8.8129	9.2826	8.4228

Table 6.14: Assessment of Speckle Noise Qualification in Decrypted Images

Speckle Noise Variance	MSE			PSNR		
	MSE_R	MSE_G	MSE_B	PSNR_R	PSNR_G	PSNR_B
0	0	0	0	65535	65535	65535
0.0200	4.5493e+03	4.1515e+03	5.0460e+03	11.5513	11.9488	11.1013
0.0400	6.4486e+03	5.8321e+03	7.1083e+03	10.0362	10.4725	9.6131
0.0600	7.3735e+03	6.6699e+03	8.0643e+03	9.4541	9.8896	9.0651
0.0800	7.8419e+03	7.1013e+03	8.5629e+03	9.1866	9.6174	8.8046
0.1000	8.1402e+03	7.3546e+03	8.8943e+03	9.0244	9.4652	8.6397
0.1200	8.3164e+03	7.4717e+03	9.0776e+03	8.9314	9.3966	8.5511
0.1400	8.4323e+03	7.5679e+03	9.2053e+03	8.8713	9.3411	8.4904
0.1600	8.5103e+03	7.6242e+03	9.3141e+03	8.8313	9.3089	8.4394
0.1800	8.5369e+03	7.6789e+03	9.3395e+03	8.8178	9.2778	8.4276
0.2000	8.5668e+03	7.7141e+03	9.4107e+03	8.8026	9.2579	8.3946

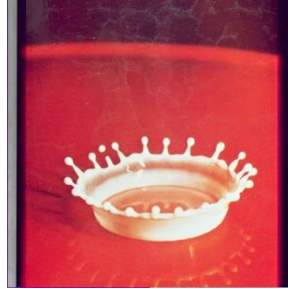
Table 6.15 analyzes image denoising using PSNR, SSIM, MSE, and MAE across three noise types and five test images. Figures 6.22 and 6.23 compare decrypted-noisy, and denoised images, showing that the OSELM autoencoder reduces noise and improves image quality.

Gaussian Noise



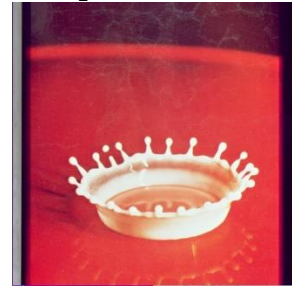
(a) Splash

Salt-and-Pepper Noise

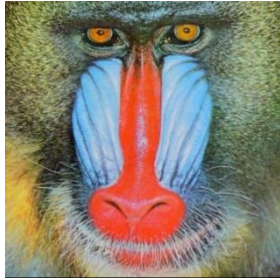


(b) Splash

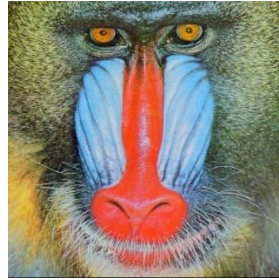
Speckle Noise



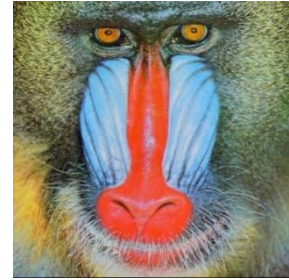
(c) Splash



(d) Baboon



(e) Baboon



(f) Baboon



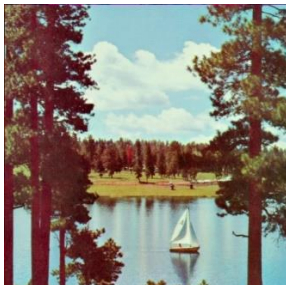
(g) Airplane



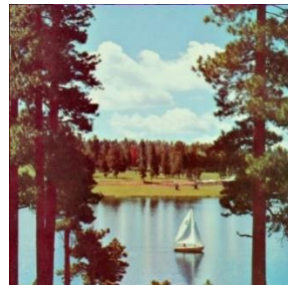
(h) Airplane



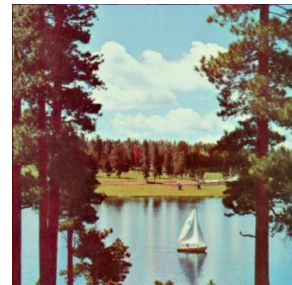
(i) Airplane



(j) Sailboat



(k) Sailboat



(l) Sailboat



(m) Pepper



(n) Pepper



(o) Pepper

Figure 6.23: Denoised Images

Table 6.15: Performance Metrics for Noisy and Denoised Images

Pictures	Noisy					Denoised			
	Noise type	PSNR	SSIM	MSE	MAE	PSNR	SSIM	MSE	MAE
Splash	Gaussian	7.174	0.163	12465.10	89.89	35.63	0.9989	17.77	3.496
	Salt and pepper	5.955	0.086	16504.04	96.43	35.63	0.9989	17.77	3.496
	Speckle	10.143	0.375	6291.17	58.99	35.63	0.9989	17.77	3.499
Baboon	Gaussian	7.332	0.106	12017.94	94.10	Inf	1	0	0
	Salt and pepper	6.493	0.057	14581.31	95.73	Inf	1	0	0
	Speckle	9.152	0.224	7904.87	72.83	Inf	1	0	0
Airplane	Gaussian	7.486	0.036	11600.85	88.43	23.63	0.9936	282.12	16.324
	Salt and pepper	6.108	0.019	15930.69	99.69	23.63	0.9936	282.12	16.324
	Speckle	7.566	0.055	11388.10	86.93	23.63	0.9936	281.62	16.310
Sailboat	Gaussian	7.247	0.059	12256.54	91.63	37.45	0.9994	11.70	2.991
	Salt and pepper	6.167	0.031	15715.68	96.54	37.45	0.9994	11.70	2.991
	Speckle	9.359	0.210	7537.34	67.85	37.45	0.9994	11.70	2.991
Pepper	Gaussian	7.283	0.138	12156.48	91.15	28.31	0.9947	95.95	8.402
	Salt and pepper	6.170	0.073	15707.49	97.70	28.31	0.9947	95.95	8.402
	Speckle	9.753	0.350	6882.20	64.97	28.31	0.9947	95.95	8.402

6.3.3.1 Peak signal-to-noise ratio

Peak signal-to-noise ratio (PSNR) measures the quality of reconstructed or denoised images by comparing the original signal's maximum power to the noise affecting it. Higher PSNR values indicate better image quality (Mahdaoui et al., 2022). It can be expressed as:

$$PSNR = 10 \cdot \log_{10} \left[\frac{MAX^2}{MSE} \right] \quad (6.39)$$

where, MAX is the maximum possible pixel value of the image and MSE is the Mean Squared Error between the original and denoised images.

6.3.3.2 Structural similarity index measure

The Structural Similarity Index Measure (SSIM) evaluates image similarity by considering structural details, luminance, and contrast, offering a more human-aligned assessment than MSE (Mahdaoui et al., 2022). The formula for SSIM is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (6.40)$$

where, μ_x and μ_y are the mean intensities, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the covariance, and C_1 and C_2 are constants to stabilize the division when the denominator is close to zero.

6.3.3.3 Mean Squared Error

The Mean Squared Error (MSE) measures image distortion by calculating the difference between estimated and actual pixel values. Higher MSE values indicate more distortion, while lower values suggest less (Mahdaoui et al., 2022). MSE is defined as follows:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \quad (6.41)$$

where, $I(i,j)$ is the pixel value of the original image, $K(i,j)$ is the pixel value of the denoised image, and m and n are the dimensions of the image.

6.3.3.4 Mean absolute error

Mean Absolute Error (MAE) gauges the average magnitude of absolute differences between original and denoised images, offering a more straightforward and direct error measure than MSE. (Bakour et al., 2024).

$$MAE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i,j) - K(i,j)| \quad (6.42)$$

where, $I(i,j)$ is the pixel value of the original image, $K(i,j)$ is the pixel value of the denoised image, and m and n are the dimensions of the image.

6.3.3.5 Running Speed Analysis

Encryption, decryption, and denoising efficiency depend on hardware, software, and programming. Despite extra DNA coding steps, the system remains secure and efficient, as shown in Table 6.16. Tested on a Windows 10 PC (Intel i7, 8GB RAM), the method is slower than others in Table 6.17. Improving software and hardware could enhance performance.

Table 6.16: The Encryption and Decryption Running Speed of Various Test Images

Test Images	Dimension	Size	Encryption time (sec.)	Decryption time (sec.)	Denoising time except training (sec.)
Splash	(512x512)	509 KB	1.489592	1.632584	0.54
Baboon	(512x512)	739 KB	1.619922	1.961664	0.54
Airplane	(512x512)	536 KB	1.543100.	1.625397	0.54
Sailboat	(512x512)	673 KB	1.470869	1.627494	0.54
Pepper	(512x512)	642 KB	1.444870	1.653887	0.54

Table 6.17: Comparison of the Encryption Time With Other Schemes

Schemes and Image sizes	Encryption time (seconds)
Proposed Average (512×512)	1.506313
Ref. (P. N. Lone et al., 2022) (256×256)	7.1783
Ref. (X. Liang et al., 2023) (512 × 768), and (768 × 1024)	4.50, and 6.63
Ref. (Alexan et al., 2022) (512×512)	4.379808
Ref. (X. Wang, Zhang, et al., 2023) (256×256), and (512×512)	0.5726, and 2.0181

6.4 Randomness Evaluation Using NIST Statistical Test Suite

The NIST SP800-22 Statistical Test Suite (STS) was used to evaluate the randomness of an encrypted image's bitstream. The t-tests assess statistical properties such as uniformity, independence, and entropy to determine if the bit sequence exhibits true randomness (Gedda et al., 1977).

Table 6.18: NIST STS Results for the Encrypted Image Bit Stream

No	Test Name	P-Value	Proportion	Accepted/Rejected	
1	Frequency (Monobit) Test	0.245660	100%	Accepted	
2	Block Frequency Test	0.199939	100%	Accepted	
3	Runs Test	0.109413	100%	Accepted	
4	Longest Run of Ones Test	0.872143	100%	Accepted	
5	Rank Test	0.827634	100%	Accepted	
6	Discrete Fourier Transform (FFT) Test	0.046323	100%	Accepted	
7	Non-Overlapping Template Matching Test Average	0.5173	100%	Accepted	
8	Overlapping Template Matching Test	0.215634	100%	Accepted	
14	Random Excursions Test Average	0.374200	100%	Accepted	
15	Random Excursions Variant Test Average	0.651700	100%	Accepted	
9	Maurer’s Universal Statistical Test	0.564126	100%	Accepted	
10	Linear Complexity Test	0.902673	100%	Accepted	
11	Serial Test	data1	0.557771	100%	Accepted
		data2	0.462419	100%	Accepted
12	Approximate Entropy Test	0.310541	100%	Accepted	
13	Cumulative Sums Test	data1	0.313443	100%	Accepted
		data2	0.464856	100%	Accepted

A single bitstream was analyzed in ASCII mode, representing each bit as '0' or '1'. Since only one bitstream was tested, the proportion for each test is either pass (100%) or fail (0%). The results in Table 6.18 confirm that all tests met the p-value threshold (≥ 0.01), ensuring no detectable patterns exist in the encrypted image's binary representation. These findings support the encryption's cryptographic security and align with the principles outlined in (Gedda et al., 1977) and (X. Wu et al., 2025). The results confirm that the encrypted image's bitstream exhibits statistical randomness, reinforcing its cryptographic security.

6.5 Performance Validation and Comparative Security Analysis

The proposed encryption framework demonstrates robust security and performance through comprehensive metric evaluation. With a massive key space exceeding 2^{797} , the system offers strong protection against brute-force attacks. Pixel correlation values near zero and slightly negative across RGB channels confirm effective decorrelation, which is essential for resisting statistical attacks. Entropy values nearing the ideal of 8 validate the algorithm's ability to produce ciphertext with high randomness. Differential attack resistance is confirmed by high NPCR (99.617%) and UACI (31.37%) values, indicating strong sensitivity to minor pixel changes. High MSE and low PSNR in encrypted images, along with near-zero SSIM, suggest significant distortion and minimal structural similarity with the original, which enhances security. Perfect decryption results ($MSE = 0$, $PSNR = \infty$, $SSIM = 1$) confirm the reversibility and fidelity of the algorithm. Additionally, the scheme shows resilience against channel noise, maintaining SSIM values above 0.7 under Gaussian and salt-and-pepper noise. Collectively, these results validate the scheme's superior security and robustness when compared to contemporary state-of-the-art methods. Table 6.19 summarizes the achieved encryption performance metrics, benchmarks, and corresponding security interpretations for the proposed scheme.

Table 6.19: Performance Evaluation Summary of the Proposed Encryption Scheme

Metric	Achieved Value (Range or Avg.)	Ideal or Acceptable Standard	Performance Interpretation
Key Space	$> 2^{797}$	$> 2^{100}$	Extremely large key space; highly secure against brute-force attacks
Pixel Correlation (Encrypted)	-0.0364 to 0.0029 (Avg: ~ 0.0012)	Close to 0 or negative	Strong decorrelation in all RGB directions
Information Entropy	7.9992 – 7.9996	≈ 8	Near-perfect randomness; excellent confusion
UACI (%)	28.32 – 33.69 (Avg: 31.37)	$> 30\%$	Excellent diffusion; highly sensitive to small pixel changes
NPCR (%)	99.608 – 99.631 (Avg: 99.617)	$> 99\%$	High resistance to differential attacks
MSE (Encrypted)	8618 – 11221 (Avg: ~ 8934.9)	High indicates distortion	High distortion; good confusion and diffusion
PSNR (Encrypted)	7.63 – 8.78 dB (Avg: ~ 8.68 dB)	< 10 dB (lower is better)	Low visual similarity; strong encryption
MAE (Encrypted)	76.9 – 112.4 (Avg: ~ 90.6)	High indicates difference	Significant average error; effective encryption
SSIM (Encrypted)	0.0041 – 0.0076	Close to 0	Very low structural similarity; excellent security
MSE (Decrypted)	0	0	Perfect reconstruction of original
PSNR (Decrypted)	∞	∞	No error in decryption
SSIM (Decrypted)	1	1	Perfect structural similarity post-decryption
Noise Resilience (SSIM under noise)	0.774 (Gaussian), 0.738 (Salt & Pepper)	> 0.7 acceptable	Robust performance under noisy transmission
Encryption Time for Image with Size ($512 \times 512 \times 3$)			1.506313 seconds

6.6 Summary

The proposed image encryption and denoising technique, which takes part in chaotic systems, DNA coding, and an OSELM-based autoencoder for denoising, demonstrated enhanced security and robustness. Results analysis have shown that information entropy values for encrypted images nearing the ideal value of 8, NPCR values from 99.587% to 99.634%, and UACI values between 31.369% and 33.917%. These metrics highlight the algorithm's efficacy in preserving image quality and integrity against noise. Histogram analysis, correlation analysis, and MSE results indicate high resistance to statistical attacks. Moreover, the NPCR and UACI results confirm the algorithm's robustness against differential attacks. The algorithm's sensitivity to key variations ensures protection even with minor key changes, and its extensive key space (10^{240} or 2^{797}) significantly enhances resistance to brute force attacks. Additionally, the NIST SP800-22 statistical tests validate the randomness of the encrypted image's bitstream, further reinforcing its cryptographic strength. This method addressed the critical need for secure digital image transmission and storage across various fields, including medicine, photography, biology, astronomy, and defense. By combining robust encryption with an effective denoising OSELM autoencoder, the proposed technique ensures both the protection and clarity of digital images, offering a comprehensive solution for secure digital image handling.

CHAPTER 7

AN ADAPTIVE DEEP CNN-BASED COMPRESSION AND HYBRID INTEGER-FRACTIONAL ORDER HYPERCHAOTIC ENCRYPTION FRAMEWORK FOR SECURE IMAGE TRANSMISSION

7.1 Introduction

This chapter introduces an innovative dual-stage framework that integrates Convolutional Neural Network (CNN) for image compression with Memelements-based fractional-order hyperchaotic system for cryptography to address the problem of security and efficiency demands. The proposed methodology employed CNN models such as VGG19, and MobileNetV2 (MNV2) to intelligently segment images into contextually significant regions, facilitating adaptive compression ratios of up to 3.75 through efficient region-specific compression and Compressed Sensing (CS). Following this, a multi-layered encryption technique incorporated as Memelements-based fractional-order hyperchaotic system with a Logistic-Sine-Cubic Iterative Map (LSCIM), bolstered by deoxyribonucleic acid (DNA) sequence coding rules and dynamic diffusion, to guarantee resistance against evolving cryptanalytic threats. This proposed framework has been validated on five diverse image datasets, demonstrating its effectiveness. The system efficiently uses adaptive compression to maintain near-lossless visual quality while achieving strong encryption with ideal entropy, minimal pixel correlation, and resistance to differential attacks. MNV2 offers faster processing than traditional models, ensuring greater efficiency and protection against brute-force attacks. This framework balances visual fidelity, security, and efficiency, making it suitable for secure medical imaging, military communication, and real-time surveillance, where maintaining data integrity and confidentiality is essential.

7.1.1 Importance Map-Based Adaptive Compression

This section discusses the study of adaptive image compression and compressed sensing using an importance map. Each proposed technique is systematically analyzed to validate its theoretical backgrounds, mathematical model, and practical implications.

7.1.1.1 Content Analysis Using Convolutional Neural Networks (CNNs)

This subsection aims to extract the required features from the input digital image to prepare importance maps indicating high- and low-importance regions. Convolutional Neural Networks (CNNs) are deep learning architectures designed for feature extraction and pattern recognition (Nakazawa & Kulkarni, 2018; Zan et al., 2020). The convolutional layers' capture spatial hierarchies and semantic features. The CNN model is employed here due to its robust

performance in visual tasks (Hashimoto et al., 2024). Specifically, the final convolutional layer, *conv5_3*, is used for extracting feature maps (S. Ren et al., 2017). VGG16, VGG19, ResNet50, DenseNet121, and MNV2 are efficient deep-learning models; each gives unique structural benefits. MNV2 is designed for mobile applications, which can optimise image processing with efficient convolutions and shortcuts, making it appropriate for image encryption due to its balance of computational effectiveness and accuracy (Varun et al., 2024).

The MNV2 structure is prepared to improve efficiency and performance. It starts with an initial convolution of 3 x 3, stride (2) that decreases input resolution from (224 x 224 x 3) to (112 x 112 x 16). Its core invention is reversed residual blocks: a 1 x 1 convolution enlarges channels, for instance from 24 to 144, followed by a lightweight 3 x 3 depth-wise spatial filter, and a 1 x 1 projection layer to compress channels, for example from 144 to 32. Strides in choice blocks have spatial dimension 56 x 56 to 28 x 28, terminating in a last feature map for downstream tasks like saliency detection. As demonstrated in Figure 7.1, MNV2 utilizes inverted residual blocks with spatial downsampling to produce high-level features for compression. The model minimises computation using depth-wise independent convolutions (decoupling spatial and channel processes). Reversed residuals momentarily enlarge channels to keep features with least parameters, while linear bottlenecks skip non-linear activations in compressed layers to protect critical information. This balance makes it supreme for edge-device uses like adaptive compression.

The mathematical feature extraction process used to create the importance map is outlined. Given an input image $I \in \mathbb{R}^{H \times W \times C}$, where H and W denote the image's height and width, respectively, and C represents the number of channels, the convolutional layers process the image to extract meaningful spatial and semantic features (Alqahtani et al., 2021; Majstorović et al., 2023). The convolutional operation in a CNN layer is defined in equation (7.1).

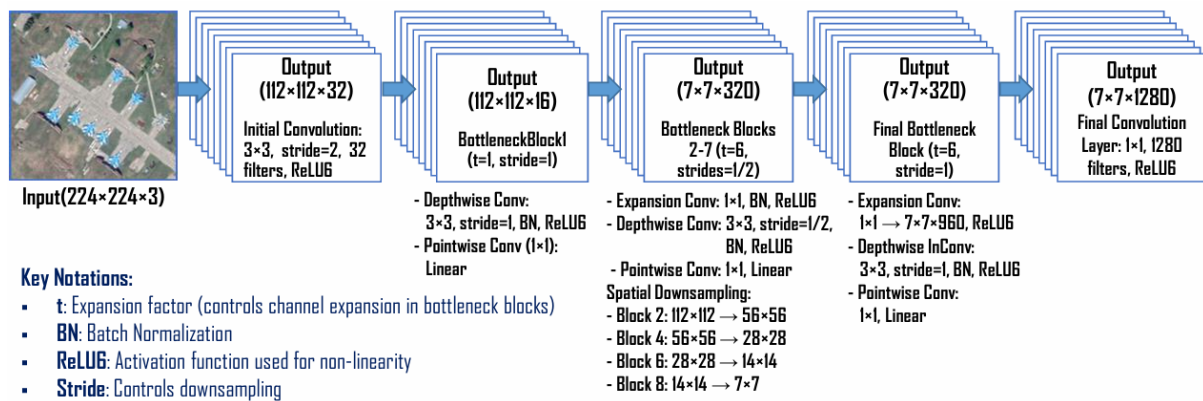


Figure 7.1: Flow diagram of the MNV2 Architecture

$$F_k(i, j) = \sigma \left(\sum_{m=1}^C \sum_{u,v} W_{k,m}(u, v) \cdot I_m(i + u, j + v) + b_k \right) \quad (7.1)$$

where, $F_k(i, j)$ is the activation at location in the k -th feature map, $W_{k,m}$ is the weights of the k -th filter, b_k is the bias of the k -th filter, and σ is the activation function (e.g., ReLU). The value $I_m(i + u, j + v)$ represents the pixel intensity in the m -th channel of the image at a location offset by u and v from the reference pixel (i, j) . This notation is crucial for defining local operations like convolution or image patch extraction.

The output feature maps are aggregated across channels to create an *importance map* (Alqahtani et al., 2021). The spatial dimensions (H' and W') of the output feature map are calculated as follows:

$$\begin{cases} H' = \left\lfloor \frac{H + 2P - K}{S} \right\rfloor + 1 \\ W' = \left\lfloor \frac{W + 2P - K}{S} \right\rfloor + 1 \end{cases} \quad (7.2)$$

The output feature maps of the convolutional layer are of size $H' \times W' \times C'$, where H' and W' are determined by the input dimensions, kernel size, padding, and stride, and C' corresponds to the number of filters. These feature maps encode the spatial and semantic information critical for downstream tasks.

The importance map is normalized for visualization and further processing:

$$I_{importance}(i, j) = \frac{\sum_{k=1}^K F_k(i, j)}{K} \quad (7.3)$$

where K is the number of feature maps.

7.1.1.2 Importance Map Generation

The goal is to create a map highlighting the spatial distribution of importance within the image, guiding the adaptive compression process. The importance map represents pixel-level importance by averaging feature map activations. High activation values correspond to regions of high semantic significance (e.g., edges, objects) (Alqahtani et al., 2021; Y. Wang, Qiu, et al., 2020).

Normalization of the importance map is performed as follows (Y. Wang, Qiu, et al., 2020):

$$I_{importance_norm}(i, j) = \frac{I_{importance}(i, j) - \min(I_{importance})}{\max(I_{importance}) - \min(I_{importance})} \quad (7.4)$$

A threshold is determined (e.g., using Otsu’s method or another heuristic (Dai et al., 2021; Smith et al., 1979)) to separate high and low-importance regions. Otsu’s method is often used to automatically select thresholds by minimizing intra-class variance. The thresholding results in two distinct binary masks:

- High-Importance Mask (M_{high}): This mask identifies pixels deemed necessary (i.e., those whose normalized importance map values exceed the threshold T).
- Low-Importance Mask (M_{low}): This mask identifies pixels with low importance (i.e., those whose normalized importance map values are below the threshold T).

The binary masks are defined as:

$$\begin{cases} M_{high}(i, j) = \begin{cases} 1, & \text{if } I_{importance_norm}(i, j) > T \\ 0, & \text{otherwise} \end{cases} \\ M_{low}(i, j) = 1 - M_{high}(i, j) \end{cases} \quad (7.5)$$

The binary masks control the compression rate applied to different image regions.

- For high-importance regions (where $M_{high}(i, j) = 1$), the image pixels are retained with minimal compression to preserve their quality.
- For low-importance regions (where $M_{low}(i, j) = 1$), higher compression is applied to reduce the overall data size, potentially with more aggressive quantization or even discarding some of the less important details.

7.1.1.3 Adaptive Compression Based on Importance Maps

Compression factors are assigned based on region importance. High-importance regions retain more information, while low-importance regions undergo higher compression to reduce data size.

The compression process for a region is modeled as follows:

$$\begin{cases} I_{compressed_high}(i, j) = I(i, j) \cdot M_{high}(i, j) \cdot Q_{high} \\ I_{compressed_low}(i, j) = I(i, j) \cdot M_{low}(i, j) \cdot Q_{low} \end{cases} \quad (7.6)$$

where Q_{high} and Q_{low} are compression quality factors for high and low-importance regions, respectively. The final compressed image is:

$$I_{compressed} = I_{compressed_high} + I_{compressed_low} \quad (7.7)$$

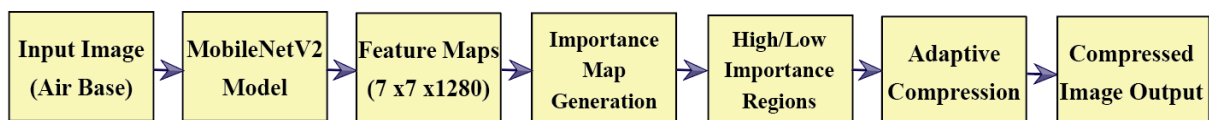


Figure 7.2: The Flow of MNV2 Based Compression Architecture

7.1.1.4 Simulated Compressed Sensing

Objective: To further reduce the size of the compressed image through down sampling, simulating the effect of compressed sensing.

Theoretical Explanation: Compressed sensing exploits the sparsity of signals to reconstruct them from fewer samples. Downsampling simulates reduced sampling rates, akin to absolute compressed sensing.

Mathematical Formulation: The downsampled image I_{CS} is generated as:

$$I_{CS}(i,j) = I_{compressed}(i \cdot s, j \cdot s) \quad (7.8)$$

where i and j are the coordinates of the reference pixel in the image, with i representing the row index and j representing the column index, and s is the scaling factor.

Figure 7.2 summarizes the MNV2 based proposed structure, which processes a $224 \times 224 \times 3$ Red, Green, and Blue image utilizing an initial convolution followed by depthwise independent convolutions and reversed residual blocks. These blocks effectively enlarge and diminish channel dimensions while preserving low computational costs. The final $7 \times 7 \times 1280$ feature map generates an importance heatmap by averaging through channels. Otsu's technique can then determine an optimum threshold for adaptive image compression, categorizing key features to reduce the image size based on intensity.

7.2 Methodology for Image Compression-Encryption Algorithm

The proposed framework integrates adaptive image compression and robust encryption to meet the needs of efficient storage and secure transmission. Compression employs CNN-based content analysis with VGG16, VGG19, and MNV2 to segment regions by importance, enabling targeted, region-specific compression. Simulated compressed sensing further enhances data efficiency. For encryption, a hybrid chaotic system combining a five-dimensional fractional-order hyperchaotic model and a Logistic-Sine-Cubic-Iterative Map (LSCIM) is used alongside DNA-based encoding, ensuring robust security. This approach balances compression efficiency with strong encryption, making it ideal for modern image-processing applications. As shown in Figure 7.3, the process involves adaptive image compression to reduce size, followed by fractional-order hyperchaotic encryption to secure the data. The encrypted image is transmitted through a potentially insecure environment and then decrypted and decompressed to reconstruct the original image with high fidelity.

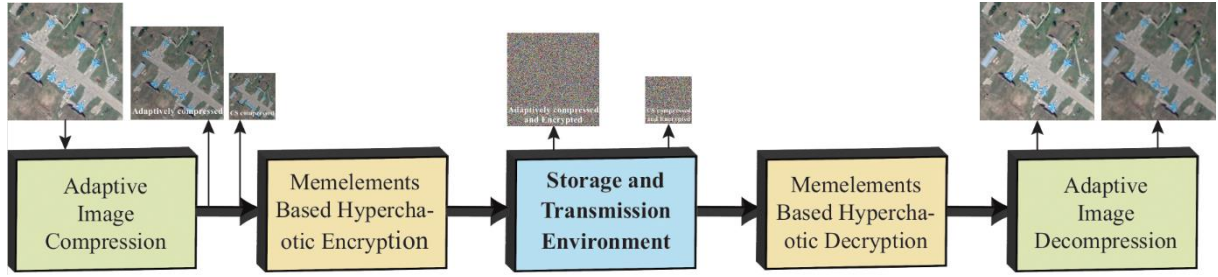


Figure 7.3: Secure Image Transmission Using Adaptive Compression and Fractional-order Hyperchaotic Encryption

7.2.1 Proposed Image Compression Algorithm

The first method uses CNN-based content analysis, adaptive region-specific compression, and simulated compressed sensing to achieve efficient image compression. Figure 7.4 illustrates the process of adaptive image compression. The image is first loaded and resized, followed by content analysis using CNN models to generate an importance map. The importance map guides the adaptive compression process. Subsequently, the content is again analyzed using CNN models, leading to two outputs: one for further processing and another for additional use, obtained through simulated compressed sensing (downsampling). The adaptive compression process consists of the following steps:

7.2.1.1 The Proposed Compression Algorithm Steps

Step 1: Input Image Loading and Resizing: Load and resize the input image to a fixed resolution (224 x 224 pixels) and save the resized image for further processing.

Step 2: Display Original Resized Image: Visualize the resized image to confirm it matches the intended dimensions.

Step 3: Content Analysis Using CNN Model: The CNN deep learning model, pre-trained on the ImageNet dataset, is utilized to extract feature maps from its *conv5_3* convolutional layer using the resized input image. The importance map is then computed by averaging the feature maps along their depth dimension and resized to match the image resolution for effective visualization.

Step 4: Normalization of Importance Map: The importance map is normalized to the 8-bit range for enhanced visibility. Next, a color map (e.g., jet) is applied to produce a visually distinct representation of the important regions, which is then saved as a colored image.

Step 5: Adaptive Compression-Based Importance Map: A threshold is calculated using Otsu's method to segment the importance map into high-importance and low-importance regions, represented by binary masks. A compression function is defined to apply region-specific compression with different quality factors, compressing each region separately based

on its mask. Finally, the compressed areas are combined to produce and save the final compressed image.

Step 6: Implement Compressed Sensing (CS): Downsample the compressed image to simulate the effect of compressed sensing by scaling it to 50% of its original size, then save the resulting images.

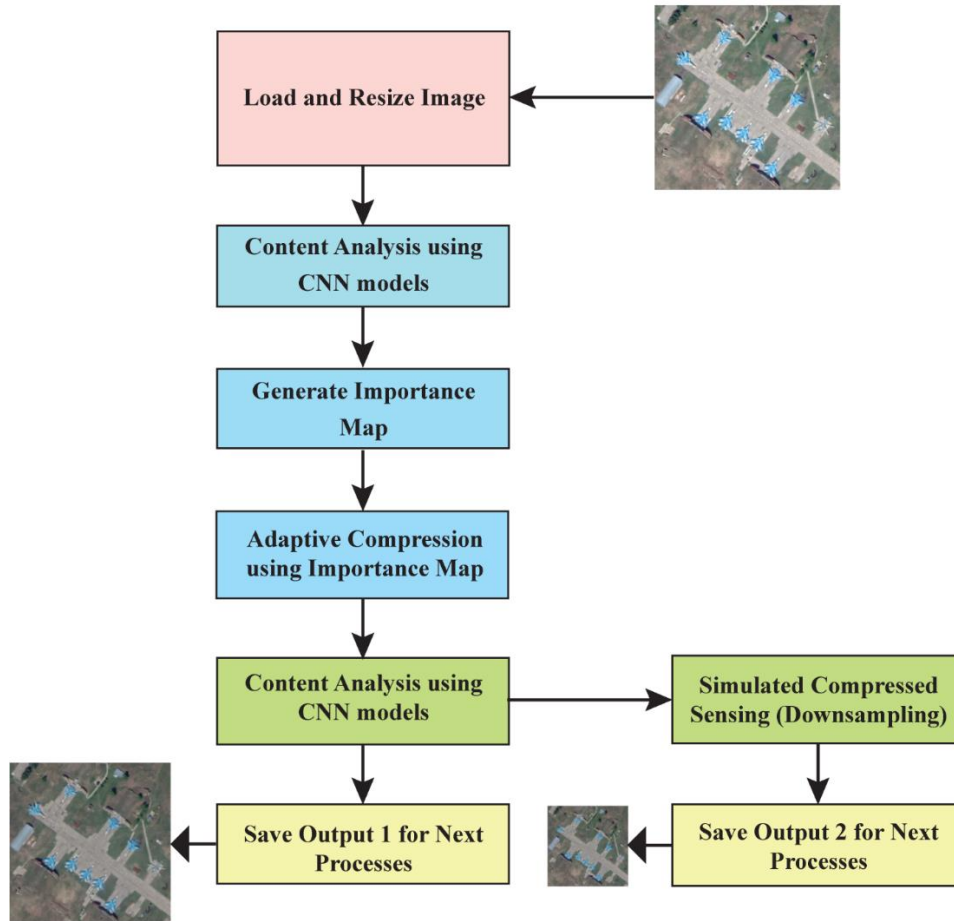


Figure 7.4: Flowchart of CNN-Based Importance Map for Adaptive Image Compression

7.2.2 Proposed Image Encryption Algorithm

The second method integrates a five-dimensional fractional-order hyperchaotic memcapacitor-meminductor model and a one-dimensional Logistic-Sine-Cubic Iterative Map (LSCIM) to generate highly sensitive, unpredictable, chaotic sequences. These sequences encode image data at the DNA level with addition, XOR, and diffusion operations, enhancing the algorithm's complexity and resilience against cryptographic attacks.

Figure 7.5 illustrates a process for adaptively compressing and encrypting color images using chaotic and fractional-order hyperchaotic systems. The input image undergoes preprocessing by splitting RGB color channels and filling zeros to match block sizes.

Table 7.1: The Proposed Adaptive Compression Pseudocode and Compressed Sensing

Algorithm: Image Processing, Compression, and Compressed Sensing
Input: Image I
Output: Compressed Image $I_{compressed}$, Simulated Compressed Sensing Image I_{CS}
Process: Adaptive Compression using CNN Models and Compressed Sensing
Preprocess Image:
$I_{resized} \leftarrow Resize(I, target_size)$ % Resize the input image
Content Analysis:
$F \leftarrow Resize(I, target_size)$ % Extract deep feature maps
$M_{importance} \leftarrow Mean(F \text{ along depth})$ % Compute the importance map
$M_{norm} \leftarrow Normalize(M_{importance})$ % Normalize importance map to [0, 1]
Segment Importance Map:
$T_{threshold} \leftarrow Otsu(M_{norm})$ % Perform thresholding
$Mask_{high} \leftarrow M_{norm} > T_{threshold}$ % Generate binary masks for high
$Mask_{low} \leftarrow M_{norm} \leq T_{threshold}$ % Generate binary masks for low
Adaptive Region-Based Compression:
% Compress high-importance regions with high quality:
$I_{compressed_high} \leftarrow Compress(I_{resized}, Mask_{high}, Q_{high})$
% Compress low-importance areas with lower quality:
$I_{compressed_low} \leftarrow Compress(I_{resized}, Mask_{low}, Q_{low})$
% Combine the compressed regions:
$I_{compressed} \leftarrow I_{compressed_high} + I_{compressed_low}$
Simulate Compressed Sensing:
% Downsample the compressed image
$I_{CS} \leftarrow Downsample(I_{compressed}, downsample_factor)$

Chaotic sequences are generated and combined with DNA encoding and arithmetic operations to diffuse the image blocks. A fractional-order hyperchaotic system solves for sequences that enhance encryption, and dynamic scrambling further rearranges and secures the channels. The encrypted image is obtained by merging the R, G, and B channels, producing adaptively compressed and encrypted outputs for subsequent applications. The process of encrypting a compressed image involves the following steps:

7.2.2.1 The Proposed Encryption Algorithm Steps

Step 1: Preprocessing and Zero Padding: Split the compressed image I into R, G, B , channels, pad each channel as needed to ensure dimensions are divisible by a predefined block size t .

Step 2: Generate LSCIM Chaotic Sequence: Compute the initial condition z_0 based on the grayscale values of the input image. Using z_0 as the starting point, generate a chaotic sequence p via the Logistic Singe-Coupled Iterative Map (LSCIM):

$$p_{i+1} = \left(r_2 \sin(\pi r_1 p_i (1 - p_i)) \right)^3 - r_3 r_2 \sin(\pi r_1 p_i (1 - p_i)) \quad (7.9)$$

Transform the sequence p into a two-dimensional matrix R , scaling its values to the range $[0, 255]$, required for encryption operations.

Step 3: Solve Fractional-order hyperchaotic Memcapacitor-Meminductor System: The Fractional-order hyperchaotic system equations are solved to generate chaotic sequences y_1, y_2, y_3, y_4, y_5 based on the following differential equations.

$$\begin{aligned}
 * D_{t_0}^{q_1} y_1 &= \frac{(a + b \sin y_4) y_2 - y_1}{ef} - \frac{(c + d \cos y_5) y_3}{f} \\
 * D_{t_0}^{q_2} y_2 &= \frac{y_1}{e} + \left(g - \frac{1}{e}\right) (a + b \sin y_4) y_2 \\
 * D_{t_0}^{q_3} y_3 &= y_1 \\
 * D_{t_0}^{q_4} y_4 &= y_2 \\
 * D_{t_0}^{q_5} y_5 &= y_3
 \end{aligned} \tag{7.10}$$

where, $a, b, d, e, f, g, q_1, q_2, q_3, q_4$, and q_5 are system parameters. These sequences are computed using the given initial conditions $y_{1_0}, y_{2_0}, y_{3_0}, y_{4_0}, y_{5_0}$, and they provide the chaotic states required for encryption operations.

Step 4: DNA Coding and Arithmetic Operations:

DNA Encoding: Encode the image I and chaotic matrix R using chaotic sequences X and Y :

$$\begin{aligned}
 Q_1 &= \text{DNAEncode}(I, X) \\
 Q_2 &= \text{DNAEncode}(R, Y)
 \end{aligned} \tag{7.11}$$

Step 5: DNA Arithmetic Operations: Perform DNA-based arithmetic operations using Z to combine Q_1 , and Q_2 .

$$Q_{\text{last}} = \text{DNAOperation}(Q_1, Q_1, Z) \tag{7.12}$$

Here, the arithmetic operations could include addition, subtraction, XOR, or XNOR, as defined by Z .

Diffuse the encoded blocks by incorporating the results from previously processed blocks. For each block Q_i :

$$Q_{\text{diffused}} = \text{DNAOperation}(Q_i, Q_{\text{previous}}, Z) \tag{7.13}$$

This ensures that changes in one block propagate throughout the encrypted image, enhancing security.

Step 6: Anti-Clipping and Dynamic Scrambling:

Generate Sorting Sequences: using the chaotic keys U_x and U_y derived from the LSCIM, compute sorting indices for rows and columns:

$$\text{SortIndices}_x = \text{Sort}(U_x, \text{descending}) \tag{7.14}$$

$$\text{SortIndices}_x = \text{Sort}(U_x, \text{descending})$$

Dynamic Scrambling: Rearrange the rows and columns of the RGB channels using the sorting indices. For each channel (R, G, B):

$$\begin{aligned} \text{Channel}_i(\text{row}) &= \text{Channel}_i(\text{SortIndices}_x) \\ \text{Channel}_i(\text{column}) &= \text{Channel}_i(\text{SortIndices}_y) \end{aligned} \quad (7.15)$$

This scrambling ensures pixel positions are dynamically altered, significantly increasing the attack resistance while maintaining the anti-clipping mechanism.

Step 7: Output Encrypted Image: The final step involves merging the processed R, G and B channels to construct the encrypted image $Q_{\text{encrypted}}$:

$$Q_{\text{encrypted}} = \text{MergeChannels}(Q_R, Q_G, Q_B) \quad (7.16)$$

The encrypted image is then saved to the designated output location for secure storage or transmission.

The decryption process reverses the steps of the encryption process to restore the original image. It splits the encrypted image into R, G , and B channels. The hyperchaotic sequences used during encryption are regenerated using the same initial conditions and parameters. Dynamic descrambling is performed to reorder the channels, followed by inverse DNA decoding and arithmetic operations to reconstruct the image blocks. Finally, the decrypted image is merged from the channels and processed to remove padding, yielding the original adaptively compressed color image.

7.3 Result and Discussion

This section presents the results of the proposed adaptive image compression and encryption algorithms, tested on six images (two standard, two military, and two medical) of varying sizes from Kaggle and USC-SIPI datasets. Simulations were performed in MATLAB 2023a on a computer with 8 GB RAM and Intel Core i7 processor with 3.20 GHz operating speed. Windows with an Intel® Core™ i7 CPU@ 3.20 GHz and 8GB RAM. For compression, CNN Models-based importance maps were used to identify high- and low-importance regions, with factors of 0.8 and 0.4, respectively, followed by 50% downsampling using simulated Compressed Sensing (CS). Encryption employed the LSCIM system with parameters ($z_0=0.409, r_1=2.5, r_2=6, r_3=6$) and a memelement-based fractional-order hyperchaotic system with initial conditions ($X_0=1, Y_0=1, Z_0=0, U_0=0, W_0=0$) and parameters ($a=0.41, b=2.81, c=10, d=3, e=0.2, f=1, g=2, q_1=0.506, q_2=0.505, q_3=0.506, q_4=0.505, q_5=0.506, h=0.001$).

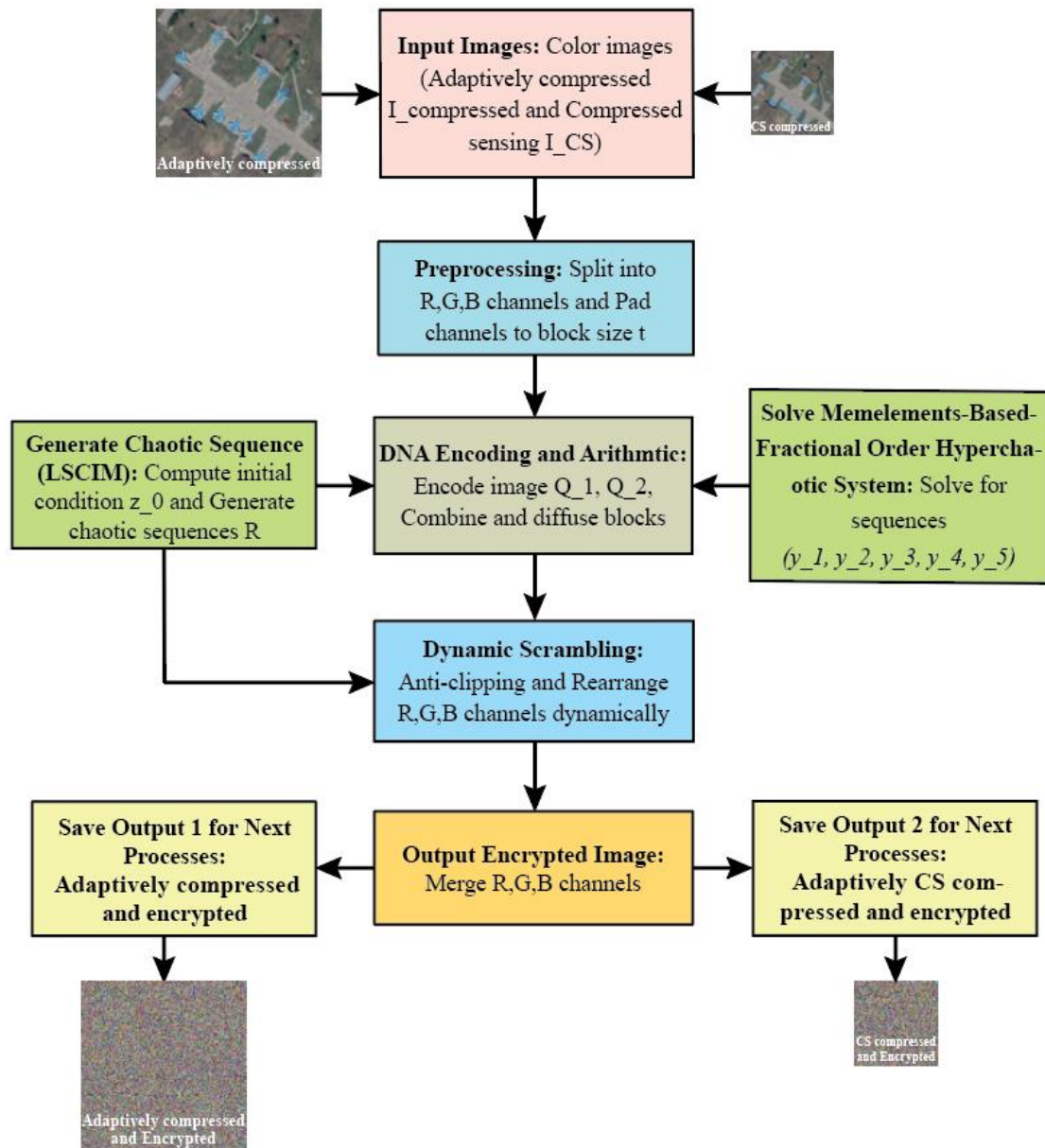


Figure 7.5: Encryption Process for Adaptively Compressed Color Images Using Fractional-order hyperchaotic and Chaotic Systems

Additionally encryption keys included padding values ($M1=0$, $N1=0$) and gray value keys ($z(0)=0.3563$, $z1(0)=0.2051$). These methods demonstrated efficient compression and secure encryption under the specified parameters and conditions. The analysis covers multiple aspects, including compression quality, computational efficiency, and the impact of encryption and compressed sensing (CS). The results demonstrate the effectiveness of the proposed method in preserving image quality while achieving significant compression and maintaining security through encryption.

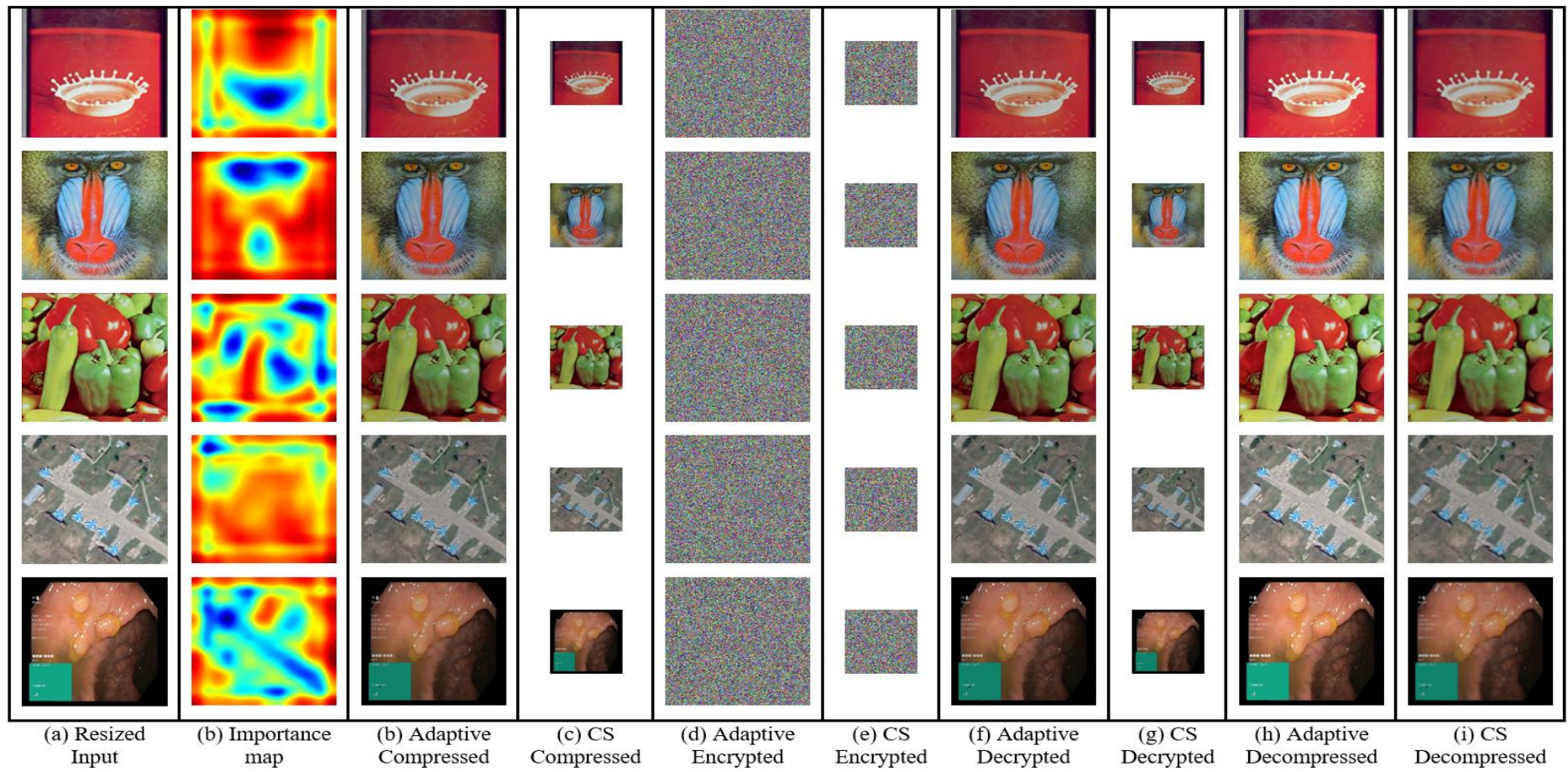


Figure 7.6: Stages of Image Processing: Adaptive Compression and Compressed Sensing with Encryption-Decryption

Figure 7.6 illustrates the complete image processing results using Adaptive Compression and Compressed Sensing (CS) combined with hyperchaotic encryption-decryption techniques. Starting with the resized input (a), an importance map (b) highlights the significant regions for adaptive compression. The images are then compressed using adaptive (c) and CS (d) methods, followed by their encryption (e, f). Subsequent decryption of the encrypted images (g, h) restores the data for final reconstruction, showcasing the decompressed results from adaptive compression (i) and CS methods (j). This figure shows the strength and quality retention of the suggested methods across all stages, ensuring efficient and secure image processing.

7.3.1 Image Compression and Decompression Efficiency Evaluation Metrics

The proposed image compression and encryption method uses an adaptive digital image compression technique, adapted using importance maps from one of the CNN models VGG16, VGG19, or MNV2. Compression features are assigned based on this importance map; minimum compression is done for high-importance regions, and maximum compression is done for low-importance regions. Compressed sensing (CS) is also applied to enhance the overall compression efficiency. The performance evaluation is based on two key metrics: Compression Ratio and Quality Metrics, which include PSNR, SSIM, and MSE.

7.3.1.1 Compression Ratio

The compression ratio is a crucial performance metric that measures the efficiency of the compression algorithm. It is defined as the ratio of the original image size to the compressed image size. A higher compression ratio implies better data reduction while preserving the image quality.

Adaptive Compression: In the adaptive compression approach, the compression ratio is influenced by the segmentation of the image into high and low-importance regions, as identified by the importance map. These regions are compressed differently based on their importance, as follows:

- High importance regions undergo minimal compression (i.e., higher retention of original quality).
- Low-importance regions undergo higher compression (i.e., more significant data reduction)

The compression ratio (adaptive compression) is calculated as:

$$\text{Compression Ratio (Adaptive)} = \frac{\text{Original Image Size}}{\text{Adaptively Compressed Image Size}} \quad (7.17)$$

where, *Original Image Size* is the total number of pixels in the original image, and *Compressed Image Size* is the weighted sum of the high and low importance regions’.

The compressed image size for the adaptive compression can be calculated by:

$$\begin{aligned} \text{Compressed Image Size} \\ = (\text{nnz}(\text{highImportanceMask}) \times \text{highCompressionFactor}) \\ + (\text{nnz}(\text{lowImportanceMask}) \times \text{lowCompressionFactor}) \end{aligned} \quad (7.18)$$

Compressed Sensing (CS): Compressed sensing (CS) is a technique that allows the compression of an image by measuring fewer samples than traditionally required. It is often used to simulate highly efficient data reduction, particularly for images with sparse representations. The Compression Ratio (CS) for the compressed sensing approach is defined as:

$$\text{Compression Ratio (CR)} = \frac{\text{Original Image Size}}{\text{Compressed Sensing Image Size}} \quad (7.19)$$

where, *Original Image* is the number of pixels in the original image, and *Compressed Sensing Image Size* is the size of the Downsampled obtained after applying CS.

7.3.1.2 Quality Metrix

PSNR, MSE, and SSIM are three basic metrics used to test and evaluate the proposed image compression method. These metrics assess the image quality after compression and decompression.

Peak Signal-to-Noise Ratio (PSNR): One of the commonly applicable metrics to assess the quality of a compressed digital image is PSNR. A higher PSNR value shows more suitable image quality.

$$\text{PSNR} = 10 \times \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) \quad (7.20)$$

where, MAX_I is the maximum possible pixel value of the image, and MSE is the Mean Squared Error.

Structural Similarity Index (SSIM): SSIM is a perceptual metric that quantifies the perceived similarity between two images. SSIM ranges from 0 to 1, where 1 indicates identical images.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (7.21)$$

where, μ_x, μ_y are the mean pixel intensities of images x and y , σ_x, σ_y are the standard deviations of x and y , σ_{xy} is the covariance between x and y , C_1 and C_2 are coefficients used to stabilize the commutation.

Mean Squared Error (MSE): MSE measures the average squared difference between the original and compressed images. Lower MSE values indicate better quality.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (I(i) - K(i))^2 \quad (7.22)$$

where, $I(i)$ and $K(i)$ are the pixel values of the original and compressed images, respectively.

Table 7.2: Metrics Comparison after Encryption, Decryption, and Decompression

Comparison of Metrics After Compression Encryption, Decryption, and Decompression (Peppers)						
Metrics	Adaptive Compression Technique			Adaptive Compressed Sensing (CS)		
	Using CNN Models			Using CNN Models		
	VGG16	VGG19	MNV2	VGG16	VGG19	MNV2
Compression Ratio	3.7503	3.7503	3.7506	4	4	4
PSNR (dB)	55.2462	55.2462	55.2438	19.4897	19.4897	19.4778
SSIM	0.99994	0.99994	0.99994	0.93202	0.93202	0.93166
MSE	0.19430	0.1943	0.1944	731.328	731.3276	733.3267
Computation Time (sec.)	0.2336	0.2154	0.04828	0.0499	0.0339	0.02837

Table 7.2 compares the performance metrics of Adaptive Compression and Compressed Sensing (CS) using CNN models (VGG16, VGG19, MNV2) on the "Peppers" image after compression, encryption, decryption, and decompression. Adaptive Compression achieves a compression ratio of 3.75, delivering high image quality with PSNR values around 55 dB, SSIM of 0.99994, and low MSE (0.194), with computation times of ~0.23 sec for VGG16, ~0.22 sec for VGG19, and ~0.05 sec for MNV2. In contrast, CS provides a higher compression ratio of 4 but lower image quality, with PSNR around 19.5 dB, SSIM ~0.93, and MSE ~731, while achieving faster computation (~0.03 to 0.05 sec). MNV2 offers significantly reduced computation time in both techniques, making it suitable for applications requiring speedier

processing at a slight trade-off in quality. This underscores the trade-offs between compression efficiency, image fidelity, and processing speed across models.

The proposed adaptive compression-encryption framework was evaluated using VGG16, VGG19, and MobileNetV2. VGG16 and VGG19 had processing times of 0.2336 seconds and 0.2154 seconds, respectively, while MobileNetV2 significantly improved this to 0.04828 seconds, outperforming VGG16 by approximately 79.34% and VGG19 by about 77.61%. This highlights MobileNetV2's lightweight design, offering fast execution while maintaining high reconstruction quality ($\text{PSNR} > 55 \text{ dB}$ and $\text{SSIM} \approx 1.0$), making it ideal for secure image transmission in real-time, resource-limited scenarios. Additionally, the adaptive compressed sensing model outperforms the standard technique with a computational time of 0.04828 seconds reduced to 0.02837 seconds, yielding a time improvement of 0.01991 seconds, about 41.24%. This optimized model is 41.24% faster while ensuring high image quality and cryptographic performance, making it suitable for real-time applications.

7.3.2 Mathematical Analysis and Performance Evaluation of Image Compression, Encryption, and Decompression Using Chaotic and Hyperchaotic Systems

This section presents the results and discussions for the image compression, encryption, decryption, and decompression processes using Convolutional Neural Networks (CNN) with VGG16, VGG19, and MNV2 architectures alongside Compressed Sensing (CS). The results are grouped into sections based on the CNN models (VGG16, VGG19, and MNV2) and further divided into six processes: Compression Metrics, Adaptive Encryption Metrics, Adaptive CS Encryption Metrics, Adaptive Decryption Metrics, Adaptive CS Decryption Metrics, and Decompression Metrics. The CNN models' compression results are analyzed and compared using test images such as Crown-Splash, Mandrill, Peppers, Air Base, and GI-tracts.

7.3.2.1 Compression and Encryption of VGG16 Model-Based Compressed Images

The results are shown in tables, followed by a mathematical analysis of key metrics such as PSNR, SSIM, compression ratio, encryption time, entropy, etc.

Compression Metrics: Compression performance is evaluated by two key factors: compression time and ratio. Table 7.3 compares the compression performance across three CNN models (VGG16, VGG19, and MNV2) for various input images. For every CNN model, the adaptive compression and compressed sensing methods assess the compression running time and ratio.

Table 7.3: Compression Time and Ratio for Various CNN Models and Images

Input Images				Crown-Splash	Mandrill	Peppers	Air-Base	GI-tract
CNN Model used	VGG16	Compression	Adaptive	0.2300	0.2336	0.2336	0.2166	0.2010
		Time (sec.)	CS	0.0308	0.0499	0.0499	0.0353	0.0300
		Compression	Adaptive	3.7504	3.7504	3.7503	3.7501	3.7504
		Ratio	CS	4.0000	4.0000	4.0000	4.0000	4.0000
	VGG19	Compression	Adaptive	0.0667	0.0646	0.2154	0.1618	0.2387
		Time (sec.)	CS	0.0637	0.0396	0.0339	0.0297	0.0480
		Compression	Adaptive	3.7503	3.7503	3.7501	3.7502	3.7502
		Ratio	CS	4.0000	4.0000	4.0000	4.0000	4.0000
	MNV2	Compression	Adaptive	0.0563	0.0415	0.0483	0.0483	0.0570
		Time (sec.)	CS	0.0263	0.0255	0.0294	0.0282	0.0245
		Compression	Adaptive	3.7511	3.7540	3.7506	3.7508	3.7542
		Ratio	CS	4.0000	4.0000	4.0000	4.0000	4.0000

The adaptive method shows consistent compression ratios with slightly higher computation times, reflecting its computational complexity. The CS technique also gives faster compression times while keeping a uniform compression ratio 4.0000 across all models and images. This analysis highlights the trade-off between computational efficiency and compression reliability across CNN architectures.

Adaptively Compressed Encryption Metrics: Table 7.4 summarizes the performance metrics of adaptive encryption methods implemented using three CNN models: VGG16, VGG19, and MNV2. The table includes the number of keys, entropy (both original and encrypted), red correlation coefficients (original and encrypted), and encryption computation time for five distinct images (Crown-Splash, Mandrill, Peppers, Air-Base, and GI-tract).

Entropy (H): Entropy measures an image's amount of information or uncertainty. It is calculated using the formula:

$$H(x) = - \sum_{i=0}^{L-1} p_i \log_2 p_i \quad (7.23)$$

where x is the image, L is the number of possible intensity levels, and p_i is the probability of occurrence of the intensity level i in the image. The entropy for the original image (before encryption) and the encrypted image (after encryption) are calculated similarly.

Correlation Coefficients (Red Correlation - R): The correlation coefficient measures the degree of linear relationship between pixel intensities in the image (X. Chen et al., 2023; Ibrahim et al., 2022). For two images, I and J, the correlation coefficient R is computed as:

$$R(I, J) = \frac{\sum_{i,j} (I(i,j) - \bar{I})(J(i,j) - \bar{J})}{\sqrt{\sum_{i,j} (I(i,j) - \bar{I})^2 \sum_{i,j} (J(i,j) - \bar{J})^2}} \quad (7.24)$$

where $I(i,j)$ and $J(i,j)$ are the pixel intensities of images I and J at location (i,j) , and $\bar{I}$ and $\bar{J}$ are their mean pixel intensities. Figure 7.7 (a-c) illustrates the correlation test outcomes for the input test image. Figure 7.8 shows adjacent pixels in the encrypted image, which lack apparent correlation and have a better correlation in the original image. This indicates that the hyperchaotic system provides effective encryption and security. The results in Table 7.4 demonstrate the robustness of the encryption process, as indicated by the high entropy values (close to 8) for encrypted images and the near-zero red correlation coefficients, ensuring effective decorrelation. MNV2 exhibits faster encryption times among the models than VGG16 and VGG19, making it more computationally efficient.

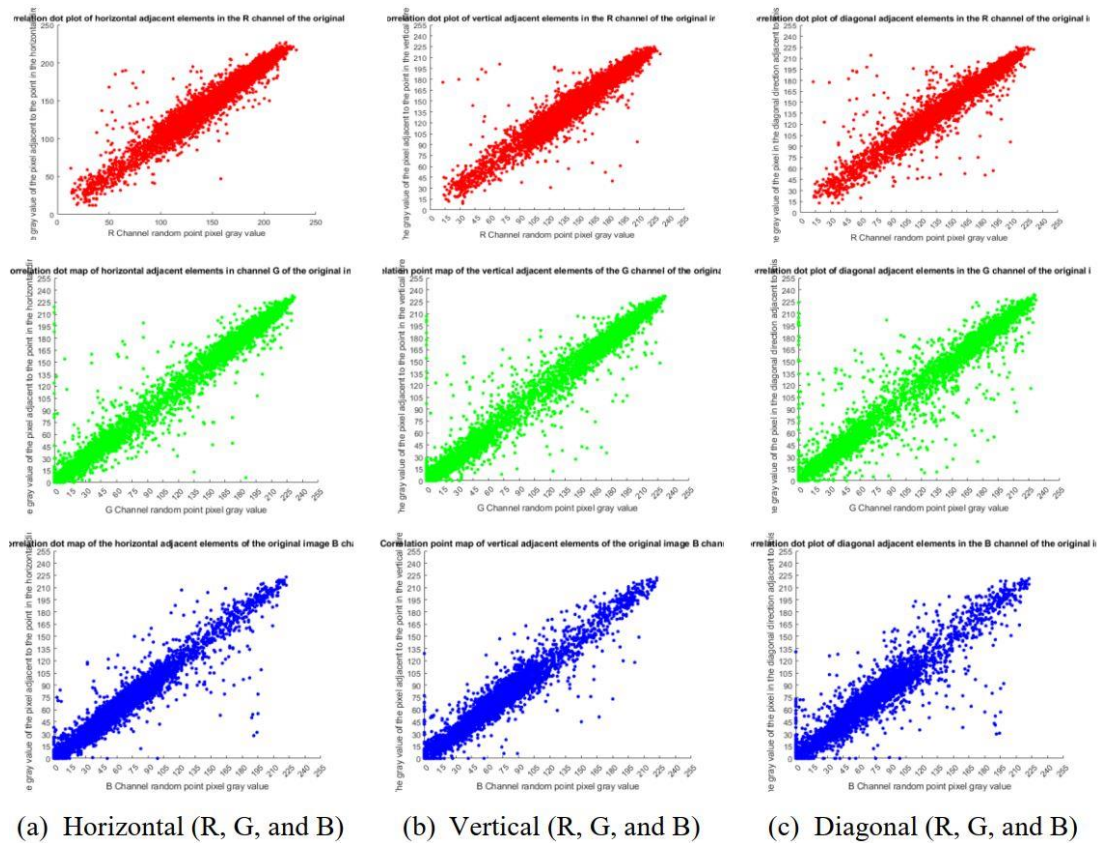
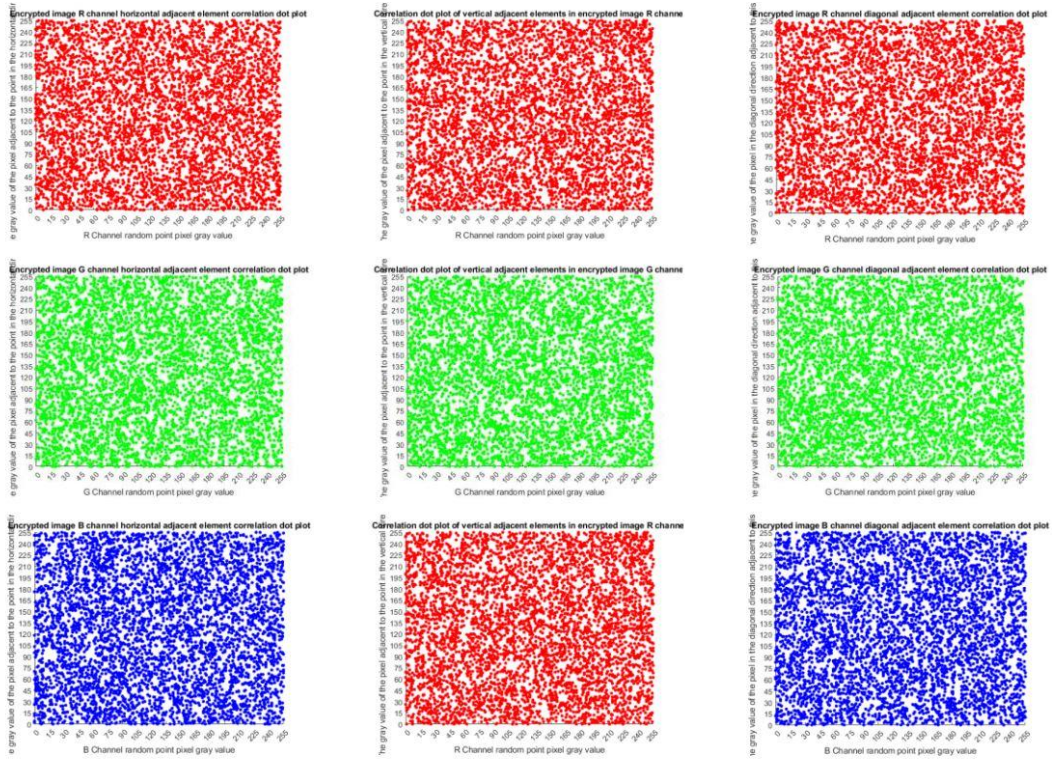


Figure 7.7: The Original Input Pepper Test Image's Horizontal-Vertical-Diagonal Correlations



(a) Horizontal (R, G, and B) (b) Vertical (R, G, and B) (c) Diagonal (R, G, and B)

Figure 7.8: Pepper's Ciphered Image Horizontal-Vertical-Diagonal Correlations

Table 7.4: Performance Metrics of Adaptive Encryption Using CNN Models

Metric	Model	Crown-Splash	Mandrill	Peppers	Air-Base	GI-tract
Number of Keys	VGG16	24 Keys	24 Keys	24 Keys	24 Keys	24 Keys
	VGG19	24 Keys	24 Keys	24 Keys	24 Keys	24 Keys
	MNV2	24 Keys	24 Keys	24 Keys	24 Keys	24 Keys
Entropy (Original)	VGG16	6.2341	7.12877	6.93003	6.37573	6.51623
	VGG19	6.23403	7.12873	6.93	6.376	6.5163
	MNV2	6.23507	7.13320	6.92997	6.37750	6.51413
Entropy (Encrypted)	VGG16	7.9964	7.99607	7.99647	7.9965	7.99640
	VGG19	7.99633	7.9964	7.9965	7.99637	7.9964
	MNV2	7.99633	7.99633	7.99633	7.99637	7.99630
Red Correlation (Original)	VGG16	0.98500	0.94398	0.96348	0.93136	0.99206
	VGG19	0.98414	0.94806	0.9618	0.92449	0.99041
	MNV2	0.98445	0.94547	0.96006	0.92965	0.99175
Red Correlation (Encrypted)	VGG16	-0.00219	0.01565	-0.0172	0.00168	0.00253
	VGG19	0.01418	0.01545	0.01861	-0.00825	-0.00059
	MNV2	0.01870	0.00182	-0.00558	-0.00261	0.00515
Encryption Time (sec.)	VGG16	0.60708	0.78602	1.1778	0.6698	0.78858
	VGG19	0.60708	0.78602	1.1778	0.6698	0.78858
	MNV2	0.37985	0.39217	0.38507	0.39643	0.38649

Compressed Encryption Histogram Analysis: Histograms reveal images' pixel concentration distribution, with input images showing distinct patterns. Encrypted images exhibit uniform distributions, ensuring resistance to statistical attacks (Alexan, Gabr, et al., 2023). Figures 7.9–7.12 compare the RGB histograms of original and adaptively compressed encrypted images, confirming the encryption method's effectiveness in concealing pixel redundancy and enhancing security (Alexan et al., 2022).

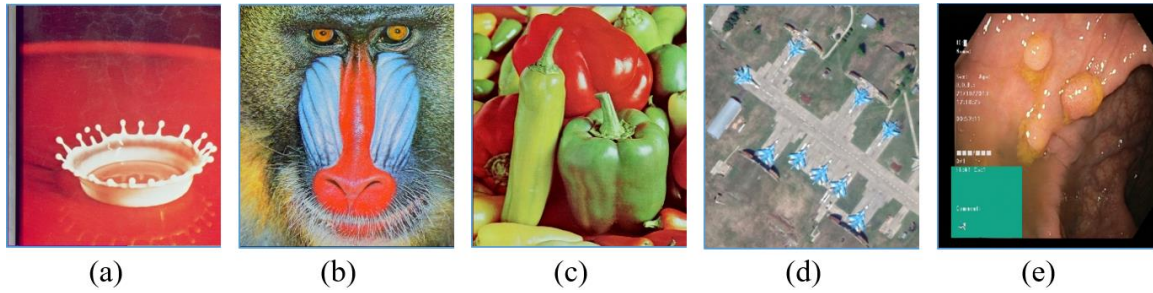


Figure 7.9: Input original images. (a) Crown splash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract

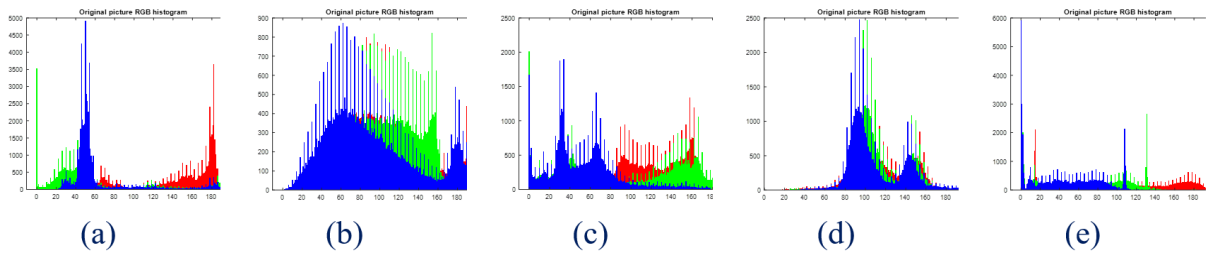


Figure 7.10: Input original images RGB Histograms. (a) Crownsplash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract

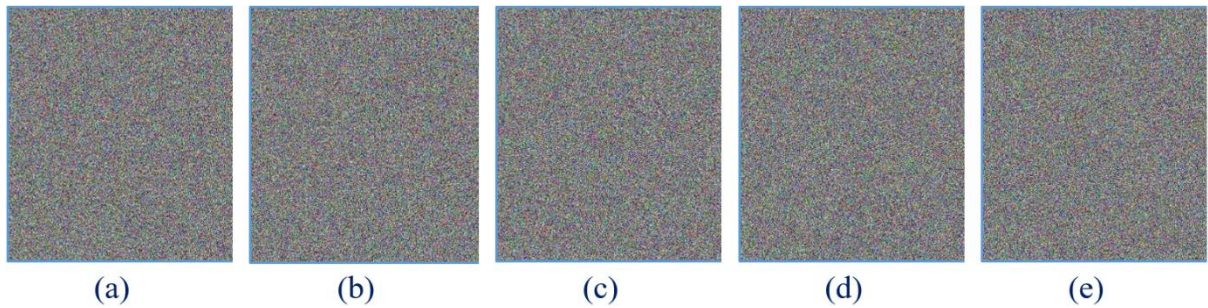


Figure 7.11: Ciphered images. (a) Crown-Splash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract

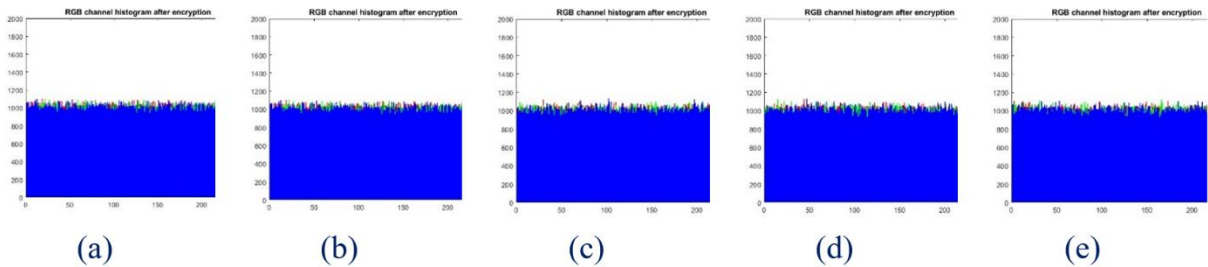


Figure 7.12: Ciphered images RGB Histograms. (a) Crown-Splash, (b) Mandrill, (c) Peppers, (d) Air-Base, (e) GI-tract

Adaptively CS Compressed Encryption Metrics: Table 7.5 integrates results from the combined table of metrics (as requested) and includes datasets such as Crown-Splash, Mandrill, Peppers, Air-Base, and GI-Tract for three CNN models (VGG16, VGG19, and MNV2) across key metrics like entropy and red correlation values.

Table 7.5: Adaptive CS Encryption Results with CNN Models Across Datasets

Metric	Model	Data Type	Crown-Splash	Mandrill	Peppers	Air-Base	GI-tract
Entropy	VGG16	Original	7.12953	7.12953	7.0073	6.39533	6.56310
		Encrypted	7.98623	7.98670	7.9852	7.98390	7.98663
	VGG19	Original	6.3370	7.12947	7.0073	6.39563	6.56423
		Encrypted	7.9863	7.98647	7.9846	7.9863	7.98567
	MNV2	Original	6.23507	7.13457	7.00723	6.39700	6.56020
		Encrypted	7.99633	7.98597	7.98547	7.98767	7.98587
Red Correlation (Original)	VGG16		0.94260	0.94260	0.92319	0.85877	0.98689
	VGG19		0.96037	0.93808	0.92527	0.85830	0.98794
	MNV2		0.95925	0.93772	0.91765	0.85948	0.98720
Red Correlation (Encrypted)	VGG16		0.00247	0.00247	-0.0309	0.01838	-0.0048
	VGG19		-0.00810	0.00049	0.03507	-0.00781	-0.00220
	MNV2		0.00993	0.01788	0.02128	-0.00016	0.01795

Adaptively Compressed Decryption Metrics: The results discussed are summarized in Table 7.6, which consolidates the metrics for decryption performance across three CNN models (VGG16, VGG19, and MNV2) for adaptive CS compression across various datasets. Decryption quality is generally high, particularly for the GI-Tract dataset, with low MSE, the Mean Absolute Error (MAE), and PSNR and SSIM values. On the other hand, the encryption process introduces significant distortion, as evidenced by high MSE, high MAE, and low PSNR and SSIM between the original and encrypted images. Despite this, encryption achieves high randomness, which is reflected in the high NPCR and UACI values, demonstrating the robustness and security of the approach.

A high MAE is required to distinguish between encrypted and original test pictures to ensure that the cryptographic system is reliable. It is mathematically expressed as:

$$MAE = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N [S(m,n) - E(m,n)] \quad (7.25)$$

The pixels of the original sample plain image are referred to as $S(m,n)$, while the pixels of the cyphered picture in m rows and n columns are referred to as $E(m,n)$. M and N represent the image's dimensions.

Table 7.6: Decryption Results of Adaptive Compression with CNN Models

Metric	Model	Data Type				
		Crown-Splash	Mandrill	Peppers	Air-Base	GI-Tract
MSE Between Original and Decrypted (dB)	VGG16	691.48	750.31	662.75	832.63	435.44
	VGG19	690.44	750.15	661.64	832.32	434.75
	MNV2	692.39	753.84	664.77	834.49	434.27
MSE Between Original and Encrypted (dB)	VGG16	11176.8	8189.9	10044.4	6773.95	12501.23
	VGG19	11148.18	8176.33	10044.13	6758.63	12512.85
	MNV2	11148.27	8211.27	10034.96	6755.06	12515.11
PSNR Between Original and Encrypted (dB)	VGG16	7.65	9.00	8.11	9.82	7.16
	VGG19	7.66	9.01	8.11	9.83	7.16
	MNV2	7.66	8.99	8.12	9.83	7.16
PSNR Between Original and Decrypted (dB)	VGG16	19.73	19.38	19.92	18.93	21.74
	VGG19	19.74	19.38	19.92	18.93	21.75
	MNV2	19.73	19.36	19.90	18.92	21.75
MAE Between Original and Encrypted (dB)	VGG16	86.367	74.45	81.913	69.193	91.513
	VGG19	86.1933	74.553	81.8867	69.08	91.5033
	MNV2	86.267	74.7767	81.8467	69.0633	91.55
NPCR (%)	VGG16	99.6074	99.5808	99.5987	99.6021	99.6034
	VGG19	99.5974	99.6007	99.5987	99.5934	99.6353
	MNV2	99.6147	99.6233	99.622	99.6014	99.5948
UACI	VGG16	33.8693	29.2606	32.1232	27.135	35.8885
	VGG19	33.802	29.2372	32.1122	27.0907	35.883
	MNV2	33.8311	29.3252	32.0969	27.0829	35.9022
SSIM (Original vs Decrypted)	VGG16	0.95249	0.95293	0.95224	0.95696	0.96127
	VGG19	0.95262	0.95297	0.95239	0.95731	0.96133
	MNV2	0.95245	0.95183	0.95188	0.95688	0.96118
SSIM (Original vs Encrypted)	VGG16	0.00587	0.00930	0.00420	0.00889	0.00604
	VGG19	0.00799	0.00977	0.00614	0.00889	0.00806
	MNV2	0.008915	0.00419	0.00772	0.00963	0.00569

The structural similarity index measure (SSIM) is used to determine how similar the processed image (E) and original image (O) (Ahuja & Doriya, 2023) (Khushwant et al., 2023). It is given as,

$$SSIM(O, E) = \frac{(2\mu_O\mu_E + \alpha)(2\sigma_{OE} + \beta)}{(\mu_O^2 + \mu_E^2 + \alpha)(\sigma_O^2 + \sigma_E^2 + \beta)} \quad (7.26)$$

The constants are represented by α and β , whereas the mean, variance, and covariance between O and E are denoted by μ , σ^2 , and σ , respectively.

The suggested method is compared with other methods shown in Table 7.7, and it performs better in handling subtle image dissimilarities and has improved attack resilience. The proposed method achieves an average NPCR of 99.6065%, indicating high pixel randomness and resilience against differential attacks. The UACI value for the proposed method averages 33.8341%, showcasing its effectiveness in altering pixel intensities during encryption.

Table 7.7: The Results Discussion of UACI and NPCR for Encrypted Image

Algorithms	NPCR				UACI			
	R	G	B	Avg.	R	G	B	Avg.
Proposed (Crown Splash)	99.6074	99.5974	99.6147	99.6065	33.8693	33.802	33.8311	33.8341
Ref. (Ibrahim et al., 2022)	99.475	99.685	99.485	99.548	32.895	33.102	32.805	32.934
Ref. (Shakir et al., 2022)	99.6124	99.6124	99.6124	99.6124	29.2827	29.2827	29.2827	29.2827
Ref. (H. Wang et al., 2023)	99.6090	99.5987	99.5972	99.6016	33.3996	33.5134	33.4727	33.4619

Table 7.8 demonstrates PSNR and MSE results for the encrypted crown splash test image and compares them with the existing techniques in the literature.

Table 7.8: MSE and PSNR Comparative Values of Input and Encrypted Emage

Different Methods	MSE				PSNR (dB)			
	R	G	B	Avg.	R	G	B	Avg.
Adaptively Compressed (VGG16) and Encrypted (Proposed)	11176.8	11176.8	11176.8	11176.8	7.65	7.65	7.65	7.65
Proposed Adaptively CS (VGG16) and Encrypted	11156.63	11156.63	11156.63	11156.63	7.66	7.66	7.66	7.66
Ref. (Ahuja & Doriya, 2023)	-	-	-	-	18.66	18.66	18.66	18.66
Ref. (M. A. Lone & Qureshi, 2023)	9004.40	9004.40	9004.40	9004.40	8.6454	8.6454	8.6454	8.6454
Ref. (Alexan et al., 2022)	10,663.3	8982.02	7021.33	8888.88	7.8518	8.5970	9.6666	8.7051
Ref. (Shakir et al., 2022)	-	-	-	8914.025	-	-	-	3.891629

Table 7.9 shows the proposed method's MAE and SSIM values, and the results are compared with equivalent works discovered in earlier research.

Table 7.9: The Proposed Adaptive Compression and Compressed Sensing

Test Images	MAE (dB) between Original and Encrypted				SSIM
	R	G	B	Average	
Crown-Splash	86.367	86.367	86.367	86.367	0.00587
Mandrill	74.45	74.45	74.45	74.45	0.00930
Peppers	81.913	81.913	81.913	81.913	0.00420
Air-Base	69.193	69.193	69.193	69.193	0.00889
Pepper Ref. (Alexan, Gabr, et al., 2023)	81.9	81.9	81.9	81.9	-----
Pepper Ref. (Yousif et al., 2022)	73.44	73.44	73.44	73.44	0.0415000
Pepper Ref. (Paul et al., 2022)	-----	-----	-----	-----	0.0514000
Pepper Ref. (Elkandoz & Alexan, 2022)	82.18	82.18	82.18	82.18	-----

Adaptively CS Compressed Decryption Metrics

Table 7.10 shows the decryption performance results for adaptive CS compression using three CNN models (VGG16, VGG19, MNV2) across five datasets. The table with the test metrics, such as PSNR, MAE, MSE, UACI, NPCR, and SSIM, calculates the quality of encrypted and decrypted images to the original image. This analysis evaluates each model's decryption efficiency across various datasets.

Decompression Metrics: The findings in Table 7.11 show the decompression results of three CNN models (VGG16, VGG19, and MNV2) using five datasets. All CNN models realized almost the same adaptive compression ratios, approximately 3.750, and a consistent CS compression ratio of 4. The PSNR values for decompressed images were high (around 55 dB), and SSIM values were near perfect (0.9999), indicating excellent image quality retention. MNV2 outperformed the others slightly, especially on the GI-Tract dataset, showing the high-

est PSNR (55.7963 dB) and SSIM. CS decompressed images showed PSNR values between 18 and 21 dB, with MNV2 demonstrating the best performance again. Overall, the results suggest that all models offer strong decompression performance, with MNV2 showing a slight edge (see Table 7.11).

Table 7.10: Image Decryption Results of Adaptively CS Decrypted Image Images

Metrics	Model	Crown-Splash	Mandrill	Peppers	Air-Base	GI-Tract
MSE from Original and Decrypted (dB)	VGG16	765.92	1318.32	782.94	877.15	534.87
	VGG19	764.90	1318.29	782.11	876.89	534.38
	MNV2	767.06	1322.33	785.07	879.12	534.03
MSE from Original and Encrypted (dB)	VGG16	11156.63	8546.97	7975.87	6788.48	12562.54
	VGG19	11312.25	8665.15	10062.79	6771.10	12450.91
	MNV2	11169.23	8552.36	10090.22	6736.68	12448.46
PSNR from Original and Decrypted (dB)	VGG16	19.29	16.93	19.19	18.70	20.85
	VGG19	19.29	16.93	19.20	18.70	20.85
	MNV2	19.28	16.92	19.18	18.69	20.86
PSNR from Original and Encrypted (dB)	VGG16	7.66	8.81	8.09	9.81	7.14
	VGG19	7.60	8.75	8.10	9.82	7.18
	MNV2	7.65	8.81	8.09	9.85	7.18
MAE from Original and Encrypted (dB)	VGG16	86.2633	76.07	82.1267	69.2667	91.78
	VGG19	86.225	76.61	81.80333	69.2433	91.1567
	MNV2	86.3167	76.1167	82.12	68.9033	91.15
NPCR (%)	VGG16	99.6359	99.5403	99.5855	99.6333	99.5775
	VGG19	99.6545	99.5828	99.5137	99.6306	99.5987
	MNV2	99.6173	99.612	99.5908	99.5881	99.5908
UACI	VGG16	33.8288	29.9234	32.2069	27.1639	35.9922
	VGG19	34.0686	30.0429	32.0787	27.1549	35.7469
	MNV2	33.8502	29.8506	32.2026	27.0212	35.7441
SSIM (Original and Decrypted)	VGG16	0.93423	0.68488	0.92404	0.99939	0.92495
	VGG19	0.93439	0.68496	0.92414	0.89697	0.925
	MNV2	0.93417	0.68465	0.92349	0.89603	0.92485
SSIM (Original and Encrypted)	VGG16	0.011107	0.0085801	0.00658	0.20077	0.00077
	VGG19	0.00644	0.00149	0.00643	0.01083	0.00942
	MNV2	0.01050	0.00849	0.00951	0.006831	0.00993

Table 7.11: Decryption Results of Adaptively CS Decompressed and Decrypted Images

Metric	Model	Crown-Splash	Mandrill	Peppers	Air-Base	GI-Tract
Compression Ratio (Adaptive)	VGG16	3.7504	3.7504	3.7503	3.7501	3.7504
	VGG19	3.7503	3.7503	3.7503	3.7502	3.7502
	MNV2	3.7511	3.754	3.7506	3.7508	3.7542
Compression Ratio (CS)	VGG16	4.0000	4.0000	4.0000	4.0000	4.0000
	VGG19	4.0000	4.0000	4.0000	4.0000	4.0000
	MNV2	4.0000	4.0000	4.0000	4.0000	4.0000
PSNR (Decompressed) (dB)	VGG16	55.2058	55.1372	55.2462	55.1037	55.8217
	VGG19	55.207	55.1376	55.2462	55.1037	55.8225
	MNV2	55.2029	55.1223	55.2438	55.0996	55.7963
SSIM (Decompressed)	VGG16	0.99991	0.99987	0.99994	0.99939	0.9989
	VGG19	0.99991	0.99987	0.99994	0.99939	0.9989
	MNV2	0.99991	0.99987	0.99994	0.99939	0.99887
MSE (Decompressed)	VGG16	0.19611	0.19923	0.19430	0.20077	0.17018
	VGG19	0.19606	0.19921	0.19430	0.20077	0.17015
	MNV2	0.19624	0.19992	0.19440	0.20097	0.17118
PSNR (CS Decompressed) (dB)	VGG16	19.4551	18.4061	19.4897	18.6739	21.2751
	VGG19	19.4607	18.4070	19.4897	18.675	21.2805
	MNV2	19.4497	18.3899	19.4778	18.6643	21.2835
SSIM (CS Decompressed)	VGG16	0.93232	0.785	0.93202	0.87963	0.93933
	VGG19	0.93244	0.78509	0.93202	0.8799	0.93938
	MNV2	0.93224	0.78433	0.93166	0.87938	0.93925
MSE (CS Decompressed)	VGG16	737.1664	938.5791	731.328	882.4602	484.8042
	VGG19	736.2207	938.383	731.3276	882.2349	484.2052
	MNV2	738.0945	942.092	733.3267	884.4075	483.8734

7.3.3 Keyspace analysis

The encryption process employed the LSCIM system with parameters z_0 , r_1 , r_2 , and r_3 alongside a memelement-based hyperchaotic system with initial conditions X_0 , Y_0 , Z_0 , H_0 , V_0 , and parameters a , b , d , e , f , g , h , q_1 , q_2 , q_3 , q_4 , and q_5 . Additional encryption keys included padding values M_1 and N_1 and gray value keys $z(0)$ and $z_1(0)$, resulting in a total of 24 unique keys ensuring robust encryption. With a computer precision of 10^{-16} ($< i >$ A Novel Image Encryption Algorithm Based on SPWLCM and DNA Coding $</i >$, n. d.), this configuration results in a total key space of $10^{16 \times 24} = 10^{384} \approx 2^{1276}$, significantly surpassing the 2^{100} benchmark for resisting brute-force attacks (S. Yan et al., 2023). Table 7.12 illustrates the exceptional strength of this method compared to other techniques.

Table 7.12: Keyspace Comparison with Current Research Works

Method	Keyspace value
Minimum required value	2^{100}
This work	2^{1276}
Ref. (S. Zhang & Liu, 2021)	2^{279}
Ref. (Ahuja & Doriya, 2023)	2^{398}
Ref. (Ibrahim et al., 2022)	2^{127}
Ref. (Alexan et al., 2022)	2^{425}

An attacker facing three 1276-bit sub-keys must try 3×2^{1276} combinations (C. M. Kumar et al., 2022). Even if it is processed at the same exponential rate per second, penetrating the key through brute force would take an estimated 7.929×10^{223} years.

$$\frac{3 \times 2^{1276}}{1000 \times 10^6 \times 60 \times 60 \times 24 \times 365} > 3.903 \times 10^{384} \text{ years} \quad (7.27)$$

7.3.4 Key sensitivity test

The proposed encryption method must be sensitive to minor key deviations. Figure 7.13 illustrates that modifying the value of r_1 from 2.5 to 2.5 ± 10^{-16} at the decryption considerably altered the output, underlining the proposed algorithm's security and strength against slight key changes.

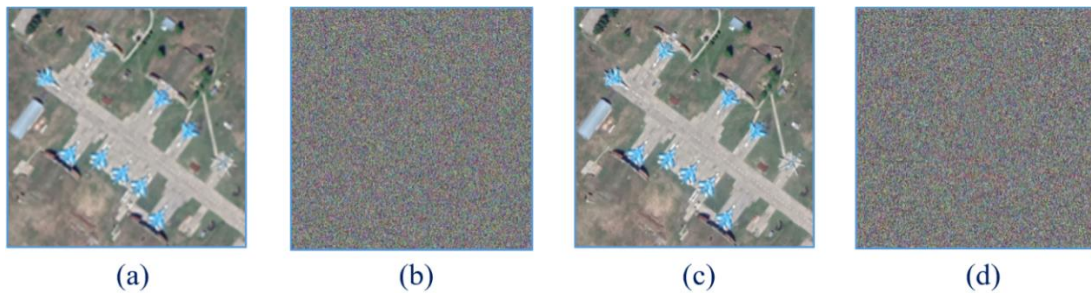


Figure 7.13: Decryption Results Using Appropriate Key and Slightly Modified Key;
(a) Input Test Image, (b) Ciphred Image, (c) Decrypted Image, (d) Decryption Attempted by Slightly Altered Key

7.4 Evaluating Randomness with the NIST Statistical Test Suite

The NIST SP800-22 Statistical Test Suite (STS) was used to assess the randomness of an encrypted image's bitstream, focusing on properties like uniformity, independence, and entropy to evaluate true randomness (Gedda et al., 1977). An analysis of a bitstream in ASCII mode, represented by '0's and '1's, showed that there were either full passes (100%) or complete failures (0%). Table 7.13 indicates that all tests had p-values above the 0.01 threshold, which suggests that there are no patterns in the binary representation of the encrypted image. These findings confirm the effectiveness of the encryption, aligning with the criteria established by (Gedda et al., 1977) and (X. Wu et al., 2025). Additionally, they demonstrate the statistical randomness and cryptographic strength of the bitstream.

Table 7.13: Proposed Framework for NIST STS Results on the Encrypted Image Bitstream

No	Test Name		P-Value	Proportion	Accepted/Rejected
1	Frequency (Monobit) Test		0.858723	100%	Accepted
2	Block Frequency Test		0.143637	100%	Accepted
3	Cumulative Sums (Cusum) Test	Forward Reverse	0.606517 0.483106	100%	Accepted
4	Runs Test		0.896541	100%	Accepted
5	Longest Run of Ones in a Block Test		0.776528	100%	Accepted
6	Rank Test		0.926854	100%	Accepted
7	Discrete Fourier Transform (Spectral) Test (FFT)		0.042557	100%	Accepted
8	Non-overlapping Template Matching Test Average		0.551500	100%	Accepted
9	Overlapping Template Matching Test		0.275457	100%	Accepted
10	Universal Statistical Test (Maurer's)		0.343219	100%	Accepted
11	Approximate Entropy Test		0.265823	100%	Accepted
12	Random Excursions Test Average		0.374200	100%	Accepted
13	Random Excursions Variant Test Average		0.651700	100%	Accepted
14	Serial Test	p_value1 p_value2	0.392719 0.269141	100%	Accepted
15	Linear Complexity Test		0.477492	100%	Accepted

7.5 Performance Validation and Comparative Security Analysis

The experimental results demonstrate the effectiveness of the proposed adaptive compression-encryption framework in achieving high image quality and security performance across various CNN models. The proposed scheme offers a vast key space exceeding 2^{1276} and is highly sensitive to key variations, ensuring strong resistance to brute-force and differential attacks.

Adaptive compression using VGG16, VGG19, and MobileNetV2 consistently produced high PSNR values (~55 dB), near-perfect SSIM (~0.99994), and very low MSE (~0.194). At the same time, MobileNetV2 stood out with the fastest computation time (0.0483 sec), making it ideal for real-time applications. In comparison, the adaptive compressed sensing (CS) approach achieved a higher compression ratio of 4 but at the cost of significantly lower image quality (PSNR ~19.5 dB, SSIM ~0.93, and MSE ~731). Entropy analysis revealed that encrypted images reached near-ideal entropy (~7.996), and correlation values approached zero, confirming strong statistical security. Furthermore, histogram uniformity and high NPCR/UACI values validated the robustness of the encryption. Decryption metrics showed that despite high distortion introduced during encryption, the framework successfully restored image quality with relatively low MSE and high PSNR, particularly with MobileNetV2, affirming its suitability for secure, high-speed image transmission. The evaluation of compression quality, encryption strength, and decryption fidelity across different models is summarized in Table 7.14

Table 7.14: Performance Evaluation Summary of the Proposed Encryption Scheme

Evaluation Aspect	Metric	VGG16	VGG19	MobileNetV2	Compressed Sensing (CS)
Compression Performance	PSNR (dB)	55.2462	55.2462	55.2462	19.4778
	SSIM	0.999946	0.999941	0.999948	0.938212
	MSE	0.19430	0.1943	0.1944	733.3267
	Compression Ratio	3.7503	3.7503	3.7506	4.00
	Processing Time (sec)	0.2336	0.2154	0.04828	0.02837
Encryption Performance	Information Entropy	7.996–7.999	7.996–7.999	7.996–7.999	7.996–7.999
	Correlation (Adjacent Px)	~0.0001	~0.0002	~0.0003	~0.0003
	NPCR (%)	99.61–99.65	99.61–99.65	99.61–99.65	99.61–99.65
	UACI (%)	33.47–33.59	33.47–33.59	33.47–33.59	33.47–33.59
Decryption Recovery	PSNR (dB)	40.22	39.88	41.11	36.04
	MSE	9.12	10.12	8.08	15.19
	SSIM	0.9985	0.9979	0.9987	0.9939
	Encryption Time	0.60708	0.60708	0.37985	Very fast

7.6 Summary

The study in this chapter has proposed a highly effective and secure framework for adaptively compressed image encryption, indicating significant improvement in multimedia processing. The proposed framework optimizes compression efficiency and data security by integrating CNN-based content importance analysis with memelements-fractional-order hyperchaotic encryption techniques. The adaptive segmentation of images into low-importance and high-importance regions, followed by region-specific compression and compressed sensing, approves high-quality image reconstruction with minimum data loss. Metrics such as high PSNR (above 55 dB) and perfect SSIM values (≈ 0.9999) validate the framework's compression performance. On the security front, integrating fractional-order hyperchaotic systems, DNA encoding, and dynamic diffusion enhances resistance to differential, statistical, and brute-force attacks. The method realized an average NPCR of 99.61%, showing high randomness and strong resistance to differential attacks. Additionally, an average UACI computed as 33.83% proves its effectiveness in significantly changing pixel intensities during encryption, ensuring strong security. The algorithm's Keyspace of 2^{1276} ensures computational Impossibility for brute-force attempts, while key sensitivity analysis shows robustness against slight parameter changes. Experimental results across various datasets tell the efficiency of MobileNetV2 in decreasing computation time while preserving high image quality, making it appropriate for real-time applications. The suggested framework enhances security and compression, making it ideal for private, healthcare, military, and secure communication applications. Future research may explore video processing and real-time applications on resource-limited devices.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

This dissertation proposed a unified approach to advanced image encryption by integrating chaos theory, biologically inspired mechanisms, and artificial intelligence. This research has addressed the growing demand for secure and intelligent encryption systems in dynamic and sensitive domains such as 5G networks, defense systems, healthcare, and cloud-based platforms. Traditional cryptographic techniques like AES, DES, and RSA, while sufficient for general-purpose encryption, are insufficient for image data due to their lack of adaptability and structural inefficiency. In contrast, the presented frameworks' complex hyperchaotic behaviour and AI adaptability give robust, lightweight, and real-time encryption mechanisms.

This study has discussed five innovative methodologies, each contributing to a stepwise advancement in image security. The proposed approaches incorporated mem-element-based hyperchaotic systems, DNA-inspired coding schemes, hybrid chaotic maps, and deep learning techniques, including Convolutional Neural Networks (CNNs) and Online Sequential Extreme Learning Machine (OSELM) autoencoders. Extensive validation has been performed to ensure practical applicability using image datasets from the USC-SIPI and publicly available Kaggle repositories. These datasets span a variety of resolutions and content types, effectively simulating diverse real-world scenarios. The results demonstrate the robustness, adaptability, and effectiveness of the proposed encryption schemes in addressing contemporary security challenges in image processing.

The first methodology of this study introduced six integer-order hyperchaotic systems designed using memelements to enhance multistability, parameter sensitivity, and chaotic complexity for secure communication applications. These systems include configurations based on memristors, memcapacitors, and meminductors, each engineered to generate high-dimensional chaotic behavior validated through Lyapunov exponents, bifurcation diagrams, and phase portraits. Notable models include the foundational Memristor-Based Hyperchaotic System, Dual Memristor-Based and Band Pass Filter-augmented variants, Memcapacitor-Based Hyperchaotic System, and hybrid architectures combining memcapacitive and meminductive dynamics. These systems exhibit features such as multi-scroll attractors, tunable chaos through key parameters, and bidirectional energy coupling, with the most advanced model achieving stabilized hyperchaotic behavior over time. Thorough numerical simulations using Runge-

Kutta methods demonstrate their reliability. Their low computational overhead and sensitivity to initial conditions highlight the effectiveness of image encryption. These systems create a hardware-friendly, high-entropy foundation for future secure image transmission frameworks.

The second proposed method introduced a novel encryption algorithm that synergistically combines memcapacitor-based hyperchaotic systems with logistic mapping and DNA coding rules. This framework demonstrated strong resistance to differential and statistical attacks, achieving NPCR values exceeding 99.6% and UACI values around 33%, confirming its robustness for secure image transmission.

The third proposed method optimised a four-dimensional memristor-based hyperchaotic system using Chaotic Particle Swarm Optimisation (CPSO). The CPSO algorithm effectively enhanced the system's chaotic behaviour and unpredictability, resulting in entropy values approaching 7.9994 and a notably ample key space (greater than 2^{1116}), ensuring resilience against brute-force attacks.

In the fourth proposed method, a joint encryption and denoising model was introduced using Online Sequential Extreme Learning Machine (OSELM) autoencoders. This framework simultaneously performed encryption and noise suppression, enhancing robustness in degraded environments. Experimental results demonstrated strong denoising capabilities, with high structural similarity (SSIM $\approx$ 0.999) and improved mean squared error (MSE) values,

Finally, the fifth proposed method introduced an adaptive compression-encryption framework that combines Convolutional Neural Networks (CNNs) for importance map extraction with a hybrid chaotic encryption scheme. This approach achieves high compression ratios of up to 3.75 while preserving image quality (PSNR > 55 dB), improving processing speed and storage efficiency for real-time and bandwidth-limited applications. A fractional-order mem-element-based hyperchaotic system is incorporated to enhance security and efficiency further, utilizing DNA coding and Logistic Sine Cubic Integrated Mapping (LSCIM). Optimized for lightweight performance through architectures such as MobileNetV2, the integrated model delivers up to 40% faster execution without compromising cryptographic robustness.

Each proposed method was rigorously evaluated using industry-standard metrics: entropy, NPCR, UACI, PSNR, SSIM, MSE, and correlation coefficients. The outcomes confirm the frameworks' robustness, efficiency, and adaptability across encryption scenarios. All models

were designed to meet the demands of modern, resource-constrained, and real-time environments.

This study successfully established innovative frameworks that advance the frontiers of secure image transmission. The proposed methods enhance visual quality, robustness, and computational efficiency by leveraging chaotic dynamics, bio-inspired coding, and artificial intelligence. These contributions hold significant promise for applications in medical imaging, military communications, digital forensics, and remote sensing, serving as a foundation for the next generation of intelligent image security systems.

Table 8.1 summarizes the proposed methods' superior NPCR, UACI, entropy, keyspace, and execution times, showing improvements over traditional image encryption techniques.

Table 8.1: Proposed methods comparison

S.No	Method	NPCR	UACI	Keyspace	Entropy	Running time(E+D)
1	Image Encryption Using A Memelement-Based Hyperchaotic System Integrated With DNA Encoding Schemes	99.587% – 99.634%	27.474% – 33.935%	2^{531}	Up to 7.9998	1.4636871 seconds
2	Enhancing Image Encryption Through Parameters Optimization of Memristive Hyperchaotic System With CPSO Algorithm And LSAIM	99.605 – 99.631 (Avg: 99.6178)	28.98 – 33.96 (Avg: 32.26)	2^{1116}	7.9992 – 7.9994 ≈ 8	0.182413 seconds (E time)
3	Denoising Encrypted Images Using Online Sequential Extreme Learning Machine With A Memelements-Based Hyperchaotic System	99.608 – 99.631 (Avg: 99.617)	28.32 – 33.69 (Avg: 31.37)	2^{797}	7.9992 – 7.9996 ≈ 8	1.506313 seconds (E time)
4	An Adaptive Deep CNN-Based Compression And Hybrid Integer-Fractional Order Hyperchaotic Encryption Framework For Secure Image Transmission	99.61–99.65 99.61%(avg)	33.47–33.59 33.83%(avg)	2^{1276}	7.996–7.999	0.37985 seconds (E time) MNV2

The research contributions effectively address significant limitations of traditional image encryption techniques by integrating advanced chaotic systems, optimization algorithms, deep learning, and compression strategies. Traditional chaos-based encryption methods often suffer

from limited Keyspace, suboptimal sensitivity, and inefficiencies in execution time (Wang et al., 2018; Li et al., 2017). In contrast, the proposed memristor-based hyperchaotic system with DNA encoding achieved a non-permutation correlation rate (NPCR) of 99.587% to 99.634% and entropy values up to 7.9998, greatly enhancing resistance to differential and statistical attacks. Building on this, the CPSO-LSAIM optimized memristive scheme addressed gaps in Keyspace and speed, achieving an exponential Keyspace of (2^{1116}) , a high average NPCR of 99.6178%, and an average unified average change intensity (UACI) of 32.26%, all with a notably low encryption time of just 0.18 seconds. This performance surpasses many optimization-free designs (El-Latif et al., 2019). Furthermore, the OSELM-based encrypted image denoising framework, with an average NPCR of 99.617%, introduces resilience against channel noise—an aspect often overlooked in previous works (Zhou et al., 2020). Finally, the Adaptive Deep CNN-Based Compression and Hybrid Integer–Fractional Order Hyperchaotic Encryption Framework significantly enhances both efficiency and security in image encryption. This method achieved an impressive NPCR range of 99.61% to 99.65% (average: 99.61%) and a UACI between 33.47% and 33.59% (average: 33.83%), surpassing many traditional schemes that struggle to manage redundancy and prevent differential attack resistance effectively. The system ensures strong key sensitivity and randomness with a key space of 21276 and entropy values reaching up to 7.999. Additionally, the encryption time of only 0.37985 seconds demonstrates high computational efficiency. Unlike earlier algorithms (e.g., Li et al., 2017; El-Latif et al., 2019), which often lacked adaptability and robust compression, this framework successfully integrates deep learning-based compression with secure hyperchaotic dynamics, particularly utilising integer–fractional order modelling. This addresses the critical gap in achieving compression and encryption for secure, real-time image transmission.

8.2 Recommendations

Based on the determinations of this study investigation, the following recommendations are suggested for coming research and development:

- **Real-World Application Testing:** Future investigations should consider encryption frameworks in live environments, such as surveillance feeds and telemedicine platforms, to evaluate the performance and adaptability of the developed encryption algorithms.

- **Quantum-Resistant Design:** As quantum computing grows, it is essential to design encryption techniques that withstand quantum algorithms by combining post-quantum secure key management and lattice-based methods.
- **Energy Efficiency:** Research should concentrate on decreasing the computational complexity of chaotic and AI-driven systems.
- **Multimedia Encryption:** Extending current frameworks to contain video encryption, 3D data, and multimodal medical imaging is critical, focusing on synchronization and efficiency.
- **Blockchain Integration:** Integrating these encryption methods with blockchain and federated learning systems can improve data security and traceability.
- **Multi-Chaotic Exploration:** Investigating the integration of different chaotic systems may enhance entropy and resistance to new attack strategies.

RESEARCH CONTRIBUTION

1. Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “Online sequential Extreme learning Machine (OSELM) based denoising of encrypted image”, **Expert Systems with Applications**, vol. 274, no. February, p. 126999, 2025, **Impact Factor: 7.5**, SCIE, ELSEVIER, Doi: 10.1016/j.eswa.2025.126999, **Status: Published**.
2. Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “Enhancing colour image encryption through parameters optimization of memristive hyperchaotic system with CPSO algorithm and LSAIM” **The Imaging Science Journal**, no. January, pp. 1–21, 2025, Taylor & Francis, **SCIE**, **Impact Factor: 1.4**, Doi: 10.1080/13682199.2025.2454055, **Status: Published**.
3. Authors: Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “Image encryption algorithm based on a memcapacitor-based hyperchaotic system and DNA coding”, **Security and Privacy (Wiley)**, June, 2024, **(Clarivate) Impact Factor 1.5**, Taylor & Francis, Web of Science, ESCI, Doi: 10.1002/spy2.432, **Status: Published**.
4. Authors: B. A. Belete, D. J. Gelmecha, R. S. Singh, D. S. Rathee, A. Arora, and H. M. Besfat, “A Novel Color Image Encryption using DNA Code and Dual Memristor Hyperchaotic System”, IEEE Conference (Scopus), February, pp. 1469–1477, 2024, Doi: 10.23919/INDIACom61295.2024.10498928, **Status: Published**.
5. Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “A Novel Image Encryption Framework Integrating OSELM Autoencoder, Hyperchaotic Systems, and DNA Coding for Enhanced Security and Noise Resilience”, **The 5th International Research Symposium** on Stepping Towards Sustainability through Circular Economy: ‘Waste into Resource’, **Status: Accepted**.
6. Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “Secure Image Transmission in Wireless Sensor Networks Using Memcapacitive Hyperchaotic System and DNA Code for Encryption and Decryption”, **Iran Journal of Computer Science**, Springer Nature (Scopus), **CiteScore: 2.743**, **Status: Under Review**.
7. Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “A Hybrid Deep Convolutional Neural Network Based Adaptive Compression and Memelements Hyperchaotic Encryption Framework for Secure Image Transmission”, **Engineering Applications of Artificial Intelligence**, SCIE, **Impact Factor (2023): 7.5**, **Status: Under Review**.
8. Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “A Survey on Image Security through Memelements-Based Chaotic and Hyperchaotic Systems: A Literature Review”, **Status: Under Process**.
9. Biniyam Ayele Belete, Demissie Jobir Gelmecha ,Ram Sewak Singh, “Secure Image Transmission Using Adaptive CNN-Based Compression and Fractional-Order Memelements-Based Hyperchaotic Encryption with LSCIM and DNA Coding”, **Status: Under Process**.

REFERENCES

- A Novel Image Encryption Algorithm Based on SPWLCM and DNA Coding, 190 723–744. <https://doi.org/10.1016/j.matcom.2021.06.012>
- Abd-El-Atty, B., & Abd EL-Latif, A. A. (2023). Applicable image cryptosystem using bit-level permutation, particle swarm optimisation, and quantum walks. *Neural Computing and Applications*, 35(25), 18325–18341. <https://doi.org/10.1007/s00521-023-08643-5>
- Abro, K. A., Siyal, A., & Atangana, A. (2023). Strange Fractal Attractors and Optimal Chaos of Memristor–Memcapacitor via Non-local Differentials. *Qualitative Theory of Dynamical Systems*, 22(4), 1–18. <https://doi.org/10.1007/s12346-023-00849-1>
- Aditya, K., Mohanty, A. K., Ragav, G. A., Thanikaiselvan, V., & Amirtharajan, R. (2020). Image encryption using dynamic DNA encoding and pixel scrambling using composite chaotic maps. *IOP Conference Series: Materials Science and Engineering*, 872(1). <https://doi.org/10.1088/1757-899X/872/1/012045>
- Ahmad, J., & Ahmed, F. (2012). *Efficiency Analysis and Security Evaluation of Image Encryption Schemes*. 04.
- Ahuja, B., & Doriya, R. (2023). A secure algorithm using high-dimensional sine map for color image encryption. *International Journal of Information Technology (Singapore)*, 15(3), 1535–1543. <https://doi.org/10.1007/s41870-023-01190-1>
- Akgül, A., Rajagopal, K., Durdu, A., Pala, M. A., Boyraz, Ö. F., & Yildiz, M. Z. (2021). A simple fractional-order chaotic system based on memristor and memcapacitor and its synchronization application. *Chaos, Solitons and Fractals*, 152. <https://doi.org/10.1016/j.chaos.2021.111306>
- Akif, O. Z., Ali, S. M., Ali, R. S., & Farhan, A. K. (2021). A new pseudorandom bits generator based on a 2d-chaotic system and diffusion property. *Bulletin of Electrical Engineering and Informatics*, 10(3), 1580–1588. <https://doi.org/10.11591/eei.v10i3.2610>
- Al-Amri, R. M., Hamood, D. N., & Farhan, A. K. (2023). Theoretical Background of Cryptography. *Mesopotamian Journal of CyberSecurity*, 2023, 7–15. <https://doi.org/10.58496/MJCS/2023/002>
- Al-Dhief, F. T., Baki, M. M., Latiff, N. M. A. A., Malik, N. N. N. A., Salim, N. S., Albader, M. A. A., Mahyuddin, N. M., & Mohammed, M. A. (2021). Voice Pathology Detection and Classification by Adopting Online Sequential Extreme Learning Machine. *IEEE Access*, 9, 77293–77306. <https://doi.org/10.1109/ACCESS.2021.3082565>
- Alawida, M. (2024). A Novel Image Encryption Algorithm Based on Cyclic Chaotic Map in Industrial IoT Environments. *IEEE Transactions on Industrial Informatics*, 20(8), 10530–10541. <https://doi.org/10.1109/TII.2024.3395631>
- Alawida, M., & Teh, J. Sen. (2023). *A New Image Encryption Algorithm Based on DNA State Machine for UAV Data Encryption*. 1–24.
- Alegro, J. K. P., Arboleda, E. R., Pereña, M. R., & Dellosa, R. M. (2019). Hybrid schnorr, rsa, and aes cryptosystem. *International Journal of Scientific and Technology Research*, 8(10), 1770–1776.
- Alexan, W., Aly, L., Korayem, Y., Gabr, M., El-Damak, D., Fathy, A., & Mansour, H. A. A. (2024). Secure Communication of Military Reconnaissance Images Over UAV-Assisted

- Relay Networks. *IEEE Access*, 12(June), 78589–78610. <https://doi.org/10.1109/ACCESS.2024.3407838>
- Alexan, W., Chen, Y. L., Por, L. Y., & Gabr, M. (2023). Hyperchaotic Maps and the Single Neuron Model: A Novel Framework for Chaos-Based Image Encryption. *Symmetry*, 15(5). <https://doi.org/10.3390/sym15051081>
- Alexan, W., Elbeltagy, M., & Aboshousha, A. (2022). RGB Image Encryption through Cellular Automata, S-Box and the Lorenz System. *Symmetry*, 14(3), 1–25. <https://doi.org/10.3390/sym14030443>
- Alexan, W., Elkandoz, M., Mashaly, M., Azab, E., & Aboshousha, A. (2023). Color Image Encryption Through Chaos and KAA Map. *IEEE Access*, 11(October 2022), 11541–11554. <https://doi.org/10.1109/ACCESS.2023.3242311>
- Alexan, W., Gabr, M., Mamdouh, E., Elias, R., & Aboshousha, A. (2023). Color Image Cryptosystem Based on Sine Chaotic Map, 4D Chen Hyperchaotic Map of Fractional-Order and Hybrid DNA Coding. *IEEE Access*, 11(March), 54928–54956. <https://doi.org/10.1109/ACCESS.2023.3282160>
- Alhudhaif, A., Ahmad, M., Alkhayyat, A., Tsafack, N., Farhan, A. K., & Ahmed, R. (2021). Block Cipher Nonlinear Confusion Components Based on New 5-D Hyperchaotic System. *IEEE Access*, 9(June), 87686–87696. <https://doi.org/10.1109/ACCESS.2021.3090163>
- Alibraheemi, H. M. M., Al-Gayem, Q., & Hussein, E. A. R. (2022). Four dimensional hyperchaotic communication system based on dynamic feedback synchronization technique for image encryption systems. *International Journal of Electrical and Computer Engineering*, 12(1), 957–965. <https://doi.org/10.11591/ijece.v12i1.pp957-965>
- Alizamir, M., Ahmed, K. O., Shiri, J., Fard, A. F., Kim, S., Heddami, S., & Kisi, O. (2023). A New Insight for Daily Solar Radiation Prediction by Meteorological Data Using an Advanced Artificial Intelligence Algorithm : Deep Extreme Learning Machine Integrated with Variati ... A New Insight for Daily Solar Radiation Prediction by Meteorological . *July*. <https://doi.org/10.3390/su151411275>
- Alizamir, M., Othman Ahmed, K., Shiri, J., Fakheri Fard, A., Kim, S., Heddami, S., & Kisi, O. (2023). A New Insight for Daily Solar Radiation Prediction by Meteorological Data Using an Advanced Artificial Intelligence Algorithm: Deep Extreme Learning Machine Integrated with Variational Mode Decomposition Technique. *Sustainability (Switzerland)*, 15(14). <https://doi.org/10.3390/su151411275>
- Almatroud Othman, A., Noorani, M. S. M., & Al-sawalha, M. M. (2016). Adaptive dual synchronization of chaotic and hyperchaotic systems with fully uncertain parameters. *Optik*, 127(19), 7852–7864. <https://doi.org/10.1016/j.ijleo.2016.05.139>
- Alqahtani, A., Xie, X., Jones, M. W., & Essa, E. (2021). Pruning CNN filters via quantifying the importance of deep visual representations. *Computer Vision and Image Understanding*, 208–209(July 2020), 103220. <https://doi.org/10.1016/j.cviu.2021.103220>
- Alruily, M., Shahin, O. R., Al-Mahdi, H., & Taloba, A. I. (2021). Asymmetric DNA encryption and decryption technique for Arabic plaintext. *Journal of Ambient Intelligence and Humanized Computing*, 0123456789. <https://doi.org/10.1007/s12652-021-03108-w>
- Amani, H. R., & Yaghoobi, M. (2019). A New Approach in Adaptive Encryption Algorithm

- for Color Images Based on DNA Sequence Operation and Hyper-Chaotic System. *Multimedia Tools and Applications*, 78(15), 21537–21556. <https://doi.org/10.1007/s11042-018-6989-y>
- An, X., Jiang, M., Deng, W., & Fang, J. (2021). A novel dual memristor hyperchaotic system and its application for secure communication based on three-fold function projection synchronization. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 34(2), 1–19. <https://doi.org/10.1002/jnm.2825>
- Aouissaoui, I., Bakir, T., Sakly, A., & Femmam, S. (2022). Improved one-dimensional piecewise chaotic maps for information security. *Journal of Communications*, 17(1), 11–16. <https://doi.org/10.12720/jcm.17.1.11-16>
- Arshad, U., Batool, S. I., & Amin, M. (2019). A Novel Image Encryption Scheme Based on Walsh Compressed Quantum Spinning Chaotic Lorenz System. *International Journal of Theoretical Physics*, 58(10), 3565–3588. <https://doi.org/10.1007/s10773-019-04221-5>
- Arthi, G., Thanikaiselvan, V., & Amirtharajan, R. (2022). 4D Hyperchaotic map and DNA encoding combined image encryption for secure communication. *Multimedia Tools and Applications*, 81(11), 15859–15878. <https://doi.org/10.1007/s11042-022-12598-5>
- Bakour, K., Bakir, R., & Adem, K. (2024). 4th International Artificial Intelligence and Data Science Congress A Deep Learning-Based Approach for Image Denoising : Harnessing Autoencoders for Removing Gaussian and Salt-Pepper Noises. March.
- Banerjee, S., & Shanmugam, P. (2022). An Improved Method for Destriping of VIIRS Day/Night Band Images. *IEEE Access*, 10(July), 82164–82184. <https://doi.org/10.1109/ACCESS.2022.3194053>
- Belete, B. A., Gelmecha, D. J., & Singh, R. S. (2024). Image encryption algorithm based on a memcapacitor-based hyperchaotic system and DNA coding . *Security and Privacy*, June. <https://doi.org/10.1002/spy2.432>
- Belete, B. A., Gelmecha, D. J., & Singh, R. S. (2025a). Enhancing colour image encryption through parameters optimization of memristive hyperchaotic system with CPSO algorithm and LSAIM. *The Imaging Science Journal*, January, 1–21. <https://doi.org/10.1080/13682199.2025.2454055>
- Belete, B. A., Gelmecha, D. J., & Singh, R. S. (2025b). Online sequential Extreme learning Machine (OSELM) based denoising of encrypted image. *Expert Systems with Applications*, 274(February), 126999. <https://doi.org/10.1016/j.eswa.2025.126999>
- Belete, B. A., Gelmecha, D. J., Singh, R. S., Rathee, D. S., Arora, A., & Besfat, H. M. (2024). A Novel Color Image Encryption using DNA Code and Dual Memristor Hyperchaotic System. *Proceedings of the 18th INDIACom; 2024 11th International Conference on Computing for Sustainable Global Development, INDIACom 2024, February*, 1469–1477. <https://doi.org/10.23919/INDIACom61295.2024.10498928>
- Bencherqui, A., Amine Tahiri, M., Karmouni, H., Alfidi, M., Motahhir, S., Abouhawwash, M., Askar, S. S., Wen, S., Qjidaa, H., & Sayyouri, M. (2024). Optimal algorithm for color medical encryption and compression images based on DNA coding and a hyperchaotic system in the moments. *Engineering Science and Technology, an International Journal*, 50(December 2023), 101612. <https://doi.org/10.1016/j.jestch.2023.101612>
- Benkouider, K., Sambas, A., Bonny, T., Al Nassan, W., Moghrabi, I. A. R., Sulaiman, I. M.,

- Hassan, B. A., & Mamat, M. (2024). A comprehensive study of the novel 4D hyperchaotic system with self-exited multistability and application in the voice encryption. *Scientific Reports*, 14(1), 1–14. <https://doi.org/10.1038/s41598-024-63779-1>
- Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., & Xu, Y. (2024). Autoencoders and their applications in machine learning: a survey. In *Artificial Intelligence Review* (Vol. 57, Issue 2). Springer Netherlands. <https://doi.org/10.1007/s10462-023-10662-6>
- Bhardwaj, K., & Srivastava, M. (2022). New Multiplier-Less Compact Tunable Charge-Controlled Memelement Emulator Using Grounded Passive Elements. *Circuits, Systems, and Signal Processing*, 41(5), 2429–2465. <https://doi.org/10.1007/s00034-021-01895-3>
- Bonny, T., Nassan, W. Al, & Baba, A. (2023). Voice encryption using a unified hyper-chaotic system. *Multimedia Tools and Applications*, 82(1), 1067–1085. <https://doi.org/10.1007/s11042-022-13317-w>
- Borah, M., & Roy, B. K. (2021). Hidden multistability in four fractional-order memristive, meminductive and memcapacitive chaotic systems with bursting and boosting phenomena. *European Physical Journal: Special Topics*, 230(7–8), 1773–1783. <https://doi.org/10.1140/epjs/s11734-021-00179-w>
- Brabin, D., Ananth, C., & Bojjagani, S. (2022). Blockchain based security framework for sharing digital images using reversible data hiding and encryption. *Multimedia Tools and Applications*, 81(17), 24721–24738. <https://doi.org/10.1007/s11042-022-12617-5>
- Cai, J., Xie, S., & Zhang, J. (2023). Image compression-encryption algorithm based on chaos and compressive sensing. *Multimedia Tools and Applications*, 82(14), 22189–22212. <https://doi.org/10.1007/s11042-022-13346-5>
- Cai, Q., Li, H., Cao, W., & Guifen Jie. (2021). Photoinduced-electron transfer coupled with DNA cross-chain displacement multiple amplification for fluorescence biosensing of MicroRNA. *Analytica Chimica Acta*, 1148, 238169. <https://doi.org/10.1016/j.aca.2020.12.044>
- Çam Taşkıran, Z. G., Sağbaş, M., Ayten, U. E., & Sedef, H. (2020). A new universal mutator circuit for memcapacitor and meminductor elements. *AEU - International Journal of Electronics and Communications*, 119. <https://doi.org/10.1016/j.aeue.2020.153180>
- Chai, X., Bi, J., Gan, Z., Liu, X., Zhang, Y., & Chen, Y. (2020). Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. *Signal Processing*, 176, 107684. <https://doi.org/10.1016/j.sigpro.2020.107684>
- Chai, X., Chen, Y., & Broyde, L. (2017). A novel chaos-based image encryption algorithm using DNA sequence operations. *Optics and Lasers in Engineering*, 88, 197–213. <https://doi.org/10.1016/j.optlaseng.2016.08.009>
- Chaotic, S. P. S. O., & Scheme, I. E. (2022). *Single-Objective Particle Swarm Optimization-Based Chaotic Image Encryption Scheme*.
- Charmouti, B., Kadri, A., & Achmad, J. (2022). A new denoising method for removing salt & pepper noise from image. *Multimedia Tools and Applications*, 3981–3993. <https://doi.org/10.1007/s11042-021-11615-3>
- Chen, B., Cheng, X., Bao, H., Chen, M., & Xu, Q. (2022). Extreme Multistability and Its Incremental Integral Reconstruction in a Non-Autonomous Memcapacitive Oscillator. *Mathematics*, 10(5). <https://doi.org/10.3390/math10050754>

- Chen, C., Sun, K., & Xu, Q. (2020). *A Color Image Encryption Algorithm Based on 2D-CIMM Chaotic Map*. May, 12–20.
- Chen, G. (1999). *Unknown. Unknown. YET ANOTHER CHAOTIC ATTRACTOR.pdf.pdf*. 9(7), 1465–1466.
- Chen, J., Chen, L., & Zhou, Y. (2020). Cryptanalysis of a DNA-based image encryption scheme. *Information Sciences*, 520, 130–141. <https://doi.org/10.1016/j.ins.2020.02.024>
- Chen, W. (n.d.). *Explore the potential of deep learning and hyperchaotic map in the meaningful visual image encryption scheme*. <https://doi.org/10.1049/ipr2.12858>
- Chen, X., Mou, J., Cao, Y., Yan, H., & Jahanshahi, H. (2023). A chaotic color image encryption scheme based on improved Arnold scrambling and dynamic DNA encoding. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-023-14826-y>
- Chen, Z., Wu, J., Gan, W., & Qi, Z. (2022). Metaverse Security and Privacy: An Overview. *Proceedings - 2022 IEEE International Conference on Big Data, Big Data 2022*, 2950–2959. <https://doi.org/10.1109/BigData55660.2022.10021112>
- Cheng, G., Wang, C., & Xu, C. (2020). A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing. *Multimedia Tools and Applications*, 79(39–40), 29243–29263. <https://doi.org/10.1007/s11042-020-09542-w>
- Chua, L. O. (1971). Memristor—The Missing Circuit Element. *IEEE Transactions on Circuit Theory*, 18(5), 507–519. <https://doi.org/10.1109/TCT.1971.1083337>
- Chua, L. O. (2018). How we predicted the memristor. *Nature Electronics*, 1(5), 322. <https://doi.org/10.1038/s41928-018-0074-4>
- Cryptography, C., & Lawnik, M. (2022). *SS symmetry New Chaotic System : M-Map and Its Application in*. 1–18.
- Cun, Q., Tong, X., Wang, Z., & Zhang, M. (2023). A new chaotic image encryption algorithm based on dynamic DNA coding and RNA computing. *Visual Computer*. <https://doi.org/10.1007/s00371-022-02750-5>
- Dai, Y., Gao, Y., & Liu, F. (2021). Transmed: Transformers advance multi-modal medical image classification. *Diagnostics*, 11(8), 1–15. <https://doi.org/10.3390/diagnostics11081384>
- Dantas, P. V., Sabino, W., Lucas, J., Cordeiro, C., & Barbosa, C. (2024). *A comprehensive review of model compression techniques in machine learning*. 11804–11844. <https://doi.org/10.1007/s10489-024-05747-w>
- Devi, A., Sharma, A., & Rangra, A. (2015a). A Review on DES, AES and Blowfish for Image Encryption & Decryption. *International Journal Of Engineering And Computer Science*, 4(6), 12646–12651. <https://www.ijecs.in/index.php/ijecs/article/download/3887/3623/>
- Devi, A., Sharma, A., & Rangra, A. (2015b). A Review on DES, AES and Blowfish for Image Encryption & Decryption. *International Journal Of Engineering And Computer Science*, 4(6), 12646–12651.
- Dhanalaxmi, B. (2017). *M ultimedia C ryptography - A R eview*. 2017 *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 764–766.

- Di Ventra, M., Pershin, Y. V., & Chua, L. O. (2009). Circuit elements with memory: Memristors, memcapacitors, and meminductors. *Proceedings of the IEEE*, 97(10), 1717–1724. <https://doi.org/10.1109/JPROC.2009.2021077>
- Dong, H., Bai, E., Jiang, X. Q., & Wu, Y. (2020). Color image compression-encryption using fractional-order hyperchaotic system and DNA coding. *IEEE Access*, 8, 163524–163540. <https://doi.org/10.1109/ACCESS.2020.3022398>
- Ejaz, M. Z., Khurshid, K., Abbas, Z., Aizaz, M. A., & Nawaz, A. (2018). A novel image encoding and communication technique of B/W images for IOT, robotics and drones using (15, 11) reed solomon scheme. *2018 Advances in Science and Engineering Technology International Conferences, ASET 2018*, 1, 1–6. <https://doi.org/10.1109/ICASET.2018.8376846>
- El-Shafai, W., Khallaf, F., El-Rabaie, E. S. M., & El-Samie, F. E. A. (2021). Robust medical image encryption based on DNA-chaos cryptosystem for secure telemedicine and healthcare applications. *Journal of Ambient Intelligence and Humanized Computing*, 12(10), 9007–9035. <https://doi.org/10.1007/s12652-020-02597-5>
- El-Shorbagy, M. A., Eldesoky, I. M., Basyouni, M. M., Nassar, I., & El-Refaey, A. M. (2022). Chaotic Search-Based Salp Swarm Algorithm for Dealing with System of Nonlinear Equations and Power System Applications. *Mathematics*, 10(9). <https://doi.org/10.3390/math10091368>
- Elkandoz, M. T., & Alexan, W. (2022). Image encryption based on a combination of multiple chaotic maps. *Multimedia Tools and Applications*, 81(18), 25497–25518. <https://doi.org/10.1007/s11042-022-12595-8>
- Elsaid, S. A., & Alotaibi, E. R. (2023). *A robust hybrid cryptosystem based on DNA and Hyperchaotic for images encryption*. 1995–2019.
- Erkan, U., Toktas, A., Enginoğlu, S., & Akbacak, E. (2022). An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN. *Multimedia Tools and Applications*, 7365–7391. <https://doi.org/10.1007/s11042-021-11803-1>
- Ersoy, D., & Kacar, F. (2023). Electronically Charge-Controlled Tunable Meminductor Emulator Circuit with OTAs and its Applications. *IEEE Access*, PP, 1. <https://doi.org/10.1109/ACCESS.2023.3281200>
- Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24–29. <https://doi.org/10.1038/s41591-018-0316-z>
- Fadhil, M. S., Farhan, A. K., & Fadhil, M. N. (2021). Designing Substitution Box Based on the 1D Logistic Map Chaotic System. *IOP Conference Series: Materials Science and Engineering*, 1076(1), 012041. <https://doi.org/10.1088/1757-899x/1076/1/012041>
- Fan, T. (2024). New forms of encryption art creation and reception in the context of virtual digital from the perspective of metauniverse. *Proceedings - 2024 International Conference on Culture-Oriented Science and Technology, CoST 2024*, 395–399. <https://doi.org/10.1109/CoST64302.2024.00084>
- Fang, P., Liu, H., & Wu, C. (2021). A novel chaotic block image encryption algorithm based on deep convolutional generative adversarial networks. *IEEE Access*, 9, 18497–18517. <https://doi.org/10.1109/ACCESS.2020.3040573>

- Fang, P., Liu, H., Wu, C., & Liu, M. (2022). A survey of image encryption algorithms based on chaotic system. *Visual Computer*, 3(1). <https://doi.org/10.1007/s00371-022-02459-5>
- Feng, W., Wang, Q., Liu, H., Ren, Y., Zhang, J., Zhang, S., & Qian, K. (2023). *Exploiting Newly Designed Fractional-Order 3D Lorenz Chaotic System and 2D Discrete Polynomial Hyper-Chaotic Map for High-Performance Multi-Image Encryption*. 1–30.
- Feng, W., Zhao, X., Zhang, J., Qin, Z., & Zhang, J. (2022). *Image Encryption Algorithm Based on Plane-Level Image Filtering and Discrete Logarithmic Transform*. 1–24.
- Gabr, M., Korayem, Y., Chen, Y. L., Yee, P. L., Ku, C. S., & Alexan, W. (2023). R3 - Rescale, Rotate, and Randomize: A Novel Image Cryptosystem Utilizing Chaotic and Hyper-Chaotic Systems. *IEEE Access*, 11(October), 119284–119312. <https://doi.org/10.1109/ACCESS.2023.3326848>
- Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. In *Archives of Computational Methods in Engineering* (Vol. 29, Issue 5). Springer Netherlands. <https://doi.org/10.1007/s11831-021-09694-4>
- Gao, H., & Wang, X. (2021). Chaotic Image Encryption Algorithm Based on Zigzag Transform with Bidirectional Crossover from Random Position. *IEEE Access*, 9(Icmic), 105627–105640. <https://doi.org/10.1109/ACCESS.2021.3099214>
- Gao, X. (2021). A color image encryption algorithm based on an improved Hénon map. *Physica Scripta*, 96(6), 12–20. <https://doi.org/10.1088/1402-4896/abed7d>
- Gao, X., Mou, J., Xiong, L., Sha, Y., Yan, H., & Cao, Y. (2022). A fast and efficient multiple images encryption based on single-channel encryption and chaotic system. *Nonlinear Dynamics*, 108(1), 613–636. <https://doi.org/10.1007/s11071-021-07192-7>
- Gao, X., Yu, J., Banerjee, S., Yan, H., & Mou, J. (2021). A new image encryption scheme based on fractional-order hyperchaotic system and multiple image fusion. *Scientific Reports*, 11(1), 1–21. <https://doi.org/10.1038/s41598-021-94748-7>
- Gao, Y., Liu, J., & Chen, S. (2024). Image encryption algorithms based on two-dimensional discrete hyperchaotic systems and parallel compressive sensing. *Multimedia Tools and Applications*, 83(19), 57139–57161. <https://doi.org/10.1007/s11042-023-17745-0>
- Gao, Y., Liu, J., Zhang, B., & Chen, S. (2024). Image encryption algorithm based on four-dimensional memristor hyperchaotic system and parallel compressive sensing. *Nonlinear Dynamics*, 112(22), 20381–20400. <https://doi.org/10.1007/s11071-024-10054-7>
- Gebereselassie, S. A., & Roy, B. K. (2022). Secure speech communication based on the combination of chaotic oscillator and logistic map. *Multimedia Tools and Applications*, 81(18), 26061–26079. <https://doi.org/10.1007/s11042-022-12803-5>
- Gebereselassie, S. A., & Roy, B. K. (2024). Comparative analysis of image encryption based on 1D maps and their integrated chaotic maps. *Multimedia Tools and Applications*, 83(27), 69511–69533. <https://doi.org/10.1007/s11042-024-18319-4>
- Gedda, L., Brenici, G., & Oddi, G. (1977). The chronogenetics of caries in primary dentition. *Acta Geneticae Medicae et Gemellologiae*, 26(1), 17–20. <https://doi.org/10.1017/S000156600001014X>
- Ghazvini, M., Mirzadi, M., & Parvar, N. (2020). A modified method for image encryption based on chaotic map and genetic algorithm. *Multimedia Tools and Applications*, 79(37–

- 38), 26927–26950. <https://doi.org/10.1007/s11042-020-09058-3>
- Gollagi, S. G. (2021). A New Method of Secure Image Encryption by Using Enhanced RSA Algorithm. *2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*, 1, 1–5. <https://doi.org/10.1109/FABS52071.2021.9702550>
- Gollagi, S. G., Srividya, R., Santhosh Kumar, G., & Pareek, P. K. (2021). A New Method of Secure Image Encryption by Using Enhanced RSA Algorithm. *2021 International Conference on Forensics, Analytics, Big Data, Security, FABS 2021*, 1, 1–5. <https://doi.org/10.1109/FABS52071.2021.9702550>
- Guang, Y., Ming, Y., Niu, X., Hua, Y., Wei, Z., Shi, M., & Hua, D. (2023). A visually secure image encryption algorithm based on block compressive sensing and deep neural networks. *Multimedia Tools and Applications*, 0123456789. <https://doi.org/10.1007/s11042-023-16702-1>
- Guo, M., Yang, R., Zhang, M., Liu, R., Zhu, Y., & Dou, G. (2021). A novel memcapacitor and its application in a chaotic circuit. *Nonlinear Dynamics*, 105(1), 877–886. <https://doi.org/10.1007/s11071-021-06627-5>
- Guo, Z., Wen, J., & Mou, J. (2023). Dynamic Analysis and DSP Implementation of Memristor Chaotic Systems with Multiple Forms of Hidden Attractors. *Mathematics*, 11(1), 1–13. <https://doi.org/10.3390/math11010024>
- Gupta, N., & Vijay, R. (2022). Hybrid image compression-encryption scheme based on multilayer stacked autoencoder and logistic map. *China Communications*, 19(1), 238–252. <https://doi.org/10.23919/JCC.2022.01.017>
- Hadj Brahim, A., Ali Pacha, A., & Hadj Said, N. (2024). An image compression-encryption scheme based on compressive sensing and hyperchaotic system. *Journal of Optics (India)*, 0123456789. <https://doi.org/10.1007/s12596-024-02062-y>
- Hameed, M. A., Abdel-Aleem, O. A., & Hassaballah, M. (2023). A secure data hiding approach based on least-significant-bit and nature-inspired optimization techniques. *Journal of Ambient Intelligence and Humanized Computing*, 14(5), 4639–4657. <https://doi.org/10.1007/s12652-022-04366-y>
- Hamza, A., & Kumar, B. (2020). A Review Paper on DES, AES, RSA Encryption Standards. *Proceedings of the 2020 9th International Conference on System Modeling and Advancement in Research Trends, SMART 2020*, 333–338. <https://doi.org/10.1109/SMART50582.2020.9336800>
- Hao, J. I. N., Li, H., Yan, H., & Mou, J. U. N. (2021). A New Fractional Chaotic System and Its Application in Image Encryption With DNA Mutation. 9. <https://doi.org/10.1109/ACCESS.2021.3069977>
- Hashimoto, F., Onishi, Y., Ote, K., Tashima, H., Reader, A. J., & Yamaya, T. (2024). Deep learning - based PET image denoising and reconstruction : a review. *Radiological Physics and Technology*, 24–46. <https://doi.org/10.1007/s12194-024-00780-3>
- He, Y., Zhang, Y. Q., He, X., & Wang, X. Y. (2021). A new image encryption algorithm based on the OF-LSTMS and chaotic sequences. *Scientific Reports*, 11(1), 1–22. <https://doi.org/10.1038/s41598-021-85377-1>
- Heltberg, M. L., Krishna, S., & Jensen, M. H. (2019). On chaotic dynamics in transcription factors and the associated effects in differential gene regulation. *Nature Communications*,

10(1), 1–10. <https://doi.org/10.1038/s41467-018-07932-1>

- Hong, Q., Chen, H., Sun, J., & Wang, C. (2022). Memristive Circuit Implementation of a Self-Repairing Network Based on Biological Astrocytes in Robot Application. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5), 2106–2120. <https://doi.org/10.1109/TNNLS.2020.3041624>
- Hong, Q., Shi, Z., Sun, J., & Du, S. (2021). Memristive self-learning logic circuit with application to encoder and decoder. *Neural Computing and Applications*, 33(10), 4901–4913. <https://doi.org/10.1007/s00521-020-05281-z>
- Houssein, E. H., Gad, A. G., Hussain, K., & Suganthan, P. N. (2021a). Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application. *Swarm and Evolutionary Computation*, 63(March), 100868. <https://doi.org/10.1016/j.swevo.2021.100868>
- Houssein, E. H., Gad, A. G., Hussain, K., & Suganthan, P. N. (2021b). Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application. *Swarm and Evolutionary Computation*, 63(March 2020), 100868. <https://doi.org/10.1016/j.swevo.2021.100868>
- Hu, F., Pu, C., Gao, H., Tang, M., & Li, L. (2016). Image compression and encryption scheme based on deep learning. *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu*, 6, 142–148.
- Hu, M., Li, J., & Di, X. (2023). Quantum image encryption scheme based on 2D Sine2-Logistic chaotic map. *Nonlinear Dynamics*, 111(3), 2815–2839. <https://doi.org/10.1007/s11071-022-07942-1>
- Hu, S., Nalisnick, E., & Welling, M. (2022). *Adversarial Defense via Image Denoising with Chaotic Encryption*. <http://arxiv.org/abs/2203.10290>
- Hu, Z., Li, Y., Jia, L., & Yu, J. (2010). Chaotic oscillator based on voltage-controlled memcapacitor. *2010 International Conference on Communications, Circuits and Systems, ICCAS 2010 - Proceedings*, 5, 824–827. <https://doi.org/10.1109/ICCASCAS.2010.5581863>
- Hu, Z. T., Zeng, P. P., Ding, H. X., Gong, L. H., & Chen, S. H. (2024). Multi-color-image compression-encryption scheme in the quaternion discrete Fresnel domain. *Optica Applicata*, 54(2), 259–276. <https://doi.org/10.37190/oa240211>
- Huang, D., Zhang, Z., Tu, Y., & Xie, L. (2024). Colour Image Encryption System Based on Four-dimensional Memristor Hyperchaotic. *IEEE Access*, 12(July), 161530–161544. <https://doi.org/10.1109/ACCESS.2024.3439191>
- Huang, W., Jiang, D., An, Y., Liu, L., & Wang, X. (2021). A Novel Double-Image Encryption Algorithm Based on Rossler Hyperchaotic System and Compressive Sensing. *IEEE Access*, 9, 41704–41716. <https://doi.org/10.1109/ACCESS.2021.3065453>
- Hussain, K., Rahmatyar, A. R., Riskhan, B., Sheikh, M. A. U., & Sindiramutty, S. R. (2024). Threats and Vulnerabilities of Wireless Networks in the Internet of Things (IoT). *2024 IEEE 1st Karachi Section Humanitarian Technology Conference, Khi-HTC 2024*, 1–8. <https://doi.org/10.1109/KHI-HTC60760.2024.10482197>
- Ibrahim, D., Ahmed, K., Abdallah, M., & Ali, A. A. (2022). A New Chaotic-Based RGB Image Encryption Technique Using a Nonlinear Rotational 16×16 DNA Playfair Matrix.

Cryptography, 6(2). <https://doi.org/10.3390/cryptography6020028>

- Imam, R., Areeb, Q. M., Alturki, A., & Anwer, F. (2021). Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status. *IEEE Access*, 9, 155949–155976. <https://doi.org/10.1109/ACCESS.2021.3129224>
- Iqbal, N., Naqvi, R. A., Atif, M., Khan, M. A., Hanif, M., Abbas, S., & Hussain, D. (2021). On the Image Encryption Algorithm Based on the Chaotic System, DNA Encoding, and Castle. *IEEE Access*, 9, 118253–118270. <https://doi.org/10.1109/ACCESS.2021.3106028>
- Isaac, S. D., Njitacke, Z. T., Tsafack, N., & Tchapgá, C. T. (2022). P HYSICAL J OURNAL Novel compressive sensing image encryption using the dynamics of an adjustable gradient Hopfield neural. *The European Physical Journal Special Topics*, 231(10), 1995–2016. <https://doi.org/10.1140/epjs/s11734-022-00472-2>
- Jessica, J. (2020). *ANNSESDIHM: An Artificial Neural Network based Signal Enhancement Scheme for Digital Inline Holography (DIH) Microscope*. 2020–2022.
- Jiang, X., Jiang, G., Wang, Q., & Shu, D. (2023). Image encryption algorithm based on 2D-CLICM chaotic system. *IET Image Processing*, December 2022, 2127–2141. <https://doi.org/10.1049/ipr2.12778>
- Jithin, K. C., & Sankar, S. (2020). Colour image encryption algorithm combining, Arnold map, DNA sequence operation, and a Mandelbrot set. *Journal of Information Security and Applications*, 50, 102428. <https://doi.org/10.1016/j.jisa.2019.102428>
- Jugale, A. A., Bhoomika, C. M., Anjanakumari, B. T., & Ahmed, M. R. (2019). Memristor based chaotic circuit design for secure communication. *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019, 2019-April(Icoei)*, 185–189. <https://doi.org/10.1109/icoei.2019.8862515>
- Kakkad, V., Patel, M., & Shah, M. (2019). Biometric authentication and image encryption for image security in cloud framework. *Multiscale and Multidisciplinary Modeling, Experiments and Design*, 2(4), 233–248. <https://doi.org/10.1007/s41939-019-00049-y>
- Kaur, R., Kumar, R., & Shalini, R. (2023). *Multilayer extreme learning machine : a systematic review*. 40269–40307.
- Kebbatí, Y. (2022). *Memristor , Memcapacitor , Meminductor : Models and Experimental Circuit Emulators*. 12(3), 8683–8687.
- Kengne, J., Negou, A. N., & Tchiotso, D. (2017). Antimonotonicity, chaos and multiple attractors in a novel autonomous memristor-based jerk circuit. *Nonlinear Dynamics*, 88(4), 2589–2608. <https://doi.org/10.1007/s11071-017-3397-1>
- Kengnou Telem, A. N., Fotsin, H. B., & Kengne, J. (2021). Image encryption algorithm based on dynamic DNA coding operations and 3D chaotic systems. In *Multimedia Tools and Applications* (Vol. 80, Issue 12). Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-021-10549-0>
- Kermány, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., ... Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5), 1122–1131.e9. <https://doi.org/10.1016/j.cell.2018.02.010>

- Khushwant, S., Raghvendra, S., Singh, P., Singh, S., Wadhera, S., Kasturi, P., Jain, G., & Manoj, S. (2023). Secure digital image watermarking using memristor-based hyperchaotic circuit. *The Visual Computer*, 39(10), 4459–4485. <https://doi.org/10.1007/s00371-022-02601-3>
- Kilbas, A. A., & Marzan, S. A. (2005). *Nonlinear Differential Equations with the Caputo Fractional Derivative in the Space of Continuously Differentiable Functions*. 41(1), 84–89.
- Kirtas, M., Passalis, N., Oikonomou, A., Moralis-Pegios, M., Giamougiannis, G., Tsakyridis, A., Mourgiyas-Alexandris, G., Pleros, N., & Tefas, A. (2023). Mixed-precision quantization-aware training for photonic neural networks. *Neural Computing and Applications*, 35(29), 21361–21379. <https://doi.org/10.1007/s00521-023-08848-8>
- Kolivand, H., Hamood, S. F., Asadianfam, S., Mohd Rahim, M. S., & Hurst, W. (2024). Image encryption framework based on multi-chaotic maps and equal pixel values quantization. In *Multimedia Tools and Applications* (Issue 8). <https://doi.org/10.1007/s11042-024-19771-y>
- Kou, L., Huang, Z., Jiang, C., Zhang, F., Ke, W., Wan, J., Liu, H., Li, H., & Lu, J. (2022). Data encryption based on 7D complex chaotic system with cubic memristor for smart grid. *Frontiers in Energy Research*, 10(September), 1–13. <https://doi.org/10.3389/fenrg.2022.980863>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- Kumar, C. M., Vidhya, R., & Brindha, M. (2022). An efficient chaos based image encryption algorithm using enhanced thorp shuffle and chaotic convolution function. *Applied Intelligence*, 52(3), 2556–2585. <https://doi.org/10.1007/s10489-021-02508-x>
- Kumar, S., Dehradun, W. I. T., & Dehradun, W. I. T. (n.d.). *A survey on Symmetric and Asymmetric Key based Image Encryption*.
- Kumar Sharma, D., Chidananda Singh, N., Noola, D. A., Nirmal Doss, A., & Sivakumar, J. (2021). A review on various cryptographic techniques & algorithms. *Materials Today: Proceedings*, 51(xxxx), 104–109. <https://doi.org/10.1016/j.matpr.2021.04.583>
- Kumar, V., & Girdhar, A. (2021). A 2D logistic map and Lorenz-Rossler chaotic system based RGB image encryption approach. *Multimedia Tools and Applications*, 80(3), 3749–3773. <https://doi.org/10.1007/s11042-020-09854-x>
- Lan, Y. C., & Wang, C. M. (2025). A Novel Multi-Image Encryption Scheme Using Generalized Rectangular Transform and Advanced 5-D Hyperchaotic Map. *IEEE Access*, 13(February), 43316–43337. <https://doi.org/10.1109/ACCESS.2025.3547891>
- Lawnik, M., & Berezowski, M. (2022). New Chaotic System: M-Map and Its Application in Chaos-Based Cryptography. *Symmetry*, 14(5), 1–18. <https://doi.org/10.3390/sym14050895>
- Li, C., & Sprott, J. C. (2014). Coexisting hidden attractors in a 4-D simplified lorenz system. *International Journal of Bifurcation and Chaos*, 24(3), 1–12. <https://doi.org/10.1142/S0218127414500345>
- Li, G. H. (2007). Generalized projective synchronization between Lorenz system and Chen's

- system. *Chaos, Solitons and Fractals*, 32(4), 1454–1458. <https://doi.org/10.1016/j.chaos.2005.11.073>
- Li, H., Zhu, J., Ma, C., Zhang, J., & Zong, C. (2017). Multi-modal summarization for asynchronous collection of text, image, audio and video. *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 1, 1092–1102. <https://doi.org/10.18653/v1/d17-1114>
- Li, M., Pan, S., Meng, W., Guoyong, W., Ji, Z., & Wang, L. (2022). Medical image encryption algorithm based on hyper-chaotic system and DNA coding. *Cognitive Computation and Systems*, 4(4), 378–390. <https://doi.org/10.1049/ccs2.12070>
- Li, P., Xu, J., Mou, J., & Yang, F. (2019). Fractional-order 4D hyperchaotic memristive system and application in color image encryption. *Eurasip Journal on Image and Video Processing*, 2019(1). <https://doi.org/10.1186/s13640-018-0402-7>
- Li, R., Liu, Q., & Liu, L. (2019). Novel image encryption algorithm based on improved logistic map. *IET Image Processing*, 13(1), 125–134. <https://doi.org/10.1049/iet-ipr.2018.5900>
- Li, Y., Yu, H., Song, B., & Chen, J. (2021). Image encryption based on a single-round dictionary and chaotic sequences in cloud computing. *Concurrency and Computation: Practice and Experience*, 33(7), 1. <https://doi.org/10.1002/cpe.5182>
- Li, Z., Yang, S., Tan, W., Huang, Z., & Wang, J. (2024). An novel dynamic compressed sensing method for image encryption based on a new coupled map lattices model. *Nonlinear Dynamics*, 112(20), 18501–18525. <https://doi.org/10.1007/s11071-024-09861-9>
- Liang, W., Ning, Z., Xie, S., Hu, Y., Lu, S., & Zhang, D. (2021). Secure fusion approach for the Internet of Things in smart autonomous multi-robot systems. *Information Sciences*, 579, 468–482. <https://doi.org/10.1016/j.ins.2021.08.035>
- Liang, X., Tao, L., & Hu, B. (2023). Image bit planes approximate reconstruction and encryption based on Gaussian function and multiple parameters chaos. *Multimedia Systems*, 29(1), 305–321. <https://doi.org/10.1007/s00530-022-00994-8>
- Liang, Z., Qin, Q., Zhou, C., & Xu, S. (2023a). Color Image Encryption Algorithm Based on Four-Dimensional Multi-stable Hyper Chaotic System and DNA Strand Displacement. *Journal of Electrical Engineering and Technology*, 18(1), 539–559. <https://doi.org/10.1007/s42835-022-01157-5>
- Liang, Z., Qin, Q., Zhou, C., & Xu, S. (2023b). Color Image Encryption Algorithm Based on Four - Dimensional Multi - stable Hyper Chaotic System and DNA Strand Displacement. *Journal of Electrical Engineering & Technology*, 18(1), 539–559. <https://doi.org/10.1007/s42835-022-01157-5>
- Lin, D., Zhang, Q., Chen, X., Li, Z., & Wang, S. (2022). A Color Image Encryption Using One Quaternion-Valued Neural Network. *SSRN Electronic Journal, Zhongshan Li*, 1–29. <https://doi.org/10.2139/ssrn.3998893>
- Lin, H., Wang, C., Cui, L., Sun, Y., Zhang, X., & Yao, W. (2022). Hyperchaotic memristive ring neural network and application in medical image encryption. *Nonlinear Dynamics*, 110(1), 841–855. <https://doi.org/10.1007/s11071-022-07630-0>
- Lin, R., & Li, S. (2021). An Image Encryption Scheme Based on Lorenz Hyperchaotic System and RSA Algorithm. *Security and Communication Networks*, 2021. <https://doi.org/10.1155/2021/5586959>

- Liu, J., Wang, Z., Chen, M., Zhang, P., Yang, R., & Yang, B. (2022). Chaotic system dynamics analysis and synchronization circuit realization of fractional-order memristor. *European Physical Journal: Special Topics*, 231(16–17), 3095–3107. <https://doi.org/10.1140/epjs/s11734-022-00640-4>
- Liu, L., & Miao, S. (2018). A new simple one-dimensional chaotic map and its application for image encryption. *Multimedia Tools and Applications*, 77(16), 21445–21462. <https://doi.org/10.1007/s11042-017-5594-9>
- Liu, M., & Ye, G. (2021a). A new DNA coding and hyperchaotic system based asymmetric image encryption algorithm. *Mathematical Biosciences and Engineering*, 18(4), 3887–3906. <https://doi.org/10.3934/mbe.2021194>
- Liu, M., & Ye, G. (2021b). A new DNA coding and hyperchaotic system based asymmetric image encryption algorithm. 18(April), 3887–3906.
- Liu, Q., & Liu, L. (2020). Color Image Encryption Algorithm Based on DNA Coding and Double Chaos System. *IEEE Access*, 8, 83596–83610. <https://doi.org/10.1109/ACCESS.2020.2991420>
- Liu, X., & Wang, J. (2022). The Simplest Memristor Circuit With Hyperchaos. *Frontiers in Physics*, 10(June), 1–11. <https://doi.org/10.3389/fphy.2022.904200>
- Liu, Y., Jiang, Z., Xu, X., Zhang, F., & Xu, J. (2020). Optical image encryption algorithm based on hyper-chaos and public-key cryptography. *Optics and Laser Technology*, 127(December 2019), 106171. <https://doi.org/10.1016/j.optlastec.2020.106171>
- Liu, Z., Wu, C., Wang, J., & Hu, Y. (2019). A Color Image Encryption Using Dynamic DNA and 4-D Memristive Hyper-Chaos. *IEEE Access*, 7, 78367–78378. <https://doi.org/10.1109/ACCESS.2019.2922376>
- Lone, M. A., & Qureshi, S. (2023). Encryption scheme for RGB images using chaos and affine hill cipher technique. *Nonlinear Dynamics*, 111(6), 5919–5939. <https://doi.org/10.1007/s11071-022-07995-2>
- Lone, P. N., Singh, D., & Mir, U. H. (2022). Image encryption using DNA coding and three-dimensional chaotic systems. *Multimedia Tools and Applications*, 81(4), 5669–5693. <https://doi.org/10.1007/s11042-021-11802-2>
- Lu, J., Zeng, W., Zhang, L., & Shi, Y. (2022). A Novel Key Features Screening Method Based on Extreme Learning Machine for Alzheimer ' s Disease Study. 14(May), 1–14. <https://doi.org/10.3389/fnagi.2022.888575>
- Ma, X., Mou, J., Liu, J., Ma, C., Yang, F., & Zhao, X. (2020). A novel simple chaotic circuit based on memristor–memcapacitor. *Nonlinear Dynamics*, 100(3), 2859–2876. <https://doi.org/10.1007/s11071-020-05601-x>
- Mahadik, S. S., Pawar, P. M., Muthalagu, R., Prasad, N. R., Hawkins, S. K., Stripelis, D., Rao, S., Ejim, P., & Hecht, B. (2024). Digital Privacy in Healthcare: State-of-the-Art and Future Vision. *IEEE Access*, 12(June), 84273–84291. <https://doi.org/10.1109/ACCESS.2024.3410035>
- Maharaja Vannia Moorthi, V., Krishna, M. R., Kumar, G. A., Thanikaiselvan, V., Subashanthini, S., & Amirtharajan, R. (2024). Key-Independent Image Encryption using DNA Encoding and Chaotic Dynamics. *Proceedings of the 2024 10th International Conference on Communication and Signal Processing, ICCSP 2024*, 964–969.

<https://doi.org/10.1109/ICCSP60870.2024.10544118>

- Mahdaoui, A. El, Ouahabi, A., & Moulay, M. S. (2022). Image Denoising Using a Compressive Sensing Approach Based on Regularization Constraints. *Sensors*, 22(6). <https://doi.org/10.3390/s22062199>
- Majstorović, J., Giffard-Roisin, S., & Poli, P. (2023). Interpreting convolutional neural network decision for earthquake detection with feature map visualization, backward optimization and layer-wise relevance propagation methods. *Geophysical Journal International*, 232(2), 923–939. <https://doi.org/10.1093/gji/ggac369>
- Malik, A., Ali, M., Alsubaei, F. S., Ahmed, N., & Kumar, H. (2023). A Color Image Encryption Scheme Based on Singular Values and Chaos. *CMES - Computer Modeling in Engineering and Sciences*, 137(1), 965–999. <https://doi.org/10.32604/cmes.2023.022493>
- Malik, A., Gupta, S., & Dhall, S. (2020). Analysis of traditional and modern image encryption algorithms under realistic ambience. *Multimedia Tools and Applications*, 79(37–38), 27941–27993. <https://doi.org/10.1007/s11042-020-09279-6>
- Man, Z., Li, J., Di, X., Sheng, Y., & Liu, Z. (2021). Double image encryption algorithm based on neural network and chaos. *Chaos, Solitons and Fractals*, 152, 111318. <https://doi.org/10.1016/j.chaos.2021.111318>
- Mandal, M. K., Banik, G. D., Chattopadhyay, D., & Nandi, D. (2012). An image encryption process based on Chaotic logistic map. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 29(5), 395–404. <https://doi.org/10.4103/0256-4602.103173>
- Manikandan, C. S., Rajesh, T., Dhanaraj, K., & Nayyar, A. (2023). DNA Encoded Color Image Encryption Based on Chaotic Sequence from Neural Network. *Journal of Signal Processing Systems*, 95(4), 459–474. <https://doi.org/10.1007/s11265-023-01853-z>
- Manikandan, K., Vijayarajan, V., & Karthikeyan, T. (2011). Secure Mobile Banking Application Using Elliptic Curve Cryptography & RGB Intensity Based Randomized Pixels. *International Journal of Advanced Research in Computer Science*, 2(1).
- Maniyath, S. R., & Thanikaiselvan, V. (2020). An Efficient Image encryption using Deep Neural Network and Chaotic Map. *Microprocessors and Microsystems*, 103134. <https://doi.org/10.1016/j.micpro.2020.103134>
- Marco, M. DI, Forti, M., Corinto, F., & Chua, L. (2021). Unfolding Nonlinear Dynamics in Analogue Systems with Mem-Elements. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(1), 14–24. <https://doi.org/10.1109/TCSI.2020.3024248>
- Massimiliano, P., Ventra, D., & Pershin, Y. V. (n.d.). *Memristors and Memelements Mathematics , Physics and Fiction SpringerBriefs in Physics*.
- Matignon, D. (1996). Stability results for fractional differential equations with applications to control processing. *Computational Engineering in Systems Applications*, January, 963–968. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.4859&rep=rep1&type=pdf>
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.4859>
- Meiliana, A., Dewi, N. M., & Wijaya, A. (2019). Artificial intelligent in healthcare. *Indonesian Biomedical Journal*, 11(2), 125–135. <https://doi.org/10.18585/inabj.v11i2.844>

- Mezatio, B. A., Motchongom Tingue, M., Kengne, R., Tchagna Kouanou, A., Fozin Fonzin, T., & Tchitnga, R. (2020). Complex dynamics from a novel memristive 6D hyperchaotic autonomous system. *International Journal of Dynamics and Control*, 8(1), 70–90. <https://doi.org/10.1007/s40435-019-00531-y>
- Mfungo, D. E., & Fu, X. (2023). Fractal-Based Hybrid Cryptosystem: Enhancing Image Encryption with RSA, Homomorphic Encryption, and Chaotic Maps. *Entropy*, 25(11). <https://doi.org/10.3390/e25111478>
- Miao, K., Feng, Q., & Kuang, W. (2021). Particle swarm optimization combined with inertia-free velocity and direction search. *Electronics (Switzerland)*, 10(5), 1–29. <https://doi.org/10.3390/electronics10050597>
- Mingxing Tan, Q. V. Le. (2013). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks Mingxing. *Canadian Journal of Emergency Medicine*, 15(3), 190.
- Mohamed, H. G., Elkamchouchi, D. H., & Moussa, K. H. (2020). A novel color image encryption algorithm based on hyperchaotic maps and mitochondrial DNA sequences. *Entropy*, 22(2), 7279–7297. <https://doi.org/10.3390/e22020158>
- Mohammad, O. F., Shafry, M., Rahim, M., Rafeeq, S., Zeebaree, M., & Ahmed, F. Y. H. (2017). A Survey and Analysis of the Image Encryption Methods. 12(23), 13265–13280.
- Murillo-Escobar, D., Murillo-Escobar, M. Á., Cruz-Hernández, C., Arellano-Delgado, A., & López-Gutiérrez, R. M. (2023). Pseudorandom number generator based on novel 2D Hénon-Sine hyperchaotic map with microcontroller implementation. *Nonlinear Dynamics*, 111(7), 6773–6789. <https://doi.org/10.1007/s11071-022-08101-2>
- Nadimpalli, A. V., & Rattani, A. (2024). Social Media Authentication and Combating Deepfakes Using Semi-Fragile Invisible Image Watermarking. *Digital Threats: Research and Practice*, 5(4). <https://doi.org/10.1145/3700146>
- Nakazawa, T., & Kulkarni, D. V. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing*, 31(2), 309–314. <https://doi.org/10.1109/TSM.2018.2795466>
- Nan, S. xian, Feng, X. fang, Wu, Y. fei, & Zhang, H. (2022). Remote sensing image compression and encryption based on block compressive sensing and 2D-LCCCM. *Nonlinear Dynamics*, 108(3), 2705–2729. <https://doi.org/10.1007/s11071-022-07335-4>
- Nazir, P., Deep, L., & Mir, H. (2022). Image encryption using DNA coding and three - dimensional chaotic systems. *Multimedia Tools and Applications*, 5669–5693. <https://doi.org/10.1007/s11042-021-11802-2>
- Njitacke, Z. T., Feudjio, C., Signing, V. F., Koumetio, B. N., Tsafack, N., & Awrejcewicz, J. (2022). Circuit and microcontroller validation of the extreme multistable dynamics of a memristive Jerk system: application to image encryption. *European Physical Journal Plus*, 137(5). <https://doi.org/10.1140/epjp/s13360-022-02821-5>
- Oresanya, B. O., Si, G., Guo, Z., Xu, X., & Bie, Y. (2021). Mathematical analysis and emulation of the fractional-order cubic flux-controlled memristor. *Alexandria Engineering Journal*, 60(5), 4315–4324. <https://doi.org/10.1016/j.aej.2021.02.060>
- Ou, J., Li, H., Huang, G., & Yang, G. (2021). Intelligent analysis of tool wear state using stacked denoising autoencoder with online sequential-extreme learning machine. *Measurement: Journal of the International Measurement Confederation*, 167, 108153.

<https://doi.org/10.1016/j.measurement.2020.108153>

- Padmavathi, B., & Kumari, S. R. (2013). *A Survey on Performance Analysis of DES , AES and RSA Algorithm along with LSB Substitution Technique*. 2(4), 170–174.
- Pang, S., & Liu, Y. (2011). A new hyperchaotic system from the Lü system and its control. *Journal of Computational and Applied Mathematics*, 235(8), 2775–2789. <https://doi.org/10.1016/j.cam.2010.11.029>
- Paradesi Priyanka, M., Kaur, N., Nazir, N., Ali Khan, A., Vikram Singh, M., Kaur, M., Behera, T., Rakhra, M., & Dahiya, O. (2022). A Comparative Review between Modern Encryption Algorithms viz. DES, AES, and RSA. *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2022*, 295–300. <https://doi.org/10.1109/CISES54857.2022.9844393>
- Pareek, N. K., Patidar, V., & Sud, K. K. (2006). Image encryption using chaotic logistic map. *Image and Vision Computing*, 24(9), 926–934. <https://doi.org/10.1016/j.imavis.2006.02.021>
- Patel, S., Bharath K P, & Rajesh Kumar M. (2020). Symmetric keys image encryption and decryption using 3D chaotic maps with DNA encoding technique. *Multimedia Tools and Applications*, 79(43–44), 31739–31757. <https://doi.org/10.1007/s11042-020-09551-9>
- Pathak, A., Mondal, H., Karmakar, J., Pal, S., Nandi, D., & Mandal, M. K. (2024). Sparse Compression-Based Image Encryption using Data Encryption Standards RC5. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 41(3), 353–365. <https://doi.org/10.1080/02564602.2023.2240286>
- Paul, L. S. J., Gracias, C., Desai, A., Thanikaiselvan, V., Suba Shanthini, S., & Rengarajan, A. (2022). A novel colour image encryption scheme using dynamic DNA coding, chaotic maps, and SHA-2. *Multimedia Tools and Applications*, 81(26), 37873–37894. <https://doi.org/10.1007/s11042-022-13095-5>
- Peng, X., & Zeng, Y. (2020). Image encryption application in a system for compounding self-excited and hidden attractors. *Chaos, Solitons and Fractals*, 139, 110044. <https://doi.org/10.1016/j.chaos.2020.110044>
- Petráš, I. (2020). Comments “Chaotic oscillator based on memcapacitor and meminductor” (Nonlinear Dyn, DOI: 10.1007/s11071-019-04781-5). *Nonlinear Dynamics*, 102(4), 2945–2950. <https://doi.org/10.1007/s11071-020-06013-7>
- Petráš, I. (2022). Oscillators Based on Fractional-Order Memory Elements. *Fractal and Fractional*, 6(6). <https://doi.org/10.3390/fractalfract6060283>
- Prerna, M., Sachdeva, A., & Mahajan, P. (2013). A Study of Encryption Algorithms AES, DES and RSA for Security AStudyofEncryptionAlgorithmsAESDESandRSAforSecurity A Study of Encryption Algorithms AES, DES and RSA for Security. *Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc*, 13(15).
- Qian, K., Xiao, Y., Wei, Y., Liu, D., Wang, Q., & Feng, W. (2023). A Robust Memristor-Enhanced Polynomial Hyper-Chaotic Map and Its Multi-Channel Image Encryption Application. *Micromachines*, 14(11), 1–25. <https://doi.org/10.3390/mi14112090>
- Rahman, Z. A. S. A., Jasim, B. H., Al-Yasir, Y. I. A., & Abd-Alhameed, R. A. (2022). Efficient Colour Image Encryption Algorithm Using a New Fractional-Order Memcapacitive Hyperchaotic System. *Electronics (Switzerland)*, 11(9).

<https://doi.org/10.3390/electronics11091505>

- Raj, N., Ranjan, R. K., Khateb, F., & Kumngern, M. (2021). Mem-Elements Emulator Design with Experimental Validation and Its Application. *IEEE Access*, 9, 69860–69875. <https://doi.org/10.1109/ACCESS.2021.3078189>
- Raja, N. K., Lydia, E. L., Acharya, T. A., Radhika, K., Yang, E., & Yi, O. (2023). Rider Optimization With Deep Learning Based Image Encryption for Secure Drone Communication. *IEEE Access*, 11(October), 121646–121655. <https://doi.org/10.1109/ACCESS.2023.3324068>
- Rajagopal, K., Jafari, S., Karthikeyan, A., Srinivasan, A., & Ayele, B. (2018). Hyperchaotic Memcapacitor Oscillator with Infinite Equilibria and Coexisting Attractors. *Circuits, Systems, and Signal Processing*, 37(9), 3702–3724. <https://doi.org/10.1007/s00034-018-0750-7>
- Ramli, K., Suryanto, Y., & Hayati, N. U. R. (2020). Novel Image Encryption Using a Pseudoset Generated by Chaotic Permutation Multicircular Shrinking With a Gradual Deletion of the Input Set. 8. <https://doi.org/10.1109/ACCESS.2020.3001949>
- Rashid, A. A., & Hussein, K. A. (2023). Image encryption algorithm based on the density and 6D logistic map. *International Journal of Electrical and Computer Engineering*, 13(2), 1903–1913. <https://doi.org/10.11591/ijece.v13i2.pp1903-1913>
- Ravichandran, D., S, A. B., Murthy, B. K., Balasubramanian, V., & Fathima, S. (2021). An efficient medical image encryption using hybrid DNA computing and chaos in transform domain. 589–605.
- Ren, H. P., Bai, C., Huang, Z. Z., & Grebogi, C. (2017). Secure Communication Based on Hyperchaotic Chen System with Time-Delay. *International Journal of Bifurcation and Chaos*, 27(5). <https://doi.org/10.1142/S0218127417500766>
- Ren, S., He, K., Girshick, R., Zhang, X., & Sun, J. (2017). Object detection networks on convolutional feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7), 1476–1481. <https://doi.org/10.1109/TPAMI.2016.2601099>
- Romero, F. J., Ohata, A., Toral-Lopez, A., Godoy, A., Morales, D. P., & Rodriguez, N. (2021). Memcapacitor and meminductor circuit emulators: A review. *Electronics (Switzerland)*, 10(11). <https://doi.org/10.3390/electronics10111225>
- Rossler, O. E. (1979). An equation for hyperchaos. *Physics Letters A*, 71(2–3), 155–157. [https://doi.org/10.1016/0375-9601\(79\)90150-6](https://doi.org/10.1016/0375-9601(79)90150-6)
- Roy, S., Shrivastava, M., Rawat, U., Pandey, C. V., & Nayak, S. K. (2021). IESCA: An efficient image encryption scheme using 2-D cellular automata. *Journal of Information Security and Applications*, 61, 31529–31567. <https://doi.org/10.1016/j.jisa.2021.102919>
- Sadia, R. T., Chen, J., & Zhang, J. (2024). CT image denoising methods for image quality improvement and radiation dose reduction. In *Journal of Applied Clinical Medical Physics* (Vol. 25, Issue 2, pp. 1–17). <https://doi.org/10.1002/acm2.14270>
- Sahin, M. E., Demirkol, A. S., Guler, H., & Hamamci, S. E. (2020). Design of a hyperchaotic memristive circuit based on wien. *Computers and Electrical Engineering*, 88(August 2019), 106826. <https://doi.org/10.1016/j.compeleceng.2020.106826>
- Sahoo, S., Gebereselassie, S. A., Nathasarma, R., & Roy, B. K. (2024). Maximization of degree

- of hyper-chaos of Lorenz hyper-chaotic system and its use for more secure image encryption. *IFAC-PapersOnLine*, 57, 238–243. <https://doi.org/10.1016/j.ifacol.2024.05.038>
- Sakran, A. A., Hadi, S. M., & Mahmoud Al-Jawher, W. A. (2023). A new approach for DNA sequence analysis using multiwavelet transform (MWT). *Journal of Physics: Conference Series*, 2432(1). <https://doi.org/10.1088/1742-6596/2432/1/012022>
- Sakthi kumar, B., & Revathi, R. (2024). A Comprehensive Review on Image Encryption Techniques using Memristor based Chaotic System for Multimedia Application. *IETE Journal of Research*. <https://doi.org/10.1080/03772063.2024.2373899>
- Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. *ACM International Conference Proceeding Series*, 02-December, 4–11. <https://doi.org/10.1145/2689746.2689747>
- Sánchez-López, C. (2020). An experimental synthesis methodology of fractional-order chaotic attractors. *Nonlinear Dynamics*, 100(4), 3907–3923. <https://doi.org/10.1007/s11071-020-05682-8>
- Sandler, M., Howard, A., Zhu, M., & Zhmoginov, A. (2018). Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.pdf. *ArXiv*, 4510–4520.
- Science, N., & Phenomena, C. (2017). *Chaos , Solitons and Fractals Secure image encryption algorithm design using a novel chaos based , o g.* 95, 92–101. <https://doi.org/10.1016/j.chaos.2016.12.018>
- Scripcariu, L., Diaconu, F., Matasaru, P. D., & Mocanu, F. (2019). Proposal and analysis of a new image encryption algorithm. *ISSCS 2019 - International Symposium on Signals, Circuits and Systems*, 5–8. <https://doi.org/10.1109/ISSCS.2019.8801772>
- Sekar, C., Falmari, V. R., & Brindha, M. (2023). Smart camera with image encryption: a secure solution for real-time monitoring in Industry 4.0. *Journal of Real-Time Image Processing*, 20(4), 1–20. <https://doi.org/10.1007/s11554-023-01338-z>
- Selvi, C. T., & Sudhakar, J. A. R. (2021). Medical image encryption and compression by adaptive sigma filterized synorr certificateless signcryptive Levenshtein entropy - coding - based deep neural learning. *Multimedia Systems*, 1059–1074. <https://doi.org/10.1007/s00530-021-00764-y>
- Senthilkumar, C., Thirumalaisamy, M., Dhanaraj, R. K., & Nayyar, A. (2023). DNA Encoded Color Image Encryption Based on Chaotic Sequence from Neural Network. *Journal of Signal Processing Systems*, 95(4), 459–474. <https://doi.org/10.1007/s11265-023-01853-z>
- Setoudeh, F., & Sedigh, A. K. (2021). Nonlinear analysis and minimum L2-norm control in memcapacitor-based hyperchaotic system via online particle swarm optimization. *Chaos, Solitons and Fractals*, 151, 111214. <https://doi.org/10.1016/j.chaos.2021.111214>
- Sha, Y., Cao, Y., Yan, H., Gao, X., & Mou, J. (2021). An Image Encryption Scheme Based on IAVL Permutation Scheme and DNA Operations. *IEEE Access*, 9, 96321–96336. <https://doi.org/10.1109/ACCESS.2021.3094563>
- Shah, U., Dave, I., Malde, J., Mehta, J., & Kodeboyina, S. (2021). Maintaining Privacy in Medical Imaging with Federated Learning, Deep Learning, Differential Privacy, and Encrypted Computation. *2021 6th International Conference for Convergence in Technology, I2CT 2021*, 1–6. <https://doi.org/10.1109/I2CT51068.2021.9417997>

- Shakir, H. R., Mehdi, S. A. A., & Hattab, A. A. (2022). Chaotic-DNA system for efficient image encryption. *Bulletin of Electrical Engineering and Informatics*, 11(5), 2645–2656. <https://doi.org/10.11591/eei.v11i5.3886>
- Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access*, 10, 10031–10061. <https://doi.org/10.1109/ACCESS.2022.3142859>
- Shanmugam, L., Mani, P., Rajan, R., & Joo, Y. H. (2020). Adaptive Synchronization of Reaction-Diffusion Neural Networks and Its Application to Secure Communication. *IEEE Transactions on Cybernetics*, 50(3), 911–922. <https://doi.org/10.1109/TCYB.2018.2877410>
- Sharma, P. K., Ranjan, R. K., Khateb, F., & Kumngern, M. (2020). Charged controlled MEM-element emulator and its application in a chaotic system. *IEEE Access*, 8, 171397–171407. <https://doi.org/10.1109/ACCESS.2020.3024769>
- Shen, Z., Tang, C., Xu, M., & Lei, Z. (2023). Removal of Speckle Noises from Ultrasound Images Using Parallel Convolutional Neural Network. *Circuits, Systems, and Signal Processing*, 42(8), 5041–5064. <https://doi.org/10.1007/s00034-023-02349-8>
- Shetty, N. P., Muniyal, B., Priyanshu, A., Kumar, D., Melroy Maben, L., Agrawal, Y., Natarajan, R., Gunda, S., & Gupta, N. (2024). Protecting Your Online Persona: A Preferential Selective Encryption Approach for Enhanced Privacy in Tweets, Images, Memes, and Metadata. *IEEE Access*, 12(April), 86403–86424. <https://doi.org/10.1109/ACCESS.2024.3415663>
- Singh, R. S. (2024). *Image encryption algorithm based on a memcapacitor-based hyperchaotic system and DNA coding*. June. <https://doi.org/10.1002/spy2.432>
- Smith, P., Reid, D. B., Environment, C., Palo, L., Alto, P., & Smith, P. L. (1979). Smith et al. - 1979 - A Threshold Selection Method from Gray-Level Histograms. *IEEE Transaction on Systems, Man and Cybernetics*, 20(1), 62–66.
- Sun, J., Han, G., & Wang, Y. (2020). Dynamical Analysis of Memcapacitor Chaotic System and Its Image Encryption Application. *International Journal of Control, Automation and Systems*, 18(5), 1242–1249. <https://doi.org/10.1007/s12555-019-0015-7>
- Sun, J., Ma, Y., Wang, Z., & Wang, Y. (2023). Dynamic analysis and cryptographic application of a 5D hyperbolic memristor-coupled neuron. *Nonlinear Dynamics*, 111(9), 8751–8769. <https://doi.org/10.1007/s11071-023-08237-9>
- Sun, Y., Qi, G., Wang, Z., Van Wyk, B. J., & Hamam, Y. (2009). Chaotic particle swarm optimization. *2009 World Summit on Genetic and Evolutionary Computation, 2009 GEC Summit - Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, GEC'09, February 2014*, 505–510. <https://doi.org/10.1145/1543834.1543902>
- Suri, S., & Vijay, R. (2019). A synchronous intertwining logistic map-DNA approach for color image encryption. *Journal of Ambient Intelligence and Humanized Computing*, 10(6), 2277–2290. <https://doi.org/10.1007/s12652-018-0825-0>
- Susanto, A., Setiadi, D. R. I. M., Rachmawanto, E. H., Mulyono, I. U. W., Sari, C. A., Sarker, M. K., & Szal, M. R. (2020). Triple layer image security using bit-shift, chaos, and stream encryption. *Bulletin of Electrical Engineering and Informatics*, 9(3), 980–987.

<https://doi.org/10.11591/eei.v9i3.2001>

- System, F. M. H., Rahman, Z. S. A., Jasim, B. H., Al-yasir, Y. I. A., & Abd-alhameed, R. A. (2022). *Efficient Colour Image Encryption Algorithm Using a New*.
- System, H., & Filtering, P. (2020). *A Novel Image Encryption Approach Based on a Hyperchaotic System, Pixel-Level Filtering with Variable Kernels, and DNA-Level Diffusion*. <https://doi.org/10.3390/e22010005>
- System, H., & Sequence, D. N. A. (2022). *A Novel Color Image Encryption Algorithm Based on 5-D Hyperchaotic System and DNA Sequence*.
- Teng, L., Wang, X., & Ye, X. (2020). Hyperchaotic Behavior in the Novel Memristor-Based Symmetric Circuit System. *IEEE Access*, 8, 151535–151545. <https://doi.org/10.1109/ACCESS.2020.3017159>
- Theramban, A., & Ravi, R. V. (2021). Colour Image Encryption Using DNA Coding and Logistic Diffusion. *Proceedings of the 4th International Conference on Microelectronics, Signals and Systems, ICMSS 2021*. <https://doi.org/10.1109/ICMSS53060.2021.9673628>
- Tian, Y., & Jiang, Z. (2024). Design of Real-Time Media Encrypted Conference System Based on WebRTC. *2024 13th International Conference of Information and Communication Technology (ICTech)*, 326–331. <https://doi.org/10.1109/ICTech63197.2024.00067>
- Torabi, H., Mirtaheri, S. L., & Greco, S. (2023). Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, 6(1), 1–13. <https://doi.org/10.1186/s42400-022-00134-9>
- Tutueva, A. V., Nepomuceno, E. G., Karimov, A. I., Andreev, V. S., & Butusov, D. N. (2020). Adaptive chaotic maps and their application to pseudo-random numbers generation. *Chaos, Solitons and Fractals*, 133. <https://doi.org/10.1016/j.chaos.2020.109615>
- Valandar, M. Y., Barani, M. J., & Ayubi, P. (2019). A fast color image encryption technique based on three dimensional chaotic map. *Optik*, 193(April). <https://doi.org/10.1016/j.ijleo.2019.06.021>
- Varun, V., Natarajan, S. K., Adithya, M., Nagatejas, P., A, M. C., & Hosahalli, N. M. (2024). Efficient CNN-Based Bone Fracture Detection in X- Ray Radiographs with MobileNetV2. *2024 2nd International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS)*, 72–77. <https://doi.org/10.1109/ICRAIS62903.2024.10811726>
- Velho, L. (2009). *Compressive Sensing In Image Compression*.
- Wan, Y., Gu, S., & Du, B. (2020). A new image encryption algorithm based on composite chaos and hyperchaos combined with DNA coding. *Entropy*, 22(2). <https://doi.org/10.3390/e22020171>
- Wang, B., Zou, F. C., & Cheng, J. (2018). A memristor-based chaotic system and its application in image encryption. *Optik*, 154, 538–544. <https://doi.org/10.1016/j.ijleo.2017.10.080>
- Wang, H., Dong, B., Ke, H., Qi, Q., & Liu, G. (2023). A novel image shift encryption algorithm based on the dynamic Joseph ring problem. In *Multimedia Tools and Applications*. Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-023-14947-4>
- Wang, L., Liu, J., & Lu, Z. R. (2017). Incremental response sensitivity approach for parameter

- identification of chaotic and hyperchaotic systems. *Nonlinear Dynamics*, 89(1), 153–167. <https://doi.org/10.1007/s11071-017-3442-0>
- Wang, T., & Wang, M. (2020a). Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. *Optics and Laser Technology*, 132(March), 106355. <https://doi.org/10.1016/j.optlastec.2020.106355>
- Wang, T., & Wang, M. hui. (2020b). Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding. *Optics and Laser Technology*, 132(April), 106355. <https://doi.org/10.1016/j.optlastec.2020.106355>
- Wang, X., Cao, Y., Li, H., & Li, B. (2023). A Chaos-Enhanced Fractional-Order Chaotic System with Self-Reproduction Based on a Memcapacitor and Meminductor.
- Wang, X., & Du, X. (2022). Chaotic image encryption method based on improved zigzag permutation and DNA rules. *Multimedia Tools and Applications*, 81(30), 43777–43803. <https://doi.org/10.1007/s11042-022-13012-w>
- Wang, X., & Guan, N. (2023). 2D sine-logistic-tent-coupling map for image encryption. *Journal of Ambient Intelligence and Humanized Computing*, 14(10), 13399–13419. <https://doi.org/10.1007/s12652-022-03794-0>
- Wang, X., Li, Y., & Jin, J. (2020). A new one-dimensional chaotic system with applications in image encryption. *Chaos, Solitons and Fractals*, 139, 110102. <https://doi.org/10.1016/j.chaos.2020.110102>
- Wang, X. Y., Fitch, A. L., Iu, H. H. C., & Qi, W. G. (2012). Design of a memcapacitor emulator based on a memristor. *Physics Letters, Section A: General, Atomic and Solid State Physics*, 376(4), 394–399. <https://doi.org/10.1016/j.physleta.2011.11.012>
- Wang, X., Yu, J., Jin, C., Iu, H. H. C., & Yu, S. (2019). Chaotic oscillator based on memcapacitor and meminductor. *Nonlinear Dynamics*, 96(1), 161–173. <https://doi.org/10.1007/s11071-019-04781-5>
- Wang, X., Zhang, X., Gao, M., Tian, Y., Wang, C., & Iu, H. H. C. (2023). A Color Image Encryption Algorithm Based on Hash Table, Hilbert Curve and Hyper-Chaotic Synchronization. *Mathematics*, 11(3). <https://doi.org/10.3390/math11030567>
- Wang, Y., Qiu, Y., Cheng, P., & Duan, X. (2020). Robust loop closure detection integrating visual–spatial–semantic information via topological graphs and CNN features. *Remote Sensing*, 12(23), 1–26. <https://doi.org/10.3390/rs12233890>
- Wang, Y., Wu, C., Kang, S., Wang, Q., & Mikulovich, V. I. (2020). Multi-channel chaotic encryption algorithm for color image based on DNA coding. *Multimedia Tools and Applications*, 79(25–26), 18317–18342. <https://doi.org/10.1007/s11042-020-08742-8>
- Wei, M., Du, Y., Wu, X., Su, Q., Zhu, J., Zheng, L., Lv, G., & Zhuang, J. (2020). A Benign and Malignant Breast Tumor Classification Method via Efficiently Combining Texture and Morphological Features on Ultrasound Images. *Computational and Mathematical Methods in Medicine*, 2020. <https://doi.org/10.1155/2020/5894010>
- Wen, H., Liu, Z., Lai, H., Zhang, C., Liu, L., Yang, J., Lin, Y., Li, Y., Liao, Y., Ma, L., Chen, Z., & Li, R. (2022). Secure DNA-Coding Image Optical Communication Using Non-Degenerate Hyperchaos and Dynamic Secret-Key. *Mathematics*, 10(17), 1–18. <https://doi.org/10.3390/math10173180>

- Wu, D., & Wang, X. (2021). *A Hybrid Method Based on Extreme Learning Machine and Wavelet Transform Denoising for Stock Prediction*.
- Wu, J., & Zhang, Z. (2021). *Extreme Learning Machine Denoising Algorithm Based Analysis of Transvaginal 3-Dimensional Ultrasonic Image for the Diagnostic Effect of Intrauterine Adhesion*. 2021.
- Wu, X., Han, Y., Zhang, M., Li, Y., & Cui, S. (2025). GAN-based pseudo random number generation optimized through genetic algorithms. *Complex and Intelligent Systems*, 11(1). <https://doi.org/10.1007/s40747-024-01606-w>
- Xie, H. (2021). *Novel Medical Image Cryptogram Technology Based on Segmentation and DNA Encoding*.
- Xuejing, K., & Zihui, G. (2020). A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. *Signal Processing: Image Communication*, 80(August 2019), 115670. <https://doi.org/10.1016/j.image.2019.115670>
- Yan, D., Wang, L., Duan, S., Chen, J., & Chen, J. (2021). Chaotic Attractors Generated by a Memristor-Based Chaotic System and Julia Fractal. *Chaos, Solitons and Fractals*, 146, 110773. <https://doi.org/10.1016/j.chaos.2021.110773>
- Yan, S., Li, L., Gu, B., Sun, X., Ren, Y., & Zhang, Y. (2023). A color image encryption scheme based on chaotic mapping, chaotic system, and DNA coding. *Applied Intelligence*, 53(24), 31181–31206. <https://doi.org/10.1007/s10489-023-04759-2>
- Yang, G., Zhang, X., & Moshayedi, A. J. (2023). Implementation of the Simple Hyperchaotic Memristor Circuit with Attractor Evolution and Large-Scale Parameter Permission. *Entropy*, 25(2), 1–20. <https://doi.org/10.3390/e25020203>
- Yang, X., Xi, W., Chen, A., & Wang, C. (2021). An environmental monitoring data sharing scheme based on attribute encryption in cloud-fog computing. *PLoS ONE*, 16(9 September), 1–21. <https://doi.org/10.1371/journal.pone.0258062>
- Yang, X., Xu, K., Xu, S., & Liu, P. X. (2019). Image Noise Level Estimation for Rice Noise Based on Extended ELM Neural Network Training Algorithm. *IEEE Access*, 7, 1943–1951. <https://doi.org/10.1109/ACCESS.2018.2886294>
- Yang, Y. G., Wang, Z. J., Wang, B. P., Zhou, Y. H., Shi, W. M., & Liao, X. (2024). A new visually meaningful double-image encryption algorithm combining 2D compressive sensing with fractional-order chaotic system. In *Multimedia Tools and Applications* (Vol. 83, Issue 2). Springer US. <https://doi.org/10.1007/s11042-023-15662-w>
- Yang, Y., Wang, L., Duan, S., & Luo, L. (2021). Dynamical analysis and image encryption application of a novel memristive hyperchaotic system. *Optics and Laser Technology*, 133(April 2020), 106553. <https://doi.org/10.1016/j.optlastec.2020.106553>
- Yang, Y., Zhang, H., Yuan, D., Sun, D., Li, G., Ranjan, R., & Sun, M. (2018). Hierarchical extreme learning machine based image denoising network for visual Internet of things. *Applied Soft Computing Journal*. <https://doi.org/10.1016/j.asoc.2018.08.046>
- Yao, W., Gao, K., Zhang, Z., Cui, L., & Zhang, J. (2023). An image encryption algorithm based on a 3D chaotic Hopfield neural network and random row–column permutation. *Frontiers in Physics*, 11(March), 1–14. <https://doi.org/10.3389/fphy.2023.1162887>
- Yildirim, M. (2020). DNA encoding for RGB image encryption with memristor based neuron

- model and chaos phenomenon. *Microelectronics Journal*, 104(August), 104878. <https://doi.org/10.1016/j.mejo.2020.104878>
- Yocam, E., & Wang, Y. (n.d.). A Novel Approach to Quantum-Resistant Selective Encryption for Agricultural Sensors with Limited Resources. *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, 262–271. <https://doi.org/10.1109/CCWC62904.2025.10903955>
- Youn, S. J., Yun, S. Y., Lee, H., Park, K. J., Kim, J., & Kim, S. Y. (2023). Design of Low-Noise CMOS Image Sensor Using a Hybrid-Correlated Multiple Sampling Technique. *Sensors*, 23(23). <https://doi.org/10.3390/s23239551>
- Yousif, S. F., Abboud, A. J., & Alhumaima, R. S. (2022). A new image encryption based on bit replacing, chaos and DNA coding techniques. In *Multimedia Tools and Applications* (Vol. 81, Issue 19). Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-022-12762-x>
- Yu, F., Chen, H., Kong, X., Yu, Q., Cai, S., Huang, Y., & Du, S. (2022). Dynamic analysis and application in medical digital image watermarking of a new multi-scroll neural network with quartic nonlinear memristor. *European Physical Journal Plus*, 137(4). <https://doi.org/10.1140/epjp/s13360-022-02652-4>
- Yu, F., Qian, S., Chen, X., Huang, Y., Liu, L., Shi, C., Cai, S., Song, Y., & Wang, C. (2020). A New 4D Four-Wing Memristive Hyperchaotic System: Dynamical Analysis, Electronic Circuit Design, Shape Synchronization and Secure Communication. *International Journal of Bifurcation and Chaos*, 30(10), 1–20. <https://doi.org/10.1142/S0218127420501473>
- Yu, J., Yu, X., Zhang, L., & Xie, W. (2023). A two-stage chaotic encryption algorithm for color face image based on circular diffusion. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-023-14804-4>
- Yuan, F., & Li, Y. (2019). A chaotic circuit constructed by a memristor, a memcapacitor and a meminductor. *Chaos*, 29(10). <https://doi.org/10.1063/1.5125673>
- Yuan, F., Wang, G., & Wang, X. (2017). Chaotic oscillator containing memcapacitor and meminductor and its dimensionality reduction analysis. *Chaos*, 27(3). <https://doi.org/10.1063/1.4975825>
- Zan, T., Liu, Z., Wang, H., Wang, M., & Gao, X. (2020). Control chart pattern recognition using the convolutional neural network. *Journal of Intelligent Manufacturing*, 31(3), 703–716. <https://doi.org/10.1007/s10845-019-01473-0>
- Zhang, J., Li, Y., Xiao, W., & Zhang, Z. (2020). Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines. *Journal of the Franklin Institute*. <https://doi.org/10.1016/j.jfranklin.2020.04.033>
- Zhang, J., Liu, E., & Guo, Y. (2024). A new three-dimensional memristor chaotic circuit design and its application in image encryption. *Journal of Supercomputing*, 80(10), 14694–14724. <https://doi.org/10.1007/s11227-024-05983-4>
- Zhang, J., & Xu, L. (2023). A novel asymmetrical double-wing hyperchaotic system with multiple different attractors: application to finite-time synchronization and image encryption. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-023-15117-2>
- Zhang, M., Tong, X. J., Liu, J., Wang, Z., Liu, J., Liu, B., & Ma, J. (2020). Image Compression

- and Encryption Scheme Based on Compressive Sensing and Fourier Transform. In *IEEE Access* (Vol. 8, pp. 40838–40849). <https://doi.org/10.1109/ACCESS.2020.2976798>
- Zhang, N., Ying, S., Zhu, K., & Zhu, D. (2022). Software defect prediction based on stacked sparse denoising autoencoders and enhanced extreme learning machine. *IET Software*, 16(1), 29–47. <https://doi.org/10.1049/sfw2.12029>
- Zhang, Q., & Han, J. (2021). A novel color image encryption algorithm based on image hashing, 6D hyperchaotic and DNA coding. *Multimedia Tools and Applications*, 80(9), 13841–13864. <https://doi.org/10.1007/s11042-020-10437-z>
- Zhang, S., & Liu, L. (2021). A novel image encryption algorithm based on SPWLCM and DNA coding. *Mathematics and Computers in Simulation*, 190, 723–744. <https://doi.org/10.1016/j.matcom.2021.06.012>
- Zhang, S., Zhang, H., & Wang, C. (2023). Dynamical analysis and applications of a novel 2-D hybrid dual-memristor hyperchaotic map with complexity enhancement. *Nonlinear Dynamics*, 111(16), 15487–15513. <https://doi.org/10.1007/s11071-023-08652-y>
- Zhang, T., & Wang, S. (2023). Image encryption scheme based on a controlled zigzag transform and bit-level encryption under the quantum walk. *Frontiers in Physics*, 10(January), 1–12. <https://doi.org/10.3389/fphy.2022.1097754>
- Zhang, X., & Ye, R. (2021). A novel RGB image encryption algorithm based on DNA sequences and chaos. *Multimedia Tools and Applications*, 80(6), 8809–8833. <https://doi.org/10.1007/s11042-020-09465-6>
- Zhang, X., Zhou, L., Gu, H., & Chen, W. (2015). Image encryption based on complicated chaotic. *Journal of Fiber Bioengineering and Informatics*, 8(2), 381–389. <https://doi.org/10.3993/jfbim00115>
- Zhang, Y. (2018). Test and Verification of AES Used for Image Encryption. *3D Research*, 9(1). <https://doi.org/10.1007/s13319-017-0154-7>
- Zhao, M., Li, L., & Yuan, Z. (2024). An image encryption approach based on a novel two-dimensional chaotic system. *Nonlinear Dynamics*, 112(22), 20509–20843. <https://doi.org/10.1007/s11071-024-10053-8>
- Zhao, Q., Wang, C., & Zhang, X. (2019). A universal emulator for memristor, memcapacitor, and meminductor and its chaotic circuit. *Chaos*, 29(1). <https://doi.org/10.1063/1.5081076>
- Zhong, Y., Liu, H., Lan, R., Wang, T., Sun, X., & Luo, X. (2018). 2D chebyshev-sine map for image encryption. *Proceedings - 7th International Conference on Digital Home, ICDH 2018*, 1–8. <https://doi.org/10.1109/ICDH.2018.00008>
- Zhou, S., & Tan, B. (2020). Electrocardiogram soft computing using hybrid deep learning CNN-ELM. *Applied Soft Computing Journal*, 86(xxxx), 105778. <https://doi.org/10.1016/j.asoc.2019.105778>
- Zhou, W., Wang, G., Iu, H. H. C., Shen, Y., & Liang, Y. (2020). Complex dynamics of a non-volatile memcapacitor-aided hyperchaotic oscillator. *Nonlinear Dynamics*, 100(4), 3937–3957. <https://doi.org/10.1007/s11071-020-05722-3>
- Zolfaghari, B., & Koshiba, T. (2022). Chaotic Image Encryption: State-of-the-Art, Ecosystem, and Future Roadmap. *Applied System Innovation*, 5(3), 1–38. <https://doi.org/10.3390/asi5030057>

- Zou, C., Zhang, Q., Wei, X., & Liu, C. (2020). Image Encryption Based on Improved Lorenz System. *IEEE Access*, 8, 75728–75740. <https://doi.org/10.1109/ACCESS.2020.2988880>
- Zou, Q., Zhong, Q., Mao, J., Ding, R., Lu, D., Li, J., & Li, X. (2023). Impact of Perturbation Schemes on the Ensemble Prediction in a Coupled Lorenz Model. *Advances in Atmospheric Sciences*, 40(3), 501–513. <https://doi.org/10.1007/s00376-022-1376-z>

APPENDIX A: MATHEMATICAL MODELS

The following fractional-order differential equations depict the dynamic evolution of the chaotic system utilised in the encryption framework.

Equation (A1):

$$\begin{cases} c_1^2 = (ac_2^1 + b(c_2^1 \sin c_4^0 + c_2^0 c_4^1 \cos c_4^0) - c_1^1/(ef)) \\ \quad - (cc_3^1 + d(-c_5^1 c_5^0 \sin c_5^0 + c_3^1 \cos c_5^0))/f \\ c_2^2 = c_1^1/e + (g - 1/e)(ac_2^1 + b(c_2^1 \sin c_4^0 + c_2^0 c_4^1 \cos c_4^0)) \\ c_3^2 = c_1^1 \\ c_4^2 = c_1^2 \\ c_5^2 = c_3^1 \end{cases} \quad (\text{A1})$$

Equation (A2):

$$\begin{cases} c_1^3 = \left(\frac{ac_2^2 + b(c_2^2 \sin c_4^0 + c_2^0 c_4^2 \cos c_4^0)}{+(c_2^1 c_4^1 \cos c_4^0 - 0.5c_2^0 c_4^1 c_4^1 \sin c_4^0) \frac{\Gamma(2q+1)}{\Gamma^2(q+1)} - c_1^2} \right) / (ef) \\ \quad - \left(cc_3^2 + d \left(\frac{c_3^2 \cos c_5^0 - c_5^2 c_3^0 \sin c_5^0}{+(-c_5^1 c_3^1 \sin c_5^0 - 0.5 c_5^1 c_5^1 c_3^0 \cos c_5^0) \frac{\Gamma(2q+1)}{\Gamma^2(q+1)}} \right) \right) / f \\ c_2^3 = c_1^2/e + (g - 1/e) \left(ac_2^2 + b \left(\frac{c_2^2 \sin c_4^0 + c_2^0 c_4^2 \cos(c_4^0)}{+ \left(\frac{c_2^1 c_4^1 \cos c_4^0}{-0.5c_2^0 c_4^1 c_4^1 \sin c_4^0} \right)} \right) \frac{\Gamma(2q+1)}{\Gamma^2(q+1)} \right) \\ c_3^3 = c_1^2 \\ c_4^3 = c_2^2 \\ c_5^3 = c_3^2 \end{cases} \quad (\text{A2})$$

Equation (A3):

$$\begin{aligned}
 c_1^4 = & \left(ac_2^3 + b \left(\begin{aligned} & c_2^3 \sin(c_4^0) + c_2^0 c_4^3 \cos(c_4^0) \\ & + (c_2^1 c_4^2 + c_2^2 c_4^1) \cos(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - c_2^0 c_4^1 c_4^2 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & 0.5 c_2^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - \frac{1}{6} c_2^0 c_4^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} - c_1^3 \end{aligned} \right) / (ef) \right) / f \\
 & - \left(cc_3^3 + d \left(\begin{aligned} & c_3^3 \cos(c_5^0) - c_5^3 c_3^0 \sin(c_5^0) \\ & - (c_5^2 c_3^1 - c_5^1 c_3^2) \sin(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - c_5^1 c_5^2 c_3^0 \cos(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - 0.5 c_3^1 c_5^1 c_5^1 \cos(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - \frac{1}{6} c_3^0 c_5^1 c_5^1 c_5^1 \sin(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \end{aligned} \right) \right) \\
 c_2^4 = c_1^3/e + (g-1/e) & \left(ac_2^3 + b \left(\begin{aligned} & c_2^3 \sin(c_4^0) + c_2^0 c_4^3 \cos(c_4^0) \\ & + (c_5^2 c_3^1 - c_5^1 c_3^2) \cos(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - c_2^0 c_4^1 c_4^2 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - 0.5 c_2^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ & - \frac{1}{6} c_2^0 c_4^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \end{aligned} \right) \right) \\
 c_3^4 = c_1^3 & \\
 c_4^4 = c_2^3 & \\
 c_5^4 = c_3^3 &
 \end{aligned} \tag{A3}$$

Equation (A4):

$$\left\{ \begin{array}{l}
 c_1^4 = \left(\left(ac_2^3 + b \left(\begin{array}{l} c_2^3 \sin(c_4^0) + c_2^0 c_4^3 \cos(c_4^0) \\ + (c_2^1 c_4^2 + c_2^2 c_4^1) \cos(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - c_2^0 c_4^1 c_4^2 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ 0.5 c_2^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - \frac{1}{6} c_2^0 c_4^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} - c_1^3 \end{array} \right) / (ef) \right) / f \\
 - \left(dc_3^3 + d \left(\begin{array}{l} c_3^3 \cos(c_5^0) - c_5^3 c_3^0 \sin(c_5^0) \\ - (c_5^2 c_3^1 - c_5^1 c_3^2) \sin(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - c_5^1 c_5^2 c_3^0 \cos(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - 0.5 c_3^1 c_5^1 c_5^1 \cos(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - \frac{1}{6} c_3^0 c_5^1 c_5^1 c_5^1 \sin(c_5^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \end{array} \right) \right) \\
 c_2^4 = c_1^3/e + (g - 1/e) \left(ac_2^3 + b \left(\begin{array}{l} c_2^3 \sin(c_4^0) + c_2^0 c_4^3 \cos(c_4^0) \\ + (c_5^2 c_3^1 - c_5^1 c_3^2) \cos(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - c_2^0 c_4^1 c_4^2 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - 0.5 c_2^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \\ - \frac{1}{6} c_2^0 c_4^1 c_4^1 c_4^1 \sin(c_4^0) \frac{\Gamma(3q+1)}{\Gamma(q+1)\Gamma(2q+1)} \end{array} \right) \right) \\
 c_3^4 = c_1^3 \\
 c_4^4 = c_2^3 \\
 c_5^4 = c_3^3
 \end{array} \right. \quad (A4)$$